An Efficient Data Search and Replication Strategy in Wireless Sensor Networks

張益銘 羅壽之 邱奕庭
東華大學資訊工程學系
sclo@mail.ndhu.edu.tw

摘 要

資料搜尋在無線感測網路中，是一個重要的服務。而一個有效的資料搜尋方法，除了可以增加資料搜尋的速度之外，更可以進一步的節省搜尋資料所耗損的能量。在這篇論文中，我們提出一種結合資料複製機制的搜尋策略。這個策略主要建構於叢集架構上，透過叢集我們將整個網路切割成數個區域，然後，資料的複製與搜尋主要沿著區域的邊緣進行。模擬實驗證實這個方式比另一種在無線感測網路中熟知的謠言路由協定表現的還要好。

關鍵字：無線感測器網路，叢集，資料搜尋，資料複製。

Abstract

Data search is a very important service in wireless sensor networks. A good data search strategy can not only shorten the search latency but also can save the energy. In this paper, we propose an efficient search strategy which combines a data replication mechanism. Our proposed strategy is based on a cluster architecture. Through the clusters, we divide the entire network into zones. Then we replicate and search data along the borders of zones. The simulation results show that this strategy can outperform the rumor routing protocol which is a well-known strategy in wireless sensor networks.

Keywords: wireless sensor networks, clusters, data search, data replication.

1. Introduction

Wireless sensor networks are highly distributed networks of small and lightweight sensor nodes [3], which are deployed in large numbers to monitor the physical environment such as temperature, pressure, light, or humidity. The applications of wireless sensor networks include battlefield detection, home security, and industrial environment surveillance.

The wireless sensor networks usually contain thousands or millions of sensors, which are randomly and widely deployed by pre-planning or dropped by a flying vehicle, to cover the entire sensor field. A sensor network provides a global view of the monitored area with local observations measured by each sensor.

The wireless sensor nodes can configure themselves, in an ad hoc fashion, to form a network so that sensed data can be transmitted across the sensor field hop by hop. Sensor nodes are powered by battery and are difficult to get recharged after deployment. Thus, energy efficiency is an important issue in sensor networks. When sensor nodes sense some specific events, they will advertise the events to the neighboring sensor nodes. When a sink node is interested in some specific events, it will transmit a query which describes the desired type of events to the neighboring sensor nodes or the entire sensor network.

For the deliveries of events or queries, the routing protocol plays an important role. The routing protocol must be designed in a way that the limited power in each sensor node is efficiently used.

In this paper, we propose a new routing protocol which contains both the data replication and data search mechanisms. This protocol is more efficient and could save more energy as well. Besides, we adopt a cluster architecture to construct the network, where we mainly replicate and search data in the overlapping areas of clusters.

The remaining paper is organized as follows. Section 2 gives the related work. Section 3 describes our proposed protocol. Section 4 introduces the performance evaluation using the simulator NS-2. In Section 5, we draw a conclusion.

2. Related Work

In wireless sensor networks, data search [3] is performed by a routing protocol. We can classify the routing protocols into three categories [5, 6, 7]: pure push, pure pull, and hybrid. In the pure push [8], a sensor node when detecting an event will actively announce the information to other sensor nodes along a certain path. Any interested node to this event will eventually receive this announcement. The advantage of this approach is that any interested node can passively receive data without issuing any request. This advantage becomes more significant when there are many interested nodes in the network. The pure push can be viewed as a data replication process by the source node of the event. The disadvantage of this approach is the possible long latency for an interested node getting its desired events. Moreover, some non-interested nodes may also receive the event data.

In the pure pull, a sensor (or sink) node when interested in a certain event will actively issue a search query to locate this event. The query which describes the type of interested events will be forwarded through the network until the desired event is found or the maximal number of forwarding times is reached. The advantage of this approach is that the number of sensor nodes involved in the search process might be small. The drawback of this approach is that it might take much time to search and retrieve the event back.

The hybrid approach [12] combines the pure push and pure pull, where data replication and data search are performed simultaneously. The major advantage of this approach is that the search time can be shortened.

Next, we discuss how to forward the replicated events or the search quires. The simplest way is to flood them into the entire network, but the energy consumption would be large. Another way is only forward them along a certain path to save the energy. This path can be built based on a random walk model. However, the random walk might be inefficient due to the blind behavior.

The rumor routing protocol [1] is another well-known way to forward events/queries along a near straight line. This protocol is a hybrid approach where there exists both replication path and search path. It has been proved that two lines in a bounded rectangle have a 69% chance of intersecting. Figure 1 shows the rumor routing protocol where a source node builds one replication path and a sink node builds one search path.

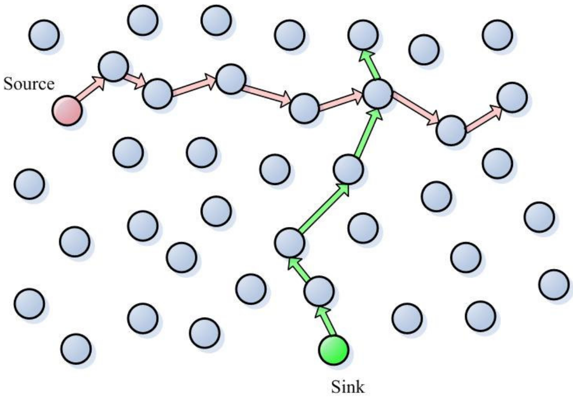

Figure 1. Rumor routing.

3. Proposed Efficient Data Search and Replication Strategy

In this section, we introduce our proposed protocol called Efficient Data Search and Replication sTrategy (abbreviated as EDSRT). EDSRT is a hybrid protocol where data replication and data search use the same mechanism.

We observe that any blind search protocol will waste time and energy to seek the desired events on sensor nodes. Therefore, our main idea is to replicate events to specific nodes only. When performing data search, we look for these specific nodes to speed up the search time.

The question is how we define these specific nodes in wireless sensor networks. Some protocols are developed using location-assisted routings [4, 9] by the global position system (GPS). In these protocols, these specific nodes are defined with system-specific locations. However, using the positioning system in wireless sensor networks may be inflexible or costly.

Here, we borrow a cluster-based architecture [2, 8, 10, 11] to define these specific nodes without extra hardware cost. Figure 2 shows one cluster architecture. The black nodes are called cluster heads which are randomly selected from the sensor nodes. Each white node called a cluster member will join the nearest cluster head to be a member.

Conceptually, the groups of cluster members of the same clusters will divide the entire network into individual zones (the polygon areas in the figure). We therefore define the sensor nodes close to the borders of these zones as specific nodes. Basically, we replicate and search events along the borders of zones. These borders become guided lines to replicate and search events.

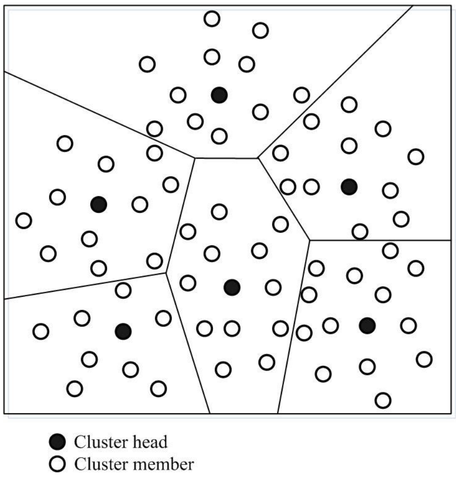

Figure 2. Cluster organization.

Figure 3 shows another view of the cluster architecture. Here we classify specific nodes into two types: border nodes and gateway nodes.

Definition 1: A *border node* is a sensor node along the border of a cluster.

Definition 2: A *gateway node* is a sensor node located in the overlapping area of clusters.

As can be seen, a gateway node is also a border node of more than one cluster. In EDSRT, we will take advantage of these gateway and border nodes in event replication and search. The operations of EDSRT involve two parts. The first part is the cluster organization which mainly identifies some sensor nodes as gateway and border nodes. The second part is the route establishment which presents how a replication path and a search path are built.

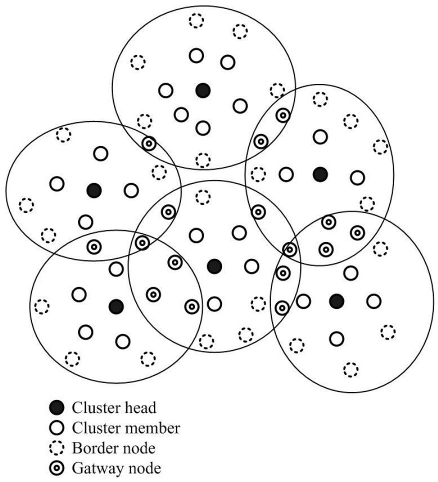

Figure 3. Gateway/border nodes.

3.1. Cluster Organization

We refer to the cluster head selection algorithm proposed in the LEACH protocol [13] to organize the cluster architecture. This selection will be periodically performed such that each sensor node has the equal probability to be a cluster head.

Each iteration of selection of cluster heads is called a round. In each round, a sensor node *s* will choose a random number from 0 and 1. This sensor node *s* will become a cluster head if the chosen number is lower than T(*s*) in Formula (1).

$$T(s) = \begin{cases} \dfrac{P}{1 - P \times (r \bmod \frac{1}{P})} & \text{if } s \in G \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

r denotes the current round number, *P* is the desired percentage of sensor nodes which are cluster heads, and *G* is the set of sensor nodes that have not been cluster heads in the past 1/*P* rounds.

The following descriptions show the steps of cluster organization.

Step 1 (**Cluster advertisements**): Those sensor nodes which become cluster heads will advertise their present to the entire network. All other sensor nodes will collect these advertisements and decide which clusters to join. The joining decision is based on the shortest distance in hop counts to a cluster head. That is, a sensor node will select the nearest cluster head to join. Then all non-cluster-head nodes will send joining messages to their corresponding cluster heads.

Step 2 (**Cluster identifications**): A cluster head after advertising will wait for a period to collect joining messages. From these joining messages, a cluster head can identify which sensor nodes are its cluster members and how far in hop counts these cluster members are away from the cluster head. We define the hop count of the furthest cluster member to be the radius of a cluster. Then each cluster head will advertise its cluster radius to all its cluster members.

Step 3 (**Node identifications**): Each cluster member after receiving any cluster-radius advertisement will identify itself to be a gateway or border node or not. A cluster member is a border node when it receives the cluster-radius advertisement only from one cluster head and its hop count to the cluster head is equal to the cluster radius. A cluster member is a gateway node when it has identified itself as a border node and moreover it receives the cluster-radius advertisement from more than one cluster. A cluster member is called a normal node if it is neither a gateway node nor a border node.

Step 4 (**Neighbor advertisements**): Each sensor node will advertise its node identification (gateway, border, or normal node) to all its one-hop neighbors. Hence, each sensor node can know locally its one-hop neighbors with their identifications and hop counts to the cluster head.

3.2. Route Establishment

In our proposed protocol, both event replication and event search follow the same route establishment. In EDSRT, we obey a priority rule to select the next-hop node to replicate or search one event. Figure 4 shows the detailed flow chart.

At the beginning, the initial node sets the maximal route length to be *n* by using TTL (Time-To-Live). Then, a next-hop node is selected among the one-hop neighbors. The priority of node selection is a gateway node, followed by a border node, followed by a normal node.

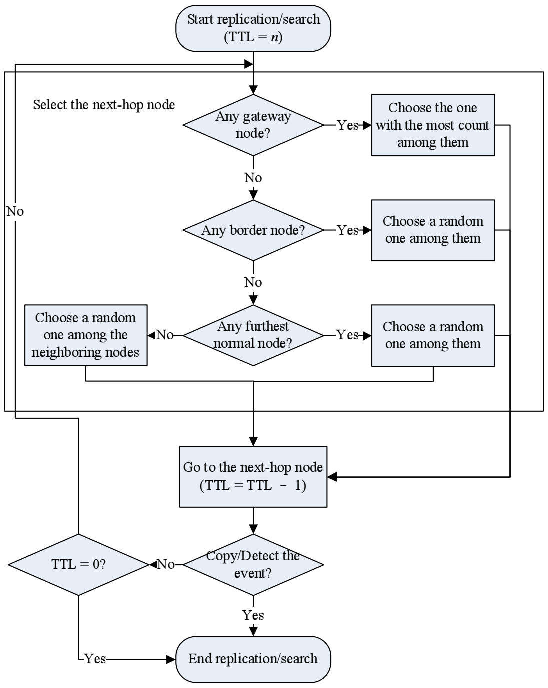
Start replication/search
(TTL = n)
Select the next-hop node
Any gateway node?
Yes
Choose the one with the most count among them
No
Any border node?
Yes
Choose a random one among them
No
Any furthest normal node?
Yes
Choose a random one among them
No
Choose a random one among the neighboring nodes
No
Go to the next-hop node (TTL = TTL – 1)
Copy/Detect the event?
No
TTL = 0?
Yes
Yes
End replication/search

Figure 4. Routing flow chart.

If there has more than one gateway node in the neighborhood, we select the one with the most count on receiving cluster-radius advertisements. That is, each gateway node will maintain a counter and this counter will be increased by one when a cluster-radius advertisement from a different cluster head is received. If the counter has a value *n*, this means that the gateway node is located in the overlapping area of *n* cluster. A gateway node that is located in the overlapping area of more clusters will store more replicated events.

If there is no gateway node but a border node in the neighborhood, we select the border node as the next-hop node. If more than one border node exists, we randomly select one border node among them. We hope to find more gateway nodes through the border node.

If there are no gateway and border nodes in the neighborhood, we select a normal node that is more away from the cluster head than the current node. This selection is based on the comparison of the hop count of each neighboring normal node with that of the current node. We expect a replication path or a search path can toward the borders of a zone where gateway and border nodes can be visited. If all the neighboring normal nodes have the same distance to the cluster head, we choose a random one. To prevent repeated paths occurring, we will not select the previous-hop node as the next-hop node unless there is no other choice.

As mentioned before, we set a TTL value to limit the maximal path length. The path will be extended until TTL is decreased to zero or the desired event is found (i.e., the replication path is intersected with the search path).

4. Simulations

To evaluate the performance of EDSRT, we write simulation programs using NS-2. We simulate an environment with a square space of 100x100 m^2. The sensor nodes are uniformly spread into the environment. We control the number of sensor nodes from 20 to 100. A special node called base station is extra installed to mainly issue search queries.

Each experiment is performed by letting one sensor node replicate an event and letting the base station issue a search query simultaneously. These replication and search processes are continuously performed until the search query matches one replicated event. Then we mainly measure the search latency and energy consumption during the experiment. The search latency is the average time duration elapsed from the moment the search query is issued to the moment one replicated event is found. The energy consumption is the total power consumption on all sensor nodes within one search latency.

We refer to the LEACH protocol to select cluster heads. In our experiments, the number of cluster heads is controlled by setting a percentage. For example, 5% means that 5% sensor nodes are chosen as cluster heads. We evaluate the performance from percentages 3% to 15%. We use the notation EDSRT (c%) in the experimental results to indicate the percentage of cluster heads.

We mainly compare the performance of our proposed protocol with that of rumor routing protocol. We evaluate different rumor routing approaches by increasing the number of parallel search paths. For example, the notation Rumor-c means that there are c parallel search paths and one replication path.

4.1. The Search Latency

Figure 5 shows the search latency of different protocols with different settings.

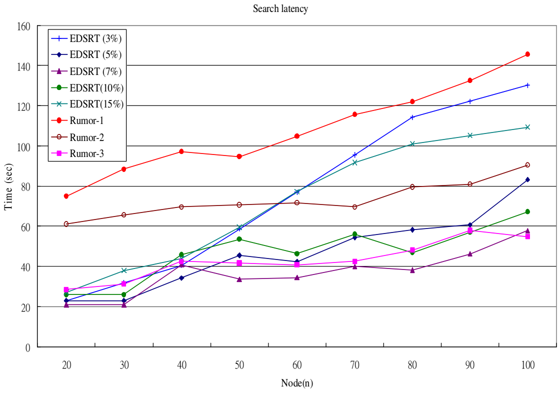

Figure 5. Average search latency.

We found that the search efficiency is affected by the number of gateway and border nodes. As 7% sensor nodes are chosen as cluster heads, the search latency is the lowest. However, when 15% sensor nodes are chosen as cluster heads, the number of gateway and border nodes becomes too large. This will cause the search query to be passed by these nodes several times before matching one replicated event. Hence, the search latency will increase as the number of cluster heads increases.

On the other hand, when 3% sensor nodes are chosen as cluster heads, the number of gateway and border nodes becomes too small. Since our proposed protocol will route along with the gateway node first, then the border node next, and the normal node finally. With the fewer gateway and border nodes, it will often choose the normal nodes, which behaves like random selections. Hence, the performance will decrease as the number of cluster heads decreases.

As can be seen in the figure, with more parallel search paths, the search latency of the rumor routing protocol becomes smaller. Rumor-3 can even compete with EDSRT (7%). However, Rumor-3 will consume lots of energy than EDSRT (7%) as discussed later.

4.2. The Energy Consumption

Figure 6 shows the average energy consumption between different protocols with different settings. We can find that EDSRT saves more energy compared with Rumor. The reason is that EDSRT only spent less time on data search. Thought EDSRT involves the energy consumption on constructing the clusters and identifying the gateway/border nodes, it benefits from the regular search than the random one.

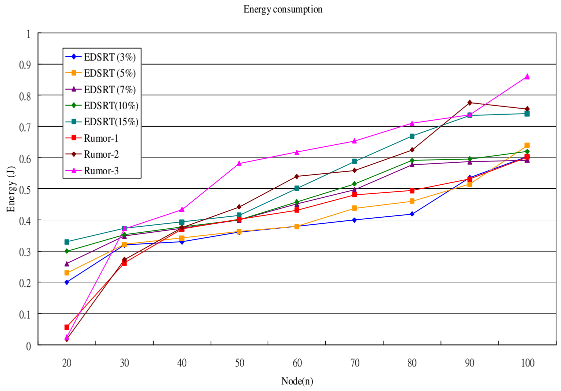

Figure 6. Average energy consumption.

In Rumor, the energy consumption increase as the number of search paths increases. In EDSRT, when the percentage of the cluster head increases, the energy consumption will raise too. This is due to the fact that each cluster head advertises the cluster head information for the formation of its cluster. As the percentage of cluster head increases, the number of cluster head advertisements will increase. Therefore, we will get more gateway and border nodes, and these nodes will advertise their node IDs too.

4.3. The Control Overhead

Figure 7 shows the comparisons of control overhead and route overhead on energy consumption in EDSRT. The control overhead is the cost taken on the cluster formation which includes the transmission cost of advertisements of cluster heads, gateway, and border nodes, and the joining messages of sensor nodes. The route overhead is the cost taken on the building of a replication path and a search path.

This simulation result shows that in EDSRT, the different percentages of cluster heads present different control overheads significantly. The reason is that when the number of cluster heads increases, the cluster organization will waste more energy. Therefore, the control overhead is directly proportional to the percentage of cluster heads.

On the contrary, the route overhead presents a tradeoff with the percentage of cluster heads. EDSRT (7%) has the lowest route overhead. With a percentage lower or higher 7%, the route overhead will be increased as explained in Section 4.1.

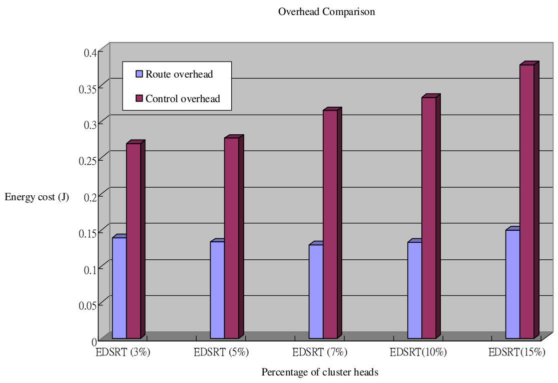

Figure 7. Control overhead.

4.4. The Energy Variance

Figure 8 shows the variance of energy consumption among sensor nodes. If the variance value is large, this means that the total energy consumptions are focused on certain nodes. As can be seen in the figure, EDSRT (7%) has a higher variance value than Rumor-3. The reason is that the specific nodes will be heavily involved in EDSRT, while each node will be uniformly involved in Rumor. Note that we will do the re-selection of cluster heads in EDSRT. This means that every node has an equal chance to be a gateway or border node. This will alleviate the problem of high energy variance.

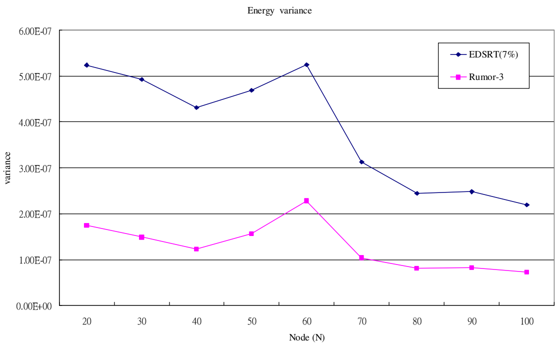

Figure 8. Energy variance.

5. Conclusions

In this paper, we provide a novel routing protocol which includes search and replication strategies to save the energy and time. Our proposed protocol is based on a clustering architecture. The simulation results show that we can speed up the search time and improve the average latency compared to the rumor routing protocol.

Acknowledgment

This research was partially supported by National Science Council of the Republic of China under Contract No. NSC 94-2213-E-259-016.

References

[1] C. F. Chou, J. J. Su and C. Y. Chen, “Straight Line Routing for Wireless Sensor Networks”, *10th IEEE Symposium on Computers and Communications*, pp. 110-115, Jun. 2005.

[2] Y. C. Chang, Z. S. Lin and J. L. Chen, “Cluster Based Self-Organization Management Protocols for Wireless Sensor Networks”, *IEEE Transactions on Consumer Electronics,* Vol. 52, No. 1, pp. 75-80, Feb. 2006.

[3] Q. Fang, F. Zhao and L. Guibas, “Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation”, *4th ACM international symposium on Mobile ad hoc networking & computing,* pp. 165-176, Jun. 2003.

[4] B. Karp and H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks,” *Proceedings of ACM MOBICOM*, pp. 243-254, Aug. 2000.

[5] J. N. A. Karaki and A. E. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey”, *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 6-28, Dec. 2004.

[6] J. N. A. Karaki and A. E. Kamal, “On the Correlated Data Gathering Problem in Wireless Sensor Networks”, *9th International Symposium on Computers and Communication,* Vol. 1, pp. 226-231, Jul. 2004.

[7] J. N. A. Karaki, R. U. Mustafa, A. E. Kamal, “Data aggregation in wireless sensor networks - exact and approximate algorithms”, *High Performance Switching and Routing*, pp. 241-245, 2004.

[8] S. Lindsey, C. S. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems”, *IEEE Aerospace Conference,* Vol. 3, pp. 1125-1130, 2002.

[9] X. Liu, Q, Huang and Y. Zhang, “Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in Large-Scale Sensor Networks”, *2nd international conference on Embedded networked sensor systems*, pp. 122-133, Nov. 2004.

[10] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, “Protocols for Self-Organization of a Wireless Sensor Network”, IEEE Personal Communications, Vol. 7, No. 5, pp. 16-27, Oct. 2000.

[11] J. K. Taek, M. Gerla, V.K. Varma, M. Barton, T.R. Hsing, “Efficient Flooding with Passive Clustering—An Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Networks” *Proceedings of IEEE,* Vol. 91, No. 8, pp. 1210-1220. Aug. 2003.

[12] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, “A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks”, *8th annual international conference on Mobile computing and networking*, pp. 148-159, Sep. 2002.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, „Energy-Efficient Communication Protocol for Wireless Microsensor Networks,“ *Proceedings of HICSS*, pp. 4-7, Jan. 2000.