A Query-aware Routing Algorithm in Sensor networks

Jinbao Li [1, 2] , Jianzhong Li [1, 2] , Shengfei Shi [1]

[1] Harbin Institute of Technology, 150001, Harbin, China
[2] Heilongjiang University, 150080, Harbin, China
lijb6912@vip.sina.com, lijzh@hit.edu.cn

Abstract. Wireless sensor networks have become increasingly popular due to the variety applications in both military and civilian fields. Routing algorithms are critical for enabling the successful operations of sensor networks. A number of routing algorithms have been proposed. However, all the routing algorithms are considered in isolation from the particular communication needs of the data management. This paper focuses on the design of the routing algorithms considering the needs of processing data query in sensor networks. A query-aware routing algorithm is proposed. The algorithm has the following advantages comparing with other routing algorithms. First, it processes as many queries as possible while routing. Second, the broadcast is executed locally so that the energy required by globe broadcasts is saved. Third, routing is executed by searching and generating a binary-tree and only two boundary nodes selected to broadcast message when broadcast is needed so that the number of broadcast is reduced dramatically and the cover range of local broadcast is increased. Finally, multiple routing paths for many routing requirements are found by merging routing requirements and through only one random walk in the sensor network. Experimental results show that the proposed algorithm has better performance and scalibility than other routing algorithms.

1 Introduction

Recent advances in micro-electronics and wireless technologies enable the creation of small, cheap, and smart sensors. In the past few years, smart sensor devices have matured to the point that it is now possible to deploy large, distributed sensor networks in an ad-hoc fashion. These sensors monitor various measurements such as temperature, pressure, humidity, movement, noise level, chemical and etc. Such networks pose new challenges in data processing and dissemination because of their limited resources such as processing ability, bandwidth and energy. Even though each single sensor has limited capabilities, the network consisting of a large number of such sensors are powerful enough to deal with complex monitoring missions. Wireless sensor networks have become increasingly popular due to their variety applications in both military and civilian fields ranging from battlefield surveillance to natural habitat monitoring. Sensor networks are attracting more and more attentions.

Routing is critical for enabling successful operations of sensor networks. Traditional routing schemes are not suitable for sensor networks, so many new routing algorithms have been developed[1, 2, 3, 4, 5].

The directed diffusion routing algorithm is proposed in [1], which provides a mechanism for doing a limited flood of a query toward the event and then setting up reverse gradients to send data back along the best route. This algorithm employs the techniques of the initial low-rate data flooding and gradual reinforcement of better paths to accommodate certain levels of network and sink dynamics. In order to find the best path, this routing algorithm resorts to flooding the query throughout the entire network. Directed diffusion results in high quality paths, but requires an initial flood of the query for exploration.

The Geo-Routing algorithms were considered in [9] and [10]. Geo-Routing algorithms rely on localized nodes, and provide savings over a complete network flood by limiting the flooding to a geographical region, but they do not work without the geography information of sensor nodes.

[2] proposed a random routing algorithm, Rumor Routing. Rumor routing intends to work in conjunction with diffusion, bringing innovations from GRAB[11] and GOSSIP[12] routing to this context. In Rumor Routing, each node maintains a neighbor table and an event table. The event table is generated by an agent. The agent broadcasts an event to the farther nodes and builds up an event table. A query can use the information in its neighbor's table and the event table to form a route path. If the querying node has a path to the sink, then the sink is looked up in this path directly. Otherwise, a neighbor node of it is selected to continue querying. Rumor Routing requires to maintain a neighbor table and an event neighbor table, which consumes too much energy.

[3] described TTDD, a Two-Tier Data Dissemination approach that provides data delivery to multiple mobile sinks. Each data source in TTDD proactively builds a grid structure which enables the mobile sinks to continuously receive data on the move by flooding queries within a local cell only. TTDD handles multiple mobile sinks efficiently, but it is only suitable for sensor networks without mobile sensor nodes.

[5] implemented a cluster based routing algorithm. The basic idea is to divide sensor nodes in a network into some clusters. Each cluster is managed by a head. Routing is executed by the head. In this routing method, not only does the maintenance of clusters require much cost, but also the head may become a bottleneck of the processing information and communication.

[4] provided a routing mechanism to obtain the information in sensor networks, ACQUIRE, which considers each query as an active entity. This entity searches result by transmitting query (in random or other way) in the network. In ACQUIRE, a neighbor table is maintained dynamically at each intermediate node. This table contains the neighbors within *d* hops away from this node. Each active entity can use its neighbors to generate part of query result. When the query result is completely generated, it is returned to the querying node along the reversed path. ACQUIRE generates efficient routes by selecting a proper *d* and refresh the frequency *c*, but it is not suitable for the case with high refresh frequency ($0.08<c<1$).

In summary, all the current routing algorithms, except the one in [4], considered the energy efficiency and extending the network lifetime in isolation from the particular communication needs of the data management and loss the opportunity for crosslayer optimization to design and adapt routing algorithms to the particular routing needs of the data management layer. Although [4] considered the query processing while routing, it is not suitable for the case with high refresh frequency (0.08<c<1).

This paper focuses on the design of the routing algorithms taking account of the needs of processing data query in sensor networks. A query-aware routing algorithm is proposed. The algorithm has the following advantages comparing with other routing algorithms. First, it processes as many queries as possible while routing. Second, the broadcast is executed locally so that the energy required by globe broadcasts is saved. Third, routing is executed by searching and generating a binary-tree and only two boundary nodes selected to broadcast message when broadcast is needed so that the number of broadcast is reduced dramatically and the cover range of local broadcast is increased. Finally, multiple routing paths for many routing requirements are found by merging routing requirements and through only one random walk in the sensor network. Experimental results show that the proposed algorithm has high performance and scalibility than other routing algorithms.

2 Query-aware Routing Algorithm

All nodes in sensor networks are assumed to be homogeneous and uniformly distributed. *S* is the sink and *D* is the source. Routing from *S* to *D* is to find a multi-hop path from *S* to *D*. In this paper, routing not only is considered as a simple path seeking, but also executs queries with routing. We call this kind of routing as query-aware routing. Query-aware routing is considered as searching on a dynamic generated binary tree whose nodes are sensor nodes in the sensor network with *S* as the root and search target as leaves. Since *D* is the search target, *D* must be on a leaf. Routing from *S* to *D* is a procesure of depth first (or width first) search. With an increase in the depth, the binary tree will increase. Once *D* is found, the routing is successful, *D* becomes a leaf node and the multi-hop path is generated. If predefined search depth is reached without finding *D*, the search fails and a failure message is sent to *S*. We first discuss some basic concepts and then give the query-aware routing algorithm DFRS in details.

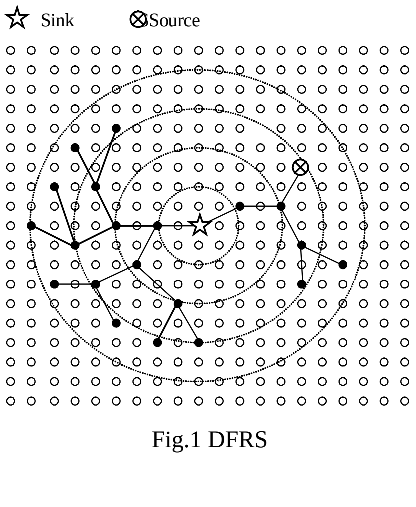

Fig.1 DFRS

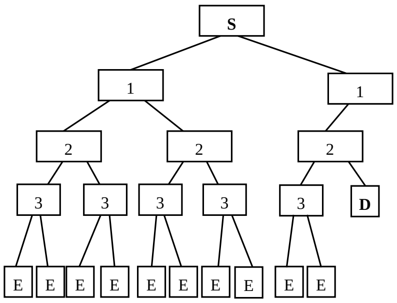

Fig. 2 Routing-Tree

A **Routing Tree** is a binary tree randomly generated for routing. The height of the tree is denoted as *Hop_End* in the rest of the paper. The generation procedure of a routing tree in a sensor network is shown in fig.1. An example routing tree is shown in fig.2.

Sink node *S* is the node which sends routing request to the source node, which is the root of the routing tree. *S* is the start node of a route.

Target node *D* is the destination node of a routing, which is on the leaf of the routing tree.

The minimum routing search depth, *Hop_End*, is determined by the diameter *R* of the sensor network and the effective transmission radius *r*, i.e. *Hop_End = R/r*. In actual applications, the search depth is often more than *R/r*, that is $Hop\_End = (1+\varepsilon)R/r$, where $0<\varepsilon<1$ is a constant.

A boundary node of a node *n* is a node in the node set *B*, where *B*=*C*-*c*, *C* is the node set in the circle with radius *r*, *c* is the node set in the circle with radius *r*-*a*, *r* is the communication radius of *n*, and *a*<*r* is a parameter.

In sensor networks, nodes exchange and transmit messages with their neighbors. A message has *message type* and *message content*. Message type includes *RRQ, RRRQ, FR* and *NT*. The message content of a message of type *RRQ* is composed by *sink node id (SID)*, *target node id (DID)*, *query id (QID)*, *query (Query)*, and *current search depth H*. The message content of a message of type *RRRQ* is composed by *sink node id (SID)*, *query node id (QID)*, and *returned result (Result)*. The message content of a message of type *FR* is composed by *target node id (DID)* and *Query or Result*. A message of type *NT* is composed by *node id (NID)* and *node type (NodeType)*. If *NodeType=boundary*, the corresponding node is a boundary node.

After the sink sends *RRQ* request to its neighbors, the nodes that accept the *RRQ* message process the message content with the following algorithm DFRS. *H* in the *RRQ* is computed by $(1+\varepsilon)R/r$.

```
Algorithm DFRS
Input: RRQ message, RRRQ message, FR message or NT message.
Output: Send the processed RRQ, RRRQ, FR or NT message out.
MessageProcessing(Message) {//function invoked after current node
   m receive message from i
 if (haveMultiMessage())
    Merge the message to sigle message;
 while (Message){
    if (Message.Type == RRRQ) { // Message is a message of RRRQ
       if (Message.Message.SID==m.ID)//m is the node which sends
   the message
         ProcessResult(Message) //process query result.
       else
         Send(Message,i);  //transmit message RRRQ to parent node
    }
   else if (Message.Type == RRQ) {  // Message is RRQ message
      if(m.ID == Message.DID) {  // m is target node
          Result=Process(Message.Query); //process query
          CreateMessage(RMessage);   //construct RRRQ message
          Send(RMessage,i);}   //return query result to parent i
     if(Message.H>0 && HaveBoundaryNode(Message.QID)){
          Message.H--;
          CreateMessage(RMessage); //construct FR message, trans-
   mit RRQ
          if (m.B(l))
             Send(RMessage,m.B(l));  //search  sub-trees  in  left
 first order
        else if (m.B(r))
           Send(RMessage,m.B(r));  //search  sub-trees  in  right first order
         else{//select  two  boundary  points  of  m,  m.B(l)  and m.B(r)
           SelectBoundaryNode(BoundaryNodeBuffer);
           Send(RMessage,m.B(l));}  //  search  sub-trees  in  left first order
     }
     else if (IsBoundary(m)) {  //m is boundary node of i
       CreateMessage(RMessage);  //construct  RM  message:m  is boundary node
           Send(RMessage,i);}//inform parent,m is boundary node
     if(Mesage.H==0){//reach max search depth, search right sub-tree
       Mesage.H++;        //refresh H of RRQ
       UpdateMessage(RMessage);
       Send(Message,i);}
     }
    else if(Message.Type == FR){
      Send(Message.Message,0) //transmit PRQ to neighbors of m
      Timer(T0);} //waiting for result from boundary node
    else if(Message.Type == RM && Message.NodeType == boundary)
      Store(Message.NID)  //store  boundary  node  of  m  to  BoundaryNodeBuffer
   Message = Message->next;
 }
}
```

Fig.1 and Fig. 2 illustrate the routing procedure of DFRS. From the sink *S*, the routing tree is depth first traversed. The traversal of the routing tree is also a procedure of the generation of this tree. When traversing the routing tree, once the target node is found, the traversal is finished when the tree is generated.

During the traversal of the routing tree, if multiple queries can be executed concurrently, the efficiency will be improved. Based on the above idea, the routing requests from other nodes are merged dynamically when executing the routing. That is, once the routing requests are satisfied, these routing requests are merged to the executing routing request. In the following routing process, once the target node of some routing is reached, then the query result is returned while the other routing requests continue being executed until all the target nodes are found or the specified search depth is reached.

In DFRS, the following strategies are used to reduce the number of broadcasts to save energy.

(1). Process as many queries as possible while routing so that the routes are sufficiently used and the required energy is reduced.

(2). Broadcast is executed locally so that the energy required by globe broadcasts is saved.

(3). Transmit message by boundary nodes so that the cover range of local broadcast is increased and the number of repeatedly received messages is decreased.

(4). Routing is executed by searching and generating a binary-tree. Only two boundary nodes selected to broadcast message when broadcast is needed, and thus the number of broadcast is reduced dramatically.

(5). Multiple routing paths for multiple routing requirements are found by merging routing requirements and through one random walk in the sensor network.

3 Experiments and Analysis

In order to test the proposed algorithm DFRS, a simulation environment for sensor networks is built. The number of the nodes in the sensor networks is varied from 1850 to 7400, and all nodes are distributed within an area of *x×750*, where x is from 1000 to 4000. For simplicity, the nodes in the network are uniformly deployed in a grid with size *20×20*.

The following assumptions are used for the experiments: (1) each transmitted message during routing is in one package; (2) it consumes one unit of energy to transmit one package; (2) no energy is consumed when a node receives a package; (4) effective communication radius of all the nodes are 100 units of length; (5) initial energy of each node is 150 units of energy; (6) the target node and sink node of a routing are generated randomly.

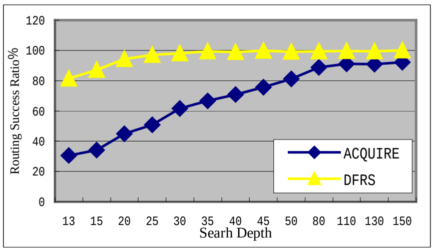

Fig.3 Routing Success Ratio,N=1850

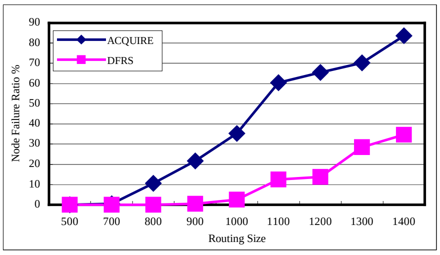

Fig. 4 Node Failure Ratio, N=1850

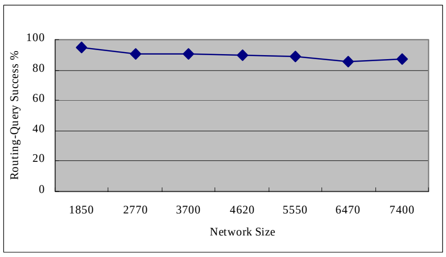

Fig.6 Average Dissipated Energy , the number of routings is 1000

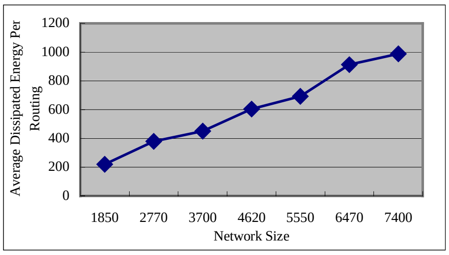

Fig. 5 Routing Success Ratio , the number of routings is 1000

The first experiment is to investigate the success ratios of DFRS and ACQUIRE in a simulation sensor network with *N*=1850 sensor nodes uniformly distributed in an area of 1000×750. DFRS and ACQUIRE are executed in the simulation sensor network. Fig. 3 shows the success ratios of DFRS and ACQUIRE while search depth varied from 13 to 150 without any failure node. From fig.3, it can be seen that DFRS has higher success ratio than ACQUIRE. The success ratio of DFRS is much higher than ACQUIRE in case of the search depth being smaller.

The second experiment is to investigate the ratio of the disabled nodes during routing in a simulation sensor network with *N*=1850 sensor nodes uniformly distributed in an area of 1000×750. Fig 4 shows the ratios of the disabled nodes by running

DFRS and ACQUIRE while increasing the number of routings with same success ratio. Fig. 4 illustrated that the ratio of disabled nodes caused by ACQUIRE higher than that caused by DFRS, and thus DFRS keeps the sensor network having longer lifetime.

The third experiment is to investigate the scalibility of DFRS in a simulation sensor network with the number of sensor nodes varying from 1850 to 7400 and the size of deployment area of the sensor nodes varying from 1000×750 to 4000×750 correspondingly in terms of the success ratio and energy consumption. Fig.5 shows the success ratio of DFRS while the number of sensor nodes varies from 1850 to 7400. The experimental results show that the success ratio of DFRS keeps about 90% while the network size varies. Fig 6 shows that the average energy consumption of DFRS increases linearly when the size of the network varies. The experimental results tell that the scalibility of DFRS is very high.

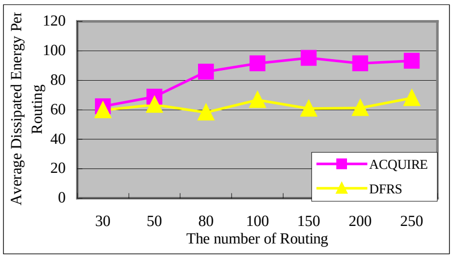

Fig.7 Average Dissipated Energy , N=1000

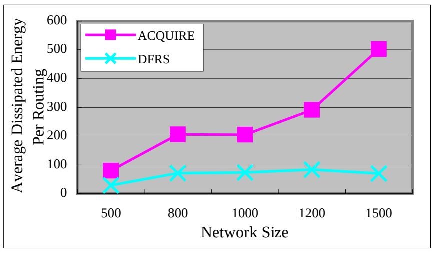

Fig.8 Average Dissipated Energy, the number of routings is 100

The fourth experiment is to investigate the average energy consumption of each routing in a simulation sensor network with *N*=1850 sensor nodes uniformly distributed in an area of 1000×750. Fig.7 shows the average energy consumption of each routing caused by DFRS and ACQUIRE while the number of routing requirements increases. Fig.8 shows the average energy consumption of each routing caused by DFRS and ACQUIRE while the network size increases. These experimental results illustrated that the average energy consumption of each routing caused by DFRS is much lower than that caused by ACQUIRE.

4 Conclusion

A query aware routing algorithm, DFRS, is presented in this paper. DFRS not only considers the energy efficiency and extending the network lifetime but also considers the particular communication needs of the query processing in sensor networks. The experimental results show that DFRS has better performance and scalibility.

References

1. Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva, Directed Diffusion for Wireless Sensor networks, *ACM/IEEE Transactions on Networking*, *11* (1), pp. 2-16, February, 2002.
2. David Braginsky and Deborah Estrin, Rumor Routing Algorithm For Sensor networks, *In Proceedings of the First Workshop on Sensor networkss and Applications (WSNA), September 28 2002*
3. HaiYun Luo, A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor networkss, *ACM MOBICOM (International Conference on Mobile Computing and Networking) 2002*, Atlanta, Georgia, Sep. 2002
4. N. Sadagopan, B. Krishnamachari, A. Helmy, The ACQUIRE Mechanism for Efficient Querying in Sensor networkss, *First IEEE International Workshop on Sensor networks Protocols and Applications (SNPA), in conjunction with IEEE ICC 2003*, May 2003, Anchorage, AK, USA.
5. M. Younis, M. Youssef, K. Arisha, Energy-Aware Routing in Cluster-Based Sensor networks, *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002),* Fort Worth, Texas, October 2002.
6. A. Helmy, S. Garg, P. Pamu, N. Nahata, "Contact Based Architecture for Resource Discovery (CARD) in Large Scale MANets", Third IEEE/ACM International Workshop on Wireless, Mobile and Ad Hoc Networks (WMAN), part of IEEE/ACM IPDPS 2003, April 2003, Nice, France.
7. Lan F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, A Survey on Sensor networks. IEEE Communications Magazine, August 2002.
8. Rajgopal Kannan, Ramaraju Kalidindi, S. S. Iyengar and V. Kumar, Energy and Rate based MAC Protocol for Wireless Sensor Networks, *ACM SIGMOD Record* Vol. 32 No. 4, pp. 60-65 (2003).
9. Yan Yu, Ramesh Govindan and Deborah Estrin.Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.
10. Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Network ing*, pages 243–254, Boston, Mass., USA, August 2000
11. "GRAdient Broadcast: A Robust, Long-lived Large Sensor Network", http://irl.cs.ucla.edu/papers/grab-tech-report.ps
12. M. Lin, K. Marzullo, S. Masini. Gossip versus deterministic flooding: Low message overhead and high reliability for broadcasting on small networks. UCSD Technical Report TR CS99- 0637. http://citeseer.nj.nec.com/278404.html