An Energy Aware Link Failure Management in Wireless Sensor Networks

Sangita Patra
Department of Mathematics
Jadavpur University
Kolkata, India
sangita.pku@gmail.com

Buddhadeb Sau
Department of Mathematics
Jadavpur University
Kolkata, India
bsau@math.jdvu.ac.in

***Abstract*—**Wireless Sensor Networks (WSN) have several in- herent limitations. Consequently, several existing sophisticated routing protocols are not adequate for WSNs. In WSNs, the link failure is an important issue for quality of data communication. In this paper, we propose an energy saving routing technique using a unique path between a pair of nodes. We also propose an efficient distributed technique to find an alternative path when a link fails. The proposed technique (link repairing) always finds an alternative path, if it exists at all. In this repairing, only the nodes in one base cycle are modified. It saves valuable energy of other nodes in the WSN.

***Keywords*—** *Link failure management, spanning tree reconstruction, routing with energy awareness in WSNs.*

_____****_____

I. INTRODUCTION

A WSN consists of a large collection of sensor nodes which have limited computational and communicative capability, and low power resources. Due to the restricted communication range, WSNs use multi-hop data transmission. Reliable communication is important to achieve a certain level of quality of data communication. However, a big challenge in routing protocols of WSN is to achieve maximal robustness against path failure with minimal energy consumption.

Directed diffusion [10], energy aware routing [16] and rumor routing [6], Low-Energy Adaptive Clustering Hierarchy [8] are popularly used for routing in WSNs. Each of these protocols establishes a route on-demand every time when a source requires to communicate to a destination. To setup such a route, they mostly use flooding which drains the valuable energy due to multiple copies of a single message. An already established path between a pair of nodes may be reused later if it suffers, no link failure. It saves huge amount of energy.

If the communication between every pair of source and destination uses unique path, the overall communication will be performed over a spanning tree T of the graph G underlying the network. Fig. 1 shows a spanning tree of the graph

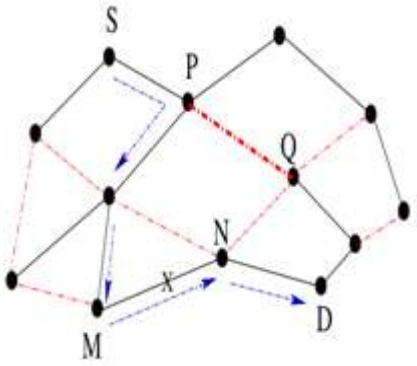

Fig. 1. A graph and its spanning tree

underlying a network. The spanning tree consists of continuous black lines. Suppose we have to send a message from the source node S to the destination node D. Using the spanning tree, a message from S is sent to D through a unique path shown by dotted blue line. If the link M N fails, the message can be sent to the destination through an alternative path. To find such an alternative path, one need not to find a spanning tree from the scratch. We use a base cycle [9])with respect to T containing the failed link to obtain an alternative unique path.

Focus and contribution:

There are many work proposing various types of failure recovery methods using cycles. In [13], they proposed a method which establishes communication tree consisting of communication paths. Where the communication paths are paths from each node to one node on the communication tree. Then, the method produces fundamental tie-sets from a communication tree, and

establishes backup paths from the fundamental tie-sets. When a link failure occurs on a communication tree, the proposed method uses a fundamental tie-set including the failed link.

Our goal is to reconstruct a new spanning tree T_1 considering an edge of G outside T with a minimum modification of T . Using base cycle, we ensure that an alternative spanning tree can be constructed, if it exists at all. Since the nodes only on the base cycle are modified the overhead to find T_1 is minimal. It saves the energy of other nodes. However, in this paper we propose a single link failure management strategy under the distributed environment. Failure management in wireless sensor network itself a novel issue, where as our proposed strategy is fully distributed, which extend the power of novelty. Its extension to multiple link failure is under process.

The rest of this paper is organized as follows. In Section II, we briefly review the related works. Section III describes the routing model and protocol under the distributed environments. It also includes the formal definition of the problem considered in this paper. Section IV describes the construction of an initial spanning tree, recognition of base cycles, and population of routing tables. Section V describes the proposed link failure management strategy. Its performance analysis is presented in Section VI. Section VII contains concluding remarks.

II. Related works

In the literature, there exist several novel protocols [14], [10], [6], [8], [16], [3], [7], [1], [5], [2], [13] addressing different difficulties in ad-hoc networks, though all of these are, not adequate for WSNs. Ad-hoc On-demand Distance Vector Routing (AODV) [14] is used for mobile ad-hoc network where a route is built on-demand. It broadcasts a route request packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In directed diffusion, a source floods the interest message. When a node starts receiving message, a neighbor from which it frequently receives data is included in the route. Sudden change in data sending may result multiple paths between a pair of nodes.

In rumor routing [6], when a source detects an event, it adds the event to its local table and sends an agent on a random walk, instead of flooding the event. When a query is generated from a node it is sent on a random walk until it finds a node that knows the route to the matching event source. The node respond to the query sender by referring its event table. If the path cannot be found, it uses flooding to find it.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [8] is one of the most popular hierarchical routing algorithms for WSNs. It uses local cluster heads as routers. This will save energy since the transmissions will only be done by such cluster heads rather than all nodes. It is only applicable for networks with small geographical size.

The routing protocols described above establishes a route on-demand every time when a source requires a route to a
destination. To setup such a route, they mostly use costly flooding technique. Our goal is to reuse an previously established path between a pair of nodes. Though several works including [16], [17], [11], [4], [18] address the energy awareness in the routing protocols for WSNs, link failure in unique path between a pair of nodes is not considered effectively. Recently, some approaches [15], [12] are proposed to detect path failure and path reconstruction mechanism in WSNs. However, it still requires a rigorous research attention.

In our work, we use spanning tree of the graph for routing. It only consumes energy due to flooding, at the time of initial spanning tree construction. It uses routing table at each node whose size is not greater than that used by the popular routing techniques. We can avoid flooding when a route is required. We can reconstruct the spanning tree, if it at all possible on a link failure, with a minimal modification of the the previous spanning tree. It requires no flooding.

III. Routing model and problem

In the protocols discussed so far, a route is constructed every time when it is required for communication. It is destroyed after a session. Thus, link failure does not much affect the quality of communication. To reduce energy consumption, our main goal is to reuse the previously established paths, and then to avoid flooding for path finding.

A. Routing model

To address the problem of path finding, we describe the routing model considered in this work as follows:

- The graph $G = (V, E)$ underlying the WSN is connected, where V is the set of nodes and E is the set of direct links between a pair of nodes.
- A link is tested on-demand when it is required for communication, instead of costly periodical testing.
- To detect a link failure, a standard technique is used.

- Processes in the nodes run in a distributed manner. Any function in an individual node is atomic.

B. Routing protocol

Each node contains a routing table which contains an entry for each node in the sub-tree with the current node as the root. An entry consists of the link to the neighbor which lie on the unique path from the current node to the destination along the spanning tree T. Note that if the routing table contains no entry for a node, by default its link is directed toward the parent in T. For the time being, we assume that the routing table is already populated accordingly. However, assuming the routing table is complete the strategy is described by an atomic procedure ROUTINGPROTOCOL( ). It starts execution on arrival of a message in the node.

```
1: procedure ROUTINGPROTOCOL( )
2:     Wake on arriving of message.
3:     if (destination node is the current node) then
4:         Stop forwarding.
5:     else if (the link to the destination is alive) then
6:         Forward the message towards the child node.
7:     else          /* the link is failed for communication
                          */
8:         Call LINKREPAIR( ) to handle the link failure
9:     end if
10: end procedure
```

We shortly describe the procedure LINKREPAIR( ) which finds an alternative path along a new spanning tree T_1. T_1 may be obtained from the previous spanning tree T with a minimal modification along a particular cycle.

C. Problem statement

In the proposed routing protocol, we assume that the routing table in every node is populated. Therefore, we address two problems.

1) Given the graph G = (V, E) underlying the WSN, construct a spanning tree T under the distributed environment.
2) Detected a link failure, find alternative path to the failed link reconstructing a new spanning tree T_1 with a minimal modification of T.

To reconstruct a spanning tree we use the concept of base cycle [9]. A cycle may uniquely be identified by the set of edges in it. Any cycle can be constructed by a linear combination of a specific set of linearly independent cycles with respect to symmetric difference operation on cycles (a cycle is represented by the set of its constituent edges). Each of these cycles is called a base cycle (in short BC).

C. Data structure

To hold the information related to the solutions, we use the data types and structures in each node as follows:

Nd={
nbrs: list of node IDs in G,
parent: a node ID in T,
tnodes: list of node IDs in T other than parent,
routeTable: contains a link for every node in the sub-tree,
bcs: list of BCs in which the node belongs to.
}

B
C={
bcID: chord which uniquely identifies a base cycle, next1, next2: two neighbor's link of the corresponding node
}

IV. POPULATION OF NODE INFORMATION

In this section, we describe the method to construct a spanning tree T and base cycles with respect to T. Each node populates relevant information including the routing table. This process is exactly performed once. For this process, we use three types of messages: 1) SPT message for spanning tree; 2) BC message for a base cycle; and 3) RETURN message created by a leaf node or a node second time receiving an SPT message. An SPT consists of the path from the root to the sender. A BC message consists of the destination and the chord which uniquely identifies the base cycle with respect to the spanning tree. A RETURN message consists of the path from the source to the sender.

A. Recognition of spanning tree and base cycles

The process of finding a spanning tree T may be triggered by sending an SPT message from an arbitrary node as the root to all its neighbors. A node receiving the SPT sets the sender as its parent and then recursively generates and sends an SPT message to all its neighbors. If the current node second time receives an SPT message, a base cycle is recognized. The node generates a BC message. It consists of the link between the sender and the current node as the unique identity of the recognized base cycle. The current node then sends the BC message to both the sender of the SPT and the parent. A node

receiving a BC message adds the base cycle into the list bcs with receiving and forwarding links as next1 and next2. If the node first time receives the particular BC message, it forwards the BC message to the parent. Otherwise, it sends a BC message with delete signal to the parent. On receiving a BC message with delete signal, a node deletes the base cycle from the list bcs.

B.Population of routing table

A node generates a RETURN message when it receives an SPT message but it has no link to forward it. If a node second time receives an SPT message, it also generates a RETURN message. In the first case, the RETURN message is sent to parent. In the second case, RETURN message is sent to both parent and the sender of the SPT message. The routing table of each node is populate on receiving an RETURN message. When a node receives an RETURN message, it inserts the receiving link as the link into routeTable for every node of the path in the RETURN message.

V. LINK FAILURE MANAGEMENT

A node can communicate to any one of its neighbors directly. During message transmission, if a link failure occurs then we have to find out another path to send the message toward destination. This process of finding alternative path is called reconstruction. Reconstruction mechanism is an impor- tant issue in WSNs to improve the quality of communication.

A. Reconstruction of spanning tree

To communicate between a source to a destination, we use a spanning tree T of the graph G underlying the network. For a given particular source to destination path is unique. It avoids sending multiple copies of an event detected by the source. A link failure in the path may break the communication. An alternative route is required to continue the communication between the source and destination pair. It is possible if and only if the failed link lies on a base cycle with respect to T . The failed link results a disconnected graph with exactly two components if and only if the fundamental edge cut set [9], (called coset), contains exactly the link. A link failure can be successfully repaired if and only if the coset corresponding to the failed link contains more than one edges. We reconstruct the spanning tree with an arbitrary edge from the coset other than the failed link. Every base cycle contains exactly one chord (which uniquely identifies the base cycle) and all the others are branches of T .

At the time of sending a message from a source to a destination, if a link A fails for communication then we have to check that if the failed link is a part of any cycle of that graph, i.e., if it is on a base cycle. If there exist such a base cycle then the chord F identifying the base cycle must be included to reconstruct a spanning tree T_1 of G − A and accordingly route the message over T_1. In Fig. 2, we show an example of finding an alternative path against a link failure. Continuous

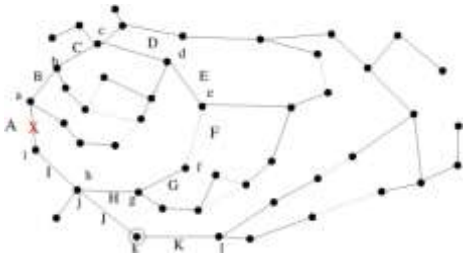

Fig. 2. Showing an alternative path against a link failure

black lines trace a spanning tree, and dotted colored lines are the chord of the graph. During a message passing suppose a link A fails down. To avoid data loss we include the link F , reconstruct a new spanning tree T_1 and update the information in every node of the base cycle identifying by F .

B. Alternative path against a failed link

Consider a path between a pair of source and destination is broken due to a link failure. We explain it with an example as in Fig 2. For convenience, we consider only the subgraph, shown in Fig 3, of Fig 2. An alternative path is reconstructed

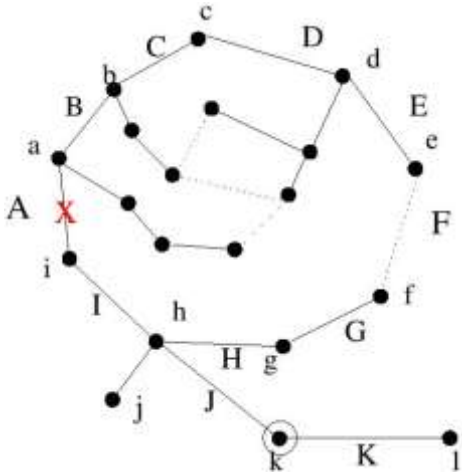

Fig.3.Showing path reconstruction

 ISSN: 2321-8169
464 - 471

with the help of a base cycle. Suppose a source node a requires communication to a destination node k. The routing table of node a is given in following Table I. During message trans- mission suppose node a fails to send message towards i. The failed link is denoted by A. We have to find out alternative path

Destination node	Link
b	B
c	B
d	B
e	B
f	A
g	A
h	A
i	A
j	A

TABLE I. ROUTING TABLE

against A which maintains connectivity between the source and destination reconstructing a spanning tree for energy aware message transmission using unique path of the given network. We have then to find out those BC s in which the failed link A belongs to. We describe the repairing the path against the link failure by an atomic procedure LINKREPAIR( ).

```
1: procedure LINKREPAIR(G)
2:     if (failure link is a branch of any base cycle) then
           /* Reconstruction is possible.
                        */
3:         Call CONSTPATH( )   /* for an alternative path
                        */
4:     else
5:         Report reconstruction is not possible at all.
6:     end if
       /* Construct base cycle for each corresponding
          chord of the spanning tree and find out the
                    alternative path
                        */
7: end procedure
```

The procedure CONSTPATH( ) describes the construction of the alternative path.

```
1: procedure CONSTPATH( )
                    /* Consider two nodes of the chord
                        */
2:     Find those base cycles from BC s which holds
       the failure link as a branch
3:     Select any one bcId arbitrarily which have
       minimum number of links
4:     Chord of the corresponding base cycle of the
       failure link will be the alternative link.
5:     Call UPDATEINFO( )
       /* the information of routing table routeTable
and the information of base cycles BC s */
6: end procedure
```

C. Updating the base cycles

We have to populate the information of each node which are affected by the failure link as well as newly included link. Node a follows its own routing table and list out those nodes whose destination node belongs to the failed link. These nodes are listed in Table II. The node a then finds out those base

Destination node	Link
f	A
g	A
h	A
i	A
j	A
k	A

TABLE I I ROUTING TABLE OF NODE a BELONGS TO THE FAILED LINK

cycles which contain the failed link A from BC s set. We consider an arbitrary base cycle among these. Suppose, the unique chord corresponding to such a base cycle is F . Node a finds that next1 is B and next2 is A with respect to the base cycle with the unique ID F .

Let us assume that Q is the set of BC s in which the failed link belongs to, and P is the set of BC s containing the current link i.e. A. The set of of base cycles which are in the failed link will no longer be a base cycle containing the current link. Therefore, the set P of base cycles of the current node needs to be modified as $P = U\ P - Q \cap P$ which is a symmetric difference of the sets P and Q. Note that if we consider the failed link A for modification of base cycles, Q and P are equal. Thus the set of base cycle containing the link A will be $P = QUQ - Q \cap Q = \varphi$. Using this process, we can find out updated base cycle set bcs for each alive branch of the spanning tree. This process also summarized in the procedure UPDATEINFO( ) along with the modification of the routing table routeTable in a node

D. Updating routing table

A link failure also reefers the modification of the routing table for those nodes which are affected by the failed link. The list of nodes which may detached from

the current node due the link failure is extracted. For routing towards the correspond- ing destination node, it is sent toward each node using the routing table. Each node finds out next1 and next2 with corresponding chord from the selected BC s. Hence next1 is A and next2 is B of the node a corresponding to the chord F . Here next1 is the failed link. Thus node a starts communication through the next2 i.e. B using the routing table. Hence the routing table of that node which is failed to send the message towards the destination will change. The link of those destination node whose link are the failed one i.e. A will change by the next2 i.e.B which is not failed. The updated routing table of the failed node a is given in the following Table III.

Routing table of of other node except the failed node of the selected base cycle will remain same till the router send

Destination node	Link
b	B
c	B
d	B
e	B
f	B
g	B
h	B
i	B
j	B
k	B

TABLE III. UPDATED ROUTING TABLE OF NODE a

the message to that node i.e. e whose next1 or next2 is the chord i.e. F of the selected base cycle from the list of BC s. When the sending message holds by that node whose next1 or next2 is the chord i.e.F of the selected base cycle, routing table from that node i.e. e towards the node whose next1 or next2 failed link i.e. A will be change. In the routing table the link of these destination node according the failed list of the failure node a will change by the selected chord i.e. F for each node.

1: procedure UPDATEINFO( )

/* Failed node follows its own routing table and list out those node whose destination node belongs to the failed link using the routing table. */

/* Now we find out a new set of P where failed link does not belongs to. Let us assume that Q is the set of BC s in which the failed link belongs to, and P is the set of BC s containing the current link */

2: Find set of base cycles from BC s which does not holds the failure link as a branch using the rule of symmetric difference $P = QU P - Q \cap P$

/* Failed node list out those base cycles, containing the failed link from BC s set. */

3: if the current node fails to send message using its existing routing table then

4: Select that next1 or next2 which is not failed. This is the current link for further communication.

5: Update the routing table by change the link of those destination node whose link are the failed one by the selected current link next1 or next2 which is not failed.

6: else if either next1 or next2 of the current node is the chord of the selected BC s then

7: Update In the routing table the link of these destination node according the failed list of the failure node will change by the selected chord for each of these node.

/* routing table from that current node towards the node whose one of the next1 or next2 is the failed link will be change. */

8: else

9: Update routing table of other node of the selected base cycle is unchanged

/* Routing table of other node except the failed node of the selected base cycle will remain same till the router send the message to that node i.e. e whose next1 or next2 is the chord */

10: end if

11: end procedure

E. Correctness of proposed algorithm

Our proposed algorithm is deadlock free. Each node except source node receives the message from its parent node along the path of the spanning tree, at the same time no other node can send or request for message to that two nodes. If any link failed to send the message towards the child node from the parent node, then the parent node check for base cycle of that corresponding failed link exist or not. If base cycle exist then the chord of that base cycle will be the new link to transfer the message towards the destination. Existing spanning tree is now included with a new link to guaranteed delivery the message towards the destination. This mechanism maintain by each node of the spanning tree of the given network for guaranteed delivery of message towards destination. Hence there does not occur any condition like mutual exclusion, hold and wait, no preemption not even

circular wait which are the necessary condition to occur deadlock. Hence our proposed algorithm is deadlock free.

VI. Performance Analysis

Comparison and performance evaluation of different rout- ing algorithms with our proposed algorithm in various aspects is given in the following Table IV.

A. Comparison with existing system

Parameters	Directed diffusion	Rumor routing	LEACH	Proposed algorithm
Network Mode	Flat	Flat	Hierarchical	Flat
Passing messages	Many	Few	Many	Few
Knowledge of locations of neighbors	Does not required	Does not required	Does not required	Does not required
Adaptivity with Dynamic Environment	Almost	A little bit	Almost	Yes
Complete path specified	No	No	No	No
Negotiation	Yes	Yes	No	No
Memory require for each node	High	Proportional to event	High	Proportional to event
Failed path recovery	Yes	No	No	Yes
Network Lifetime	High	Low	High	High
Processing Rate	Low	Low	Low	High
Power Consumption	High	High	High	Low
Robustness	High	Low	High	High
QoS	No	No	No	Yes

TABLE IV. Comparison Table

From this comparison table we can easily see that our proposed approach is more effective and energy award in WSN among all other existing algorithms. As mentioned before, directed diffusion is the most powerful data-centric protocol but it uses flooding to send interest message. Therefore rumor routing protocol is used to solve the problem of flood message to the whole network. When a node generates a query for an event, the nodes that know the route, can respond to the query by referring its event table. Hence, the cost of flooding the whole network is avoided. If the path cannot be found, the application can try resubmitting the query, or at last flooding it. This difficulties are solved in our new proposed algorithm. This new approach does not require extra overhead on flooding which is as memory aware and also energy aware.

That increases the life time of WSN, which is one of the main necessary condition of a WSN.

VII. Conclusion

In this paper we propose an energy aware routing technique which uses an unique path between a source and destination pair. It avoids sending multiple copies of a message and saves energy. We also able to avoid costly flooding when a route is required. A link failure may ceases the data communication. We repair the link failure using base cycle. The proposed can always find an alternative path against a link failure, if the alternative path exists at all. All processing are done under the distributed environment.

However, in this paper we only consider a single link failure. Extending the work to consider multiple link failure is under process.

Acknowledgment: The first author would like to acknowl- edge the Department of of Science and Technology, Govt. of India, for giving supports to carry her research work.

References

[1] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. Ad Hoc Networks, 3(3):325–349, 2005.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. Computer Networks, 38(4):393–422, 2002.

[3] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. Wireless Communications, IEEE, 11(6):6–28, dec.2004.

[4] T. Banka, G. Tandon, and AP. Jayasumana. Zonal rumor routing for wireless sensor networks. In Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, volume 2, pages 562–567 Vol. 2, April 2005.

[5] Z. Bojkovic and B. Bakmaz. A survey on wireless sensor networks deployment. WTOC, 7:1172–1181, December 2008.

[6] David Braginsky and Deborah Estrin. Rumor routing algorthim for sensor networks. In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02, pages 22–31, New York, NY, USA, 2002. ACM.

[7] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. IEEE/ACM Trans. Netw., 12:609–619, August 2004.

[8] M.J. Handy, M. Haase, and D. Timmermann. Low energy adaptive clus- tering hierarchy with deterministic cluster-head selection. In Mobile and Wireless Communications Network, 2002. 4th International Workshop on, pages 368–372, 2002.

[9] F. Harary. Graph Theory. Addison-Wesley, Reading, MA, 1969.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proc. of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000), pages 57–67, Boston, Massachusetts, aug 2000. ACM.

[11] Yunhuai Liu, L.M. Ni, and Minglu Li. A geography-free routing protocol for wireless sensor networks. In High Performance Switching and Routing, 2005. HPSR. 2005 Workshop on, pages 351–355, May 2005.

[12] Zhidan Liu, Zhenjiang Li, Mo Li, Wei Xing, and Dongming Lu. Path reconstruction in dynamic wireless sensor networks using compressive sensing. Ratio, 20(30):40, 2014.

[13] J. Nagano and N. Shinomiya. A failure recovery method based on cycle structure and its verification by openflow. In Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, pages 298–303, March 2013.

[14] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector rout- ing. In Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on, pages 90–100, Feb 1999

[15] TS Raja and R Sujatha. Path failure detection in wireless sensor networks. 2014.

[16] R.C. Shah and J.M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, volume 1, pages 350–355 vol.1, Mar 2002.

[17] Changqing Yin, Shaoyin Huang, Pengcheng Su, and Chuanshan Gao. Secure routing for large-scale wireless sensor networks. In Com- munication Technology Proceedings, 2003. ICCT 2003. International Conference on, volume 2, pages 1282–1286 vol.2, April 2003.

[18] Chang Wu Yu, Rei-Heng Chen, Tung-Kuang Wu, and Fang-Wei Jin. A small-world routing protocol for wireless sensor networks. In Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on, pages 1–4, Oct 2008