A New and Efficient Hierarchy-based Anycast Routing Protocol for Wireless Sensor Networks

Po-Jen Chuang
Department of Electrical Engineering
Tamkang University
Taipei County, Taiwan, R. O. C.
e-mail: pjchuang@ee.tku.edu.tw

Teng-Tai Hu
Department of Network Operation & Maintenance
Chunghwa Telecom Co., Ltd.
Taipei City, Taiwan, R. O. C.
hrdaya@cht.com.tw

Abstract—**This paper presents a Bidirectional Hierarchy-based Anycast Routing (BHAR) protocol for collecting data over multi-hop wireless sensor networks (WSNs). The BHAR protocol improves on existing HAR [1] mechanisms to speed up the process of constructing hierarchical trees and repairing routes, and as a result to prolong network lifetime. Different from HAR, BHAR allows sinks and sources to initialize the construction of a hierarchical tree. By knowing only its own parent and neighbor nodes, each node can join a tree, exchange/refresh its routing table, and perform route repair without geographical information or being controlled remotely. Simulation results show that our BHAR performs apparently better than traditional HAR on network construction and route repair.**

Keywords-Wireless sensor networks; Hierarchy; Routing Protocols; First Declaration Wins; Bidirectional; Location-less

I. INTRODUCTION

Clustering [2] is an effective approach for building and operating communication topologies among WSN nodes. Through data aggregation/fusion and separation between intra/inter-cluster communications, it can save energy consumption due to data transmission and consequently prolong network lifetime.

Cluster construction can be active or passive. For active clustering (such as LEACH [3]), the cluster structure is set up right after node deployment, with all nodes in the radio coverage being included into the topology. Passive clustering (such as PC [4]) is, by contrast, a source-centric based design: When data are sensed and transferred, all nodes that forward this packet will piggyback its neighbor information to construct the cluster structure.

Some clustering approaches (such as LEACH, HIT [5] and MECH [6]) determine the cluster size by calculating the default parameters. Others (such as LLC [7] and HEED [8]) will collect the initial status of the entire network for reference, which may increase the calculation complexity in this phase and delay the cluster construction process. As these protocols usually pick up a cluster head or decide which cluster a node belongs to based on signal strength or communication cost, they must keep listening to and comparing the communication signals of neighboring nodes for a period of time, thus further delaying the cluster construction process. Such protocols generally take “the cluster setup phase” → “the data transfer phase” as a round and operate in cycles. When a transmission failure happens in the network, they will fix it by setting up a new cluster structure. The problem is: Nodes can not transfer data during any cluster construction period because existing failures will not be eliminated until the next round. Besides, frequent cluster reconstruction will consume too much energy. To improve it, some protocols try to accelerate the repair process by sending periodical heartbeat/hello messages to all nodes in order to maintain normal routing. Though able to eliminate possible and imminent routing failures, this may consume excessive energy as all nodes – including the idle ones – have to deal with the periodically sent maintenance packets.

Based on the above observation, this paper presents a new and efficient routing protocol, the Bidirectional Hierarchy-based Anycast Routing (BHAR), to advance the performance of HAR protocols. The distinct designing features of BHAR include (1) allowing both sinks and sources to initialize the construction of a hierarchical tree, and (2) by knowing only its own parent and neighbor nodes, each node can join a tree, exchange/refresh its routing table and perform route repair without geographical information or being controlled remotely. Experimental evaluation shows that when compared with HAR, BHAR can rapidly repair/construct communication routes to facilitate data transmission at reduced energy consumption.

II. BACKGROUND STUDY

A WSN can be symmetric or asymmetric – depending on its communication way. An asymmetric network contains a powerful base station (BS) which can download data or commands to its neighboring nodes (and the nodes can upload data to the BS by multi-hop forwarding). In a symmetric network, the low-power sinks will download commands or data to each node through broadcasting, flooding or routing, and the sensor nodes will send the collected data back to the sinks via the reversed path.

A. *Symmetric Routing Protocols*

Symmetric routing protocols can be flat or cluster-based according to communication structures. Flat protocols are either cost-based or negotiation-based. Cost-based protocols calculate the cost (hop counts, energy consumption, reaction

978-0-7695-4190-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ISPA.2010.86

time, and so on [9]) of each communication path to pick up an appropriate one for transmission. Negotiation-based protocols consider mainly the transmission quality between neighbors when establishing the inter-node transmission paths.

B. *Cluster-based Protocols*

In cluster-based protocols, nodes are organized into various clusters: each is managed by a head and its member nodes will exchange information with the head. A cluster head will transfer messages and data directly or through a hierarchical cluster structure to the BS, to reduce energy dissipation and enhance system lifetime. Usually propagating control messages by flooding, such a protocol may increase data transmission – thus producing extra transmission overhead and degrading efficiency. The communication framework of a clustering protocol, however, can limit the number of forwarding nodes during flooding to reduce the dynamic data flow as well as energy consumption due to routing construction and maintenance.

1) *Active and Passive Clustering*

The setup modes of clusters can be active or passive. For active clustering, nodes will repeatedly broadcast the control data to all neighbors and recursively forward it to the whole network. Though the flooding of the data is restrained through the construction of the cluster structure, it will generate persistent and fixed data overhead in the network. In addition, the change of adjacent statuses caused by the rapid change of node conditions (such as situation shifts, abnormal communication or energy consumption) will also trigger cluster restructuring in the entire network [3].

Passive clustering (or on-demand clustering) can be accomplished without using protocol specific or explicit control packets or signals. Its key idea is to exploit the adjacent information carried by data packets. As passive clustering can be performed without collecting complete adjacent information, it can eliminate the setup latency to reduce the major control overhead of clustering protocols. Meanwhile, using the gateway selection to choose the lowest-cost path also helps save the energy expense of the cluster head [10].

2) *The Hierarchy-based Anycast Routing (HAR)*

The Hierarchy-based Anycast Routing (HAR), a recently proposed routing protocol for WSNs [1], lets the BS initiate tree construction by broadcasting a child exploratory packet to locate the child nodes. A non-member node will determine its parent from the received child exploratory packets: It will first spend a short period of time to collect a number of candidates (kept in a parental candidate table) and then choose a node whose defined metric is the best (such as having the highest received signal strength or the highest remaining energy) to be its parent. HAR can reduce the needed traffic loads when constructing a tree because it avoids using periodical updates and detects failed nodes by the underlying MAC layer protocol. An orphaned node, aware of the absence of its parent, can switch to a new parent immediately by choosing a most appropriate one from its parental candidate table. That is, each node in HAR depends only on the knowledge of its parent, grandparent and parental candidate table.

III. The Proposed Bidirectional Hierarchy-Based Anycast Routing (BHAR) Protocol

A. *Presumptions for Building BHAR*

- BHAR involves no geographical information in deciding transmission paths, i.e., sensor nodes use no positioning systems (such as GPS) or any aspect (angle) measuring functions to help locate their positions.
- As all sensor nodes are assumed to be homogeneous, it is inadequate to use heterogeneous or power-efficient nodes as cluster heads or gateway nodes – because they will consume more energy.
- Our simulation environment allows no direct communication between a sensor node and other remote nodes – because its radio power or coverage is not strong or long enough. Data packets hence must be transferred hop by hop.
- Sensors are assumed to have limited power and the radio transmits by steady power to keep the coverage in a fixed scope if the energy is sufficient.
- To facilitate constructing network topologies, sources can also be the root of a tree to initiate exploring all possible routes to sinks. When a source detects an event that conforms to the predefined situations, it will start instantly to establish an outward transmission routing in order to send the sensed data to the sink. A sink can also actively send out data query packets or add/adjust the to-be-sensed events and build the needed outward transmission routing.

B. *The Operation Steps*

1) In the beginning, a source or sink will broadcast a route exploratory packet to its neighbors in one hop. The message may contain the information of the data to be collected or a task modification asking the receiver node to adjust its data or task. A node receiving such a packet then becomes a source and will send back messages if the data it has sensed satisfy the required task. After receiving the response and realizing the data it requires has been sensed by some nodes, the sink will start to receive these messages.

2) Sources and ordinary nodes will forward the route exploratory packets. They will record relevant information and previous hops (that have forwarded these packets to their routing tables – see TABLE I) as their parent candidates for future route repair. Generally, an orphan or non-member node will select and declare the neighbor node that lies on the shortest path to the sink as its parent. If receiving the same route exploratory packet again, the node will not forward it but drop it.

3) When a non-member node receives a (parent) declaration message, it will become the downstream node of

TABLE I. THE ENTRY FORMAT IN THE ROUTING TABLE OF A NODE

neighbor	IDsrc	hop_count	IDsrc_state
Neighbor ID	Neighbor node's root ID	Hop count through the neighbor to its root	Which root the neighbor belongs to

this tree. If several nodes in the same local area are forwarding route exploratory packets, a node in the transmission range will use the "first declaration wins" principle to decide which tree it belongs to. If a node receives more than one declaration message at the same time, it will decide by the power of the signal. When a node becomes the downstream of a tree, it will forward the route exploratory packet in the same manner – to expand the downstream tree structure – until reaching the destination (a sink or a source). If receiving the same route exploratory packet again, the node will not forward it but drop it. If a node receives different route exploratory packets, it will perform routing integration.

4) Nodes of different trees will interchange their routing information (such as the opposite tree or the hop counts to the root of the opposite tree) and store it in their routing tables – if they are able to communicate with one another. When a parent node compares the content of its routing table and finds out the path length to the other root node in the newly added routing entry is shorter than that in the existing entries, it will upload the information of this shortest path to its parent – recursively – until to its root. If a source has routing entries to other trees in its routing table, it will also forward the received routing information to those trees through all accessible next-hop nodes. When the root of a tree receives routing information from another root, it will store the information in its routing table and look up the table for a shortest transmission path while trying to communicate with the root.

5) If node distribution in a local area is so dense as to exceed the routing table size of a node or to challenge the upper limitation of available memory, the new route is allowed to cover the longest routing record to save memory utilization – on the condition that there exist multiple routes to this specific node. As the possibility to use the longer path is low, deleting the longer path to the same destination will not affect the routing result in most situations.

6) When a neighbor node detects that certain factors (insufficient energy or unexpected obstacles) has caused node failure/communication disconnection, it will delete all routing information related to the failed node and refresh the information to the next-hop nodes able to route to the root of the tree in their routing tables. The children of the failed node will look for new parents from their parent candidates following the order: (a) the member node of the tree it belongs to with the fewest hop counts if its root is a sink, (b) the member node of another tree with shortest hop counts if its root is a sink, (c) the member node of the tree it belongs to with the shortest hop counts if the root is a source, and (d) the member node of another tree with the shortest hop counts if its root is a source. After choosing a new parent, a node adds and refreshes the hop count information to its downstreams by the hop count information of its new parent.

7) If the child of a failed parent is unable to locate a suitable parent candidate, it will send a packet to release the relationship to its downstream nodes, having them clear off their routing tables, become orphan nodes and send out parent requesting packets to search for new parents. A node receiving the parent requesting packet will reply and become the new parent of the sender node – if it is already a tree member. If a sender node receives multiple such replies, it will decide its new parent by the 'first declaration wins' principle.

C. The Designing Features of BHAR

1) Rapid reaction speed: As our tree construction adopts the "first declaration wins' principle, we can assume that rapid reaction speed represents short transmission path in situations without external interferences. Thus, the optimal transmission path for any source/sink pair will get close to their in-between shortest path, which will be an almost straight line.

2) Achieving fast topology stability: Our BHAR protocol allows the source to initiate the process of tree construction. Such a design apparently shortens a node's waiting time to join a tree that is far away from the sink. It also facilitates most of the control packet exchanges among nodes and helps stabilize the network topology in desirably earlier stage.

3) Prolonging the lifetime of nodes: When an intermediate node receives a data packet transmitted from a source to a sink, it can look up from the entries in its routing table for a shortest path (based on the destination of the packet) to decide the next-hop node and the tree to which the packet is to be forwarded. If the data packet must be transferred through one to multiple trees constructed by the source, the transmission path will be decided according to the routing table but not necessarily through the root of each tree. As data packets can be transferred through the shortest path, the lifetime of the root of each intermediate tree can be thus preserved and prolonged.

4) Reducing the impact of topology change: To repair a route, a node prefers to choose a member node whose root is a sink as its parent. The purpose is to adjust the network topology in the meantime when the tree is under repair and make the new topology incline to the sink. If the repairing node is the member of a tree whose root is a sink, it will pick up, in priority, a member of the same tree as its new parent. The same principle is applied to a repairing node whose tree root is a source (and the tree roots of whose possible parents are all sources) to avoid changing the tree it belongs to, i.e., to reduce the impact of topology change as well as the scope of information refreshing ranges.

5) Accelerating route repair without excessive data flooding: If a repairing node realizes there are no parent candidates linkable in its routing table, it will put itself in the linking status to construct a new link relation. This can avoid spending too much time attempting to perform repeated local route repair and yet resulting in unacceptable repaired topologies. Such a local repair mechanism with local route optimization can accelerate the route repair process without flooding a large quantity of control packets to the whole network or reconstructing the whole topology to repair the routes, especially in larger networks or networks with high node density. It can also improve a major problem that confronts HAR: The route repairing node and its children may fail to find their new parents when nodes are sparsely or unevenly distributed in the network.

To sum up, as our BHAR protocol allows a node to perform local route repair and adjustment when its upstream or downstream nodes fail to work, the ratio of (having to send control packets to construct and repair routes) over (the total network communication) will be low. In addition, when a root node fails, its tree members can search again for potential parents in order to join the other appropriate operating trees.

IV. Experimental Evaluation

A. The Simulation Environment

This experimental evaluation simulates and compares the performance of the BHAR and HAR mechanisms on different performance parameters. In the simulation, all nodes are randomly deployed in differently-sized square regions based on evaluation entries. Each node is immobile and will stay fixed during the entire simulation. The locations of all nodes (including sources, sinks and ordinary nodes) in the 1000 simulation environments are created by uniform distribution to evaluate the performance of HAR and BHAR under the given topologies in identical environments. A source can transfer the sensed data to the sink through the built communication routing. In each simulation, the trees can be initiated by the sinks on HAR, while they can also be initiated by sources on BHAR. The adopted performance metrics include:

1) the average waiting time: i.e., the average hop counts between a member node and its root (starting from initialization till joining the tree). It is indeed the time required for an orphan node to become the member of a tree with stable routing table built.

2) the average path length: i.e., the average hop counts required for sending a data packet from a source to its nearest sink – an indicator of how efficiently a routing protocol can utilize network resources to select the shortest path between a source and its nearest sink.

3) network robustness: i.e., the average maximum number of tolerable failed member nodes under which there still exists at least one pair of communication path between a source and a sink. It measures a protocol's ability to repair the routes of member nodes and maintain the network lifetime with the maximum number of existing failed nodes.

B. Simulation Results

1) To check how different numbers of nodes will influence the performance of HAR and BHAR, we respectively deploy 50, 70 and 100 nodes (including 1 to 4 sink nodes) in a 250m×250m square region by the uniform random distribution. Each node has a fixed radio coverage of 50 meters.

a) The average waiting time: As the result in Fig. 1 shows, when the number of sources increases, the average waiting time for an ordinary node to join a tree (i.e., the average hop counts from a member node to its root) shortens for BHAR but stays fixed for HAR. With the same number of sinks, the influence of numbers of nodes becomes negligible.

b) The average path length: In Fig. 2, the average path length will shorten when sinks increase, but the influence of the source or node amount is fairly small.

c) Network robustness: Fig. 3 shows that network robustness grows with increasing numbers of nodes and sources. The growing robustness value will reach a highest point by a specific amount of sources and maintain at that horizon ever since. In this parameter, BHAR always outperforms HAR and the difference between the two becomes stable at the horizontal growing level. The result also exhibits that the higher the sink density is, the faster the growth (of network robustness) will rise up to growing horizontally. In addition, as can be seen in the figure, network robustness grows with the increasing number of

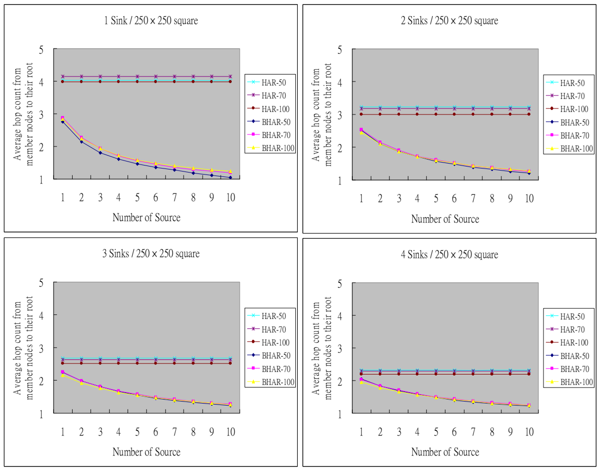

Figure 1. The average waiting time vs. numbers of nodes.

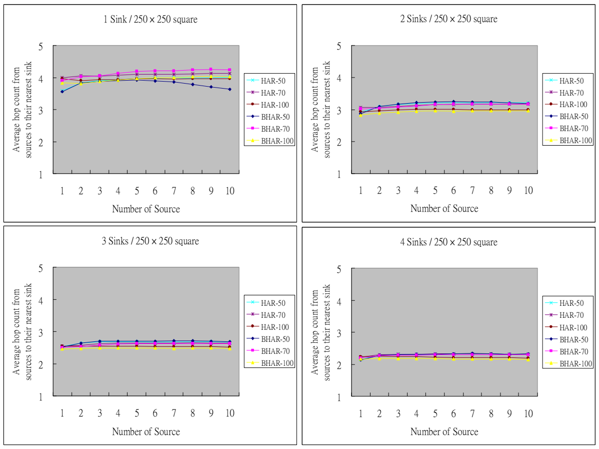

Figure 2. The average path length vs. numbers of nodes.

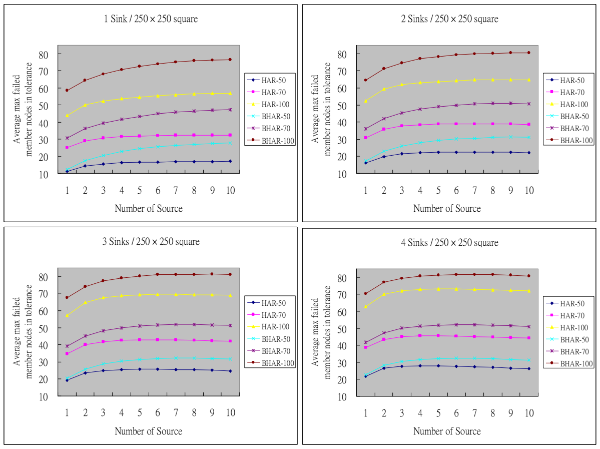

Figure 3. Network robustness vs. numbers of nodes.

sinks.

To sum up, with the same numbers of nodes, BHAR performs better than HAR in the average waiting time and network robustness, while with the same numbers of sinks, both yield quite close performance in the average path length.

2) Network Sizes

To check the impact of network sizes on the protocols, we randomly distribute nodes in the following 4 simulation environments.

- 50 nodes in a 250m×250m square region
- 100 nodes in a 350m×350m square region
- 200 nodes in a 500m×500m square region
- 400 nodes in a 700m×700m square region

(Average 1225~1250 square meters per node)

The previously mentioned configurations are made to keep node density constant over different network sizes. In this simulation, we randomly deploy 1000 groups of node locations by uniform distribution, each node having a stationary radio coverage of 50 meters.

a) The average waiting time: Fig. 4 shows that the average waiting time of member nodes remains fixed for HAR but varies for BHAR. For BHAR, the waiting time that decreases with increasing numbers of sources is lower than HAR. The waiting time gets down when network sizes grow, and when sink density increases, it shortens for both protocols.

b) The average path length: As Fig. 5 exhibits, with identical sink density, both protocols yield quite close average path length in almost all network sizes except 250m×250m with 1 deployed sink, in which the average path length is reduced due to the smaller node number. In such a situation, it is more likely that the source will fail to

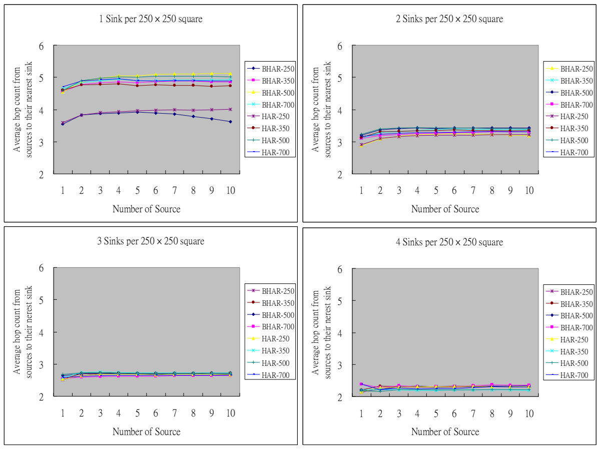

Figure 4. The average waiting time vs. network sizes.

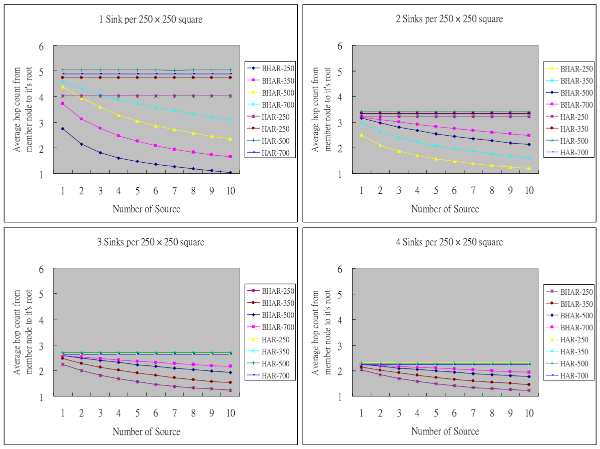

Figure 5. The average path length vs. network sizes.

constitute a path to the sink. On the other hand, both protocols depict shortened average path length when the deploying sink density grows.

c) Network robustness: In Fig. 6, we can see that for both protocols network robustness gets strengthened when network sizes or numbers of sources grow – with BHAR outperforming HAR. Under increased sink density, the strengthened degree rises even faster and the advantage still goes to BHAR – further indicating the proposed protocol's better ability to enhance network robustness.

3) *Sink Density*

To evaluate the impact of different sink density, we deploy different numbers of sinks per 250m×250m square to collect the following results.

a) The average waiting time: Fig. 7 reveals that the average waiting time for BHAR will decrease when the number of sources increases. For HAR, it remains constant without being influenced. When the number of sinks increases, HAR is able to reduce the average waiting time while BHAR is less affected. BHAR nevertheless takes much smaller average hop counts to transmit data from member nodes to the root.

b) The average path length: In Fig. 8, the average path length decreases for both protocols with more sinks but increases under enlarged network sizes (the difference is trivial on the same environment variable).

c) Network robustness: Fig. 9 shows that BHAR generates better network robustness when sources increase. Both the value and the slope of the value grow with network sizes – indicating that under the same node density, BHAR can fortify both network robustness and lifetime in larger networks.

4) *The Inverse Normal Node Distribution*

a) The simulation environment: In this simulation, node locations are settled by the inverse normal distribution

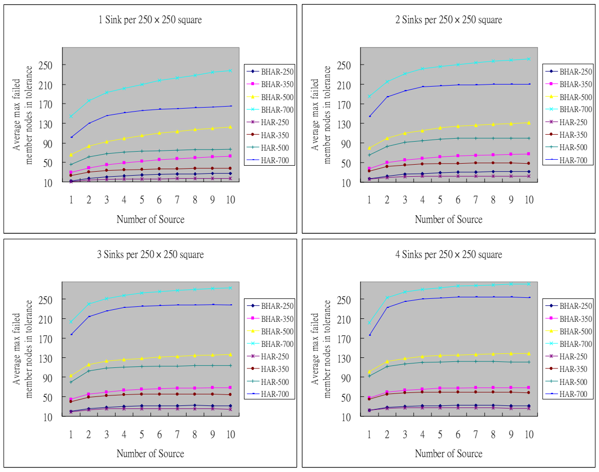

Figure 6. Network robustness vs. network sizes.

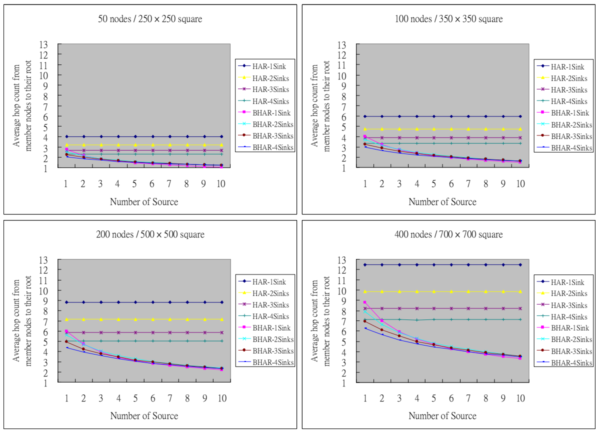

Figure 7. The average waiting time vs. the sink density.

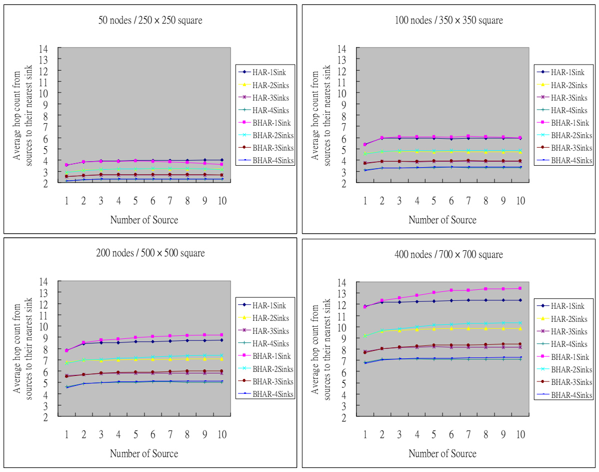

Figure 8. The average path length vs. the sink density.

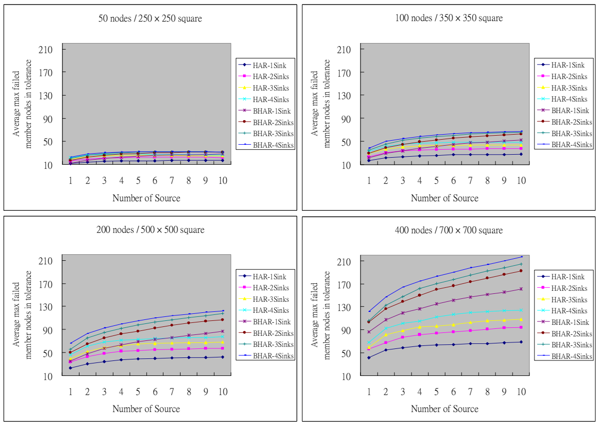

Figure 9. Network robustness vs. the sink density.

to see the impact of this node distribution in different network sizes. Again, 1000 groups of node locations are randomly deployed, each node having a stationary radio coverage of 50 meters. The sink is located on the upper left corner and the source is situated on the lower right corner of the simulation environment, i.e., both are placed at the diagonal locations. To facilitate evaluation on network robustness, we thus double the node density to lift up the success ratio for building a route between the two nodes.

- 100 nodes in a 250m×250m square region
- 200 nodes in a 350m×350m square region
- 400 nodes in a 500m×500m square region
- 800 nodes in a 700m×700m square region

(Average 612.5~625 square meters per node)

The inverse normal node distribution here indicates the distribution curve chart is inversed to the normal distribution. The center of the simulation environment is set as the mean value of the location. The closer the location gets to the mean value, the sparser the node distribution will be or the closer the location is to the corner, the denser the node distribution will be. The standard deviation value is set to 1/3.5 of the side length, to make generating the route between the sink and the source through the node near the border possible. These configurations are made to test if the two protocols can adjust the routes away from the center area of the network – so as to maintain network robustness under such a node distribution when there are fewer operable nodes near the center due to high utilization, power exhaustion or any other reasons.

b) The results: With one pair of a source and a sink is located on the diagonal positions of a network, the proposed BHAR still performs better in the average waiting time than HAR, as Fig. 10 displays. In Fig. 11, both protocols produce quite similar performance in the average path length. Fig. 12 shows that under the same network size, our BHAR outperforms HAR in network robustness, indicating that BHAR can sustain more node failures and maintain high connectivity even when node distribution is sparser around the center of a network. This is a strong fact to support that BHAR indeed acts more desirably in repairing routes than HAR.

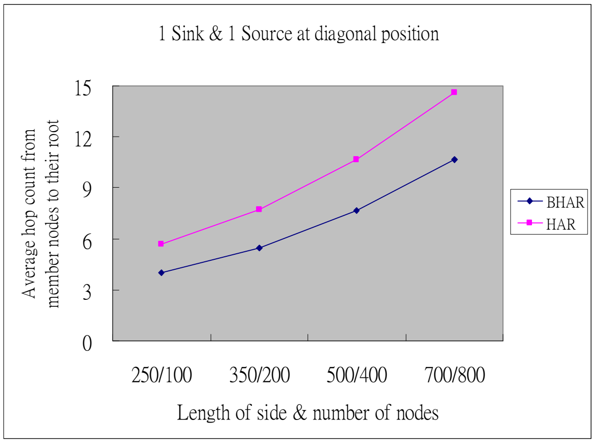

Figure 10. The average waiting time under inverse normal node distribution.

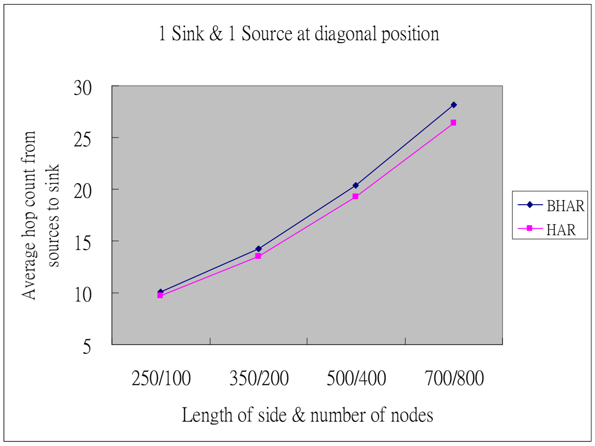

Figure 11. The average path length under inverse normal node distribution.

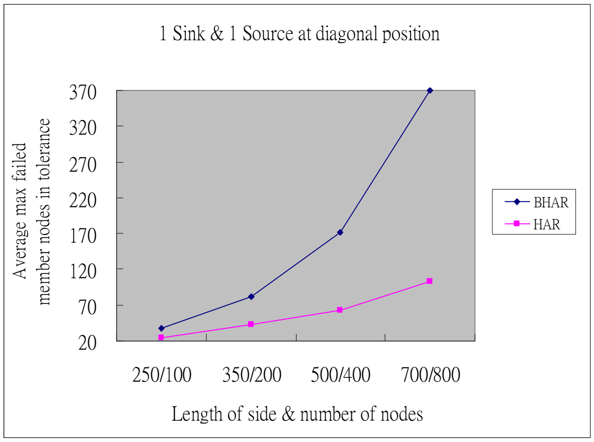

Figure 12. Network robustness under inverse normal node distribution.

V. Conclusion

This paper presents Bidirectional Hierarchy-based Anycast Routing (BHAR), a new and efficient routing protocol for WSNs. With its distinct features, BHAR is shown through simulation to outperform existing HAR protocols in most situations. For instance, BHAR can shorten the average waiting time for normal nodes to join a tree in a network with increased sinks or sources. It also attains better scalability as network sizes cast fairly small impact on its efficiency (very critical for maintaining good performance of WSNs). In all, BHAR works more efficiently than HAR in constructing network topologies and repairing routes: It can effectively and efficiently construct the needed communication routes to facilitate data transmission.

References

[1] N. Thepvilojanapong, Y. Tobe, and K. Sezaki, “HAR: Hierarchy-Based Anycast Routing Protocol for Wireless Sensor Networks,” Proc. IEEE Symp. on Applications and the Internet, Jan. 2005, pp. 204-212.

[2] P.-J. Chuang, S.-H. Yang, and C.-S. Lin, “An Energy-Efficient Clustering Approach for Wireless Sensor Networks,” Special Issue: Int’l Conf. on Algorithms and Architectures for Parallel Processing 2009, Journal of The Chinese Institute of Engineers, vol. 32, no. 7, pp. 951-957, Nov. 2009.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocols for Wireless Microsensor Networks,” Proc. 33rd Annual Hawaiian Int’l Conf. on Systems Sciences, Jan. 2000, vol. 2.

[4] T. J. Kwon, M. Gerla, V. K. Varma, M. Barton, and T. R. Shing, “Efficient Flooding with Passive Clustering - An Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Networks,” Proc. of the IEEE, vol. 91, no. 8, pp.1210-1220, Aug. 2003.

[5] B. J. Culpepper, L. Dung, and M. Moh, “Design and Analysis of Hybrid Indirect Transmissions (HIT) for Data Gathering in Wireless Micro Sensor Networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, pp. 61-83, Jan. 2004.

[6] R. S. Chang and C. J. Kuo, “An Energy Efficient Routing Mechanism for Wireless Sensor Networks,” Proc. 20th Int’l Conf. on Advanced Information Networking and Applications, June 2006, vol. 2, pp. 308-312.

[7] M. Khan, B. Bhargava, and L. Lilien, “Self-configuring Clusters, Data Aggregation, and Authentication in Microsensor Networks,” Technical Report CSD-TR03-005, Department of Computer Sciences, Purdue University, West Lafayette, Indiana, Aug. 2003.

[8] O. Younis and S. Fahmy, “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks,” IEEE Trans. on Mobile Computing, vol 3, no 4, pp. 366-379, Oct-Dec 2004.

[9] F. Ye, A. Chen, S. Lu, and L. Zhang, "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks," Proc. IEEE 10th Int’l Conf. on Computer Communications and Networks, Oct. 2001, pp. 304-309.

[10] M. Gerla, T. J. Kwon, and G. Pei, “On Demand Routing in large Ad Hoc Wireless Networks with Passive Clustering,” Proc. IEEE Wireless Communications and Networking Conf., Sep. 2000, vol. 1, pp. 100-105.