A Layer-Based Routing Protocol for Heterogeneous Wireless Sensor Networks

Xiao Chen[1], Zanxun Dai[1], Wenzhong Li[2], Hongchi Shi[1]
[1]Department of Computer Science, Texas State University, San Marcos, TX, USA
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: xc10@txstate.edu, zd1020@txstate.edu, lwz@dislab.nju.edu.cn, hs15@txstate.edu

***Abstract*—Due to requirements in different applications, sensors with various capacities are deployed. Data routing in such heterogeneous wireless sensor networks (HWSNs) poses challenges. First, the heterogeneous features create asymmetric links in the communication graphs which are not dealt with by conventional routing algorithms using undirected graphs. Second, it is important to provide an assured delivery rate for mission critical applications since sensors communicate with each other through lossy asymmetric links. In this paper, we propose *LayHet*: a layer-based routing protocol for HWSNs, which takes advantage of the asymmetric links to deliver messages to the sink with an assured delivery rate. Simulation results show that LayHet outperforms previous routing methods in terms of average delivery ratio, average hops, average packet replication and average control message overhead.**

***Index Terms*—asymmetric link, heterogeneous wireless sensor networks, layer-based, routing**

I. Introduction

Most research on wireless sensor networks (WSNs) considers homogeneous sensor networks where sensors have the same capabilities in terms of computation, communication, memory, power supply, reliability, etc. However, it is reported in [10] that when properly deployed, heterogeneity can triple the average delivery rate and provide a five-fold increase in the network lifetime. Therefore, in this paper, we study the routing protocols in *heterogeneous wireless sensor networks* (HWSNs) to deliver the data collected by the sensors to the sink. Here we just take the sensors' diverse transmission ranges brought about by their heterogeneity into account.

Because of different transmission ranges, there will be asymmetric links in the communication graph. For example, sensor A can reach sensor B, but B may not reach A. Therefore, the routing protocols developed for regular WSNs may not be directly used in HWSNs or may incur higher overhead [8]. The challenge in designing routing protocols for HWSNs is how to design distributed and energy-efficient ones by taking advantage of the asymmetric links to achieve an assured delivery rate.

We designed our first routing protocol *ProHet* for HWSNs in [4], which adopted a probabilistic data forwarding scheme. The drawback of ProHet is that it looks for route in each hop, which may incur a lot of message replication and control message overhead. In this paper, we propose a layer-based protocol called *LayHet* to overcome this disadvantage. At the beginning, LayHet assigns a layer number relative to the sink to each node which will be used to steer the routing in the right direction in each hop later. Similar to ProHet, LayHet takes advantage of the asymmetric links to deliver messages to the sink with an assured delivery rate. There are two parts in the LayHet protocol: the preparation part and the routing part. The preparation part includes finding the reverse path for each asymmetric link, the initialization of nodes' layer numbers, and the periodic adjustment of layer numbers. The routing part includes the sender broadcasting H times to guarantee an assured delivery rate, each receiver calculating its probability to forward the message to reduce the number of replicated messages and the updating of packet loss rates of links.

Specifically, we make the following contributions to deal with the heterogeneity, reliability and scalability issues:

- We propose a layer-based routing scheme in HWSNs which allows communication between node pairs by finding reverse paths for asymmetric links.
- LayHet is a distributed localized routing strategy where each node only records the information of its lower layer In-out-neighbors or Out-neighbors and node layer numbers and packet loss rates of links are updated locally.
- We present a theoretical analysis to our scheme which adopts a probabilistic forwarding strategy to achieve assured delivery rate with low message overhead.
- Last but not the least, LayHet is mobility compatible and suitable for mobile HWSNs. Specifically, the movement of nodes only causes local information update.

The rest of the paper is organized as follows: Section II references the related work. Section III presents the preliminaries. Section IV proposes the LayHet protocol. Section V does the analysis. Section VI shows the simulation results. And conclusion is in Section VII.

II. Related work

There are many routing algorithms [3], [5], [6], [7], [9], [11] available for homogeneous sensor networks in the literature, where all the sensors are the same in terms of their sensing and transmission ranges, battery power, etc. However, there are only a few routing protocols available for HWSNs [1], [2], [12], [13]. These routing algorithms are coarse grained in that they roughly divide sensors into two categories: powerful ones and less powerful ones. The sensors are organized into clusters using different algorithms with the powerful ones as cluster heads and less powerful ones as cluster members. The

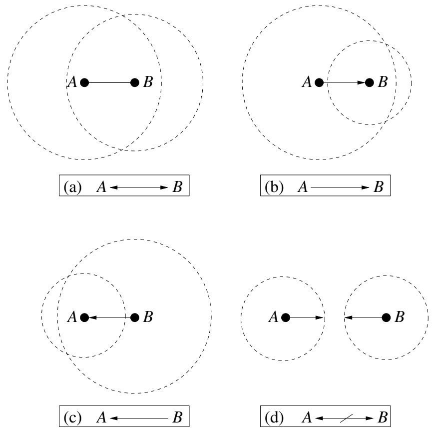


Fig. 1. The neighbor relationships between two nodes A and B. (a) A and B are each other's In-out-neighbor; (b) A is the In-neighbor of B and B is the Out-neighbor of A; (c) B is the In-neighbor of A and A is the Out-neighbor of B; (d) A and B are non-neighbors

routing protocol is hierarchical: the intracluster routing is used by the cluster members to send messages to each other and to the cluster head, and the intercluster routing is used by the cluster heads to send messages to each other and eventually to the sink. In these kinds of routing protocols, the capability of each individual sensor is not distinguished and the asymmetric links are not fully utilized. Therefore, we proposed a routing protocol called ProHet in [4] to take these into account. The drawback of ProHet is that it may incur a lot of overhead due to searching for route in each hop. In this paper, our goal is to propose a new routing protocol to overcome the drawback of ProHet.

III. Preliminaries

An HWSN can be represented by a directed graph $G = \{V, E\}$, where V is the set of sensors (also called nodes), and E is the set of links (also called edges) in the network. For example, if sensor B is in the transmission range of sensor A, then there is a directed link from A to B. We assume graph G generated from the HWSN is a strongly-connected directed graph. Therefore, the HWSN is also strongly-connected.

We categorize the neighbor relationships of sensors into four categories: (1) In-out-neighbor; (2) In-neighbor; (3) Out-neighbor; and (4) Non-neighbor. For two nodes A and B, as shown in Fig. 1, if $A \to B$ and $B \to A$, then A and B are In-out-neighbors of each other. If only $A \to B$ (or $B \to A$), then A (or B) is the In-neighbor of B (or A), and B (or A) is the Out-neighbor of A (or B). If neither $A \to B$ nor $B \to A$, they are non-neighbors of each other.

We assume data is transmitted through lossy links. The packet loss rate of a link uv is defined as 1 minus the ratio of the number of packets N_d which are successfully received by node v to the total number of packets N_s sent by u. That is,

$$P_{lossrate} = 1 - N_d/N_s \tag{1}$$

IV. LayHet Routing Protocol

In this section, we present the LayHet protocol which has two parts: The preparation part which includes finding the reverse paths for asymmetric links, assigning layer numbers to the nodes and adjusting the layer numbers periodically; And the routing part which includes the sender broadcasting H times to guarantee an assured delivery rate, each receiver calculating its probability to forward the message to reduce the number of replicated messages, and the updating of packet loss rates of links. The details are as follows:

A. Preparation

1) Finding a reverse path for each asymmetric link: We use the same algorithm as in [4], which is omitted here because of space limitations, to establish a reverse path from a node to each of its In-neighbors. The main idea of finding a reverse path from a node to its In-neighbor is using message broadcasting with the maximum reverse path length setting to three, which can lead to a successful rate of 97% [8].

2) Deciding initial layer numbers: In this part of the LayHet protocol (see Algorithm DILN), each node will find out its layer number which represents its shortest hop count to the sink. First a node u broadcasts an exploration packet EP containing a hop count c initialized to 0 and its ID to the sink. On the way, the hop count is incremented and the path is recorded. After the sink receives EP, it waits for a while for more copies of EP to arrive. Then it picks the EP with the smallest hop count. This is because multiple EPs can arrive at the sink due to the nature of broadcast. The smallest hop count represents the shortest hop distance from u to the sink. The sink increments the smallest hop count by 1, which is the final hop count c from u to the sink. Then the sink sends back an ACK of EP containing c to u via all the forwarding nodes on the path. Because of the asymmetric links, the reverse paths may be used in the process. After u receives c, it knows its layer number to the sink is c. A good point of the DILN algorithm is that each node may have multiple chances to adjust its initial layer number: Once by the ACK from the sink addressing itself, other times by the ACKs from the sink addressing other nodes if it is the relay node on the paths. Multiple adjustments are necessary because of the lossy links. The closer the node to the sink, the more accurate its layer number can be because it is more likely to be a relay node and thus has more chances to adjust its layer number. The accuracy of the layer numbers of lower layer nodes is more important than that of the higher layer nodes because lower layer nodes are more likely to relay messages for others.

3) Adjusting layer numbers periodically: After applying Algorithm DILN, because of the lossy links, some nodes may not be put into the right layers. But they still have chances to adjust their layer numbers later. To reduce the overhead, the adjustment of layer numbers can be embedded in Algorithm UPR-P in Section IV-B. When a node u communicates with its In-out-neighbor or Out-neighbor v to find out the packet loss rate of link uv, besides sending back the number of messages that v receives, v will also send back its layer number. If u's layer number is at least 2 more than v's layer number, u will adjust its layer number to v's layer number plus 1.

Algorithm DILN: Deciding Initial Layer Numbers

```
Node u broadcasts an exploration packet EP containing a hop count c = 0 and its ID.
if a node v receives EP then
    if it is the sink node then
        it waits for a while for more copies of EP to arrive. Then it picks an EP with the smallest hop count. It increments the hop count by 1 and generates an acknowledgement ACK containing the value of the current hop count c and the path to all the forwarding nodes on the path back to the source u. The later arrived copies of EP are dropped.
        When an intermediate node m on the path receives ACK, it adjusts its own layer number according to hop count c and its location on the path.
        if its previous node t is its In-out-neighbor then
            it sends ACK directly to t;
        else if m has a reverse path to t then
            m sends ACK to t via the reverse path of the asymmetric link t → m;
        else
            m simply drops ACK
        end if
    else
        it increments the hop count by 1, appends its ID to EP and rebroadcasts EP
    end if
end if
After u receives ACK, it knows its layer number to the sink is c.
```

B. Routing

The routing part of LayHet contains three phases: Broadcasting H times, Forwarding messages, and Updating packet loss rate periodically (see Algorithms BRD-H, FWD-M and UPR-P, respectively). The assured delivery rate is preset to Δ. In Algorithm BRD-H, before any routing in the network begins, the packet loss rates of the links between a source node u and its K lower layer In-out-neighbors or Out-neighbors are generated randomly because the network does not have any routing history. Later the packet loss rates are updated by Algorithm UPR-P. After u knows the packet loss rates of the links, it broadcasts the message it wants to send to the sink H times so that at least one of its K lower layer In-out-neighbors or Out-neighbors can receive the message in order to achieve the assured delivery rate Δ. Next in the FWD-M algorithm, a receiving node v will forward the message at a probability of Γ to avoid flooding the network with unnecessary messages. The formulas to calculate H and Γ are presented in Section V. The UPR-P algorithm updates the link packet loss rates periodically so that the next routing can be guided by more accurate information in the network.

We use an example in Fig. 2 to explain the LayHet protocol. Each black dot represents a sensor which is responsible for collecting data. The right most node is the sink which is responsible for processing data after collection.

In the beginning, each node applies Algorithm DILN to determine its layer number to the sink. After the initialization, the nodes are put into different layers relative to the sink. Because of the lossy links in the network, some nodes may not be put into the right layers. But the nodes can use Algorithm UPR-P to adjust their layers later.

After the node layers are identified, routing can be carried out. Suppose a source node u in Layer L_5 wants to send a message to the sink. It has K In-out-neighbors or Out-neighbors in the lower one-hop, two-hop and three-hop layers. A node may have a one-hop In-out-neighbor or Out-neighbor in the lower two or three-hop layers because we consider opportunistic communications exploiting the nature of broadcast.

Algorithm BRD-H: Broadcasting H Times

```
Source node u finds out the packet loss rates p_1, p_2, ···, p_K with its K lower layer In-out-neighbors or Out-neighbors. Before any routing in the network starts, the packet loss rates are generated randomly. Later, they are updated by Algorithm UPR-P periodically.
Node u calculates the number of times H it should broadcast using formula (2) in Section V.
Node u broadcasts the message plus its link packet loss rates p_1, p_2, ···, p_K H times.
```

Algorithm FWD-M: Forwarding Messages

```
repeat
    If a node v receives a message from a higher layer neighbor u along with the packet loss rates of u's links, it uses formula (3) in Section V to decide its probability Γ to forward the message.
    If it forwards, it becomes the new source and reapplies the BRD-H algorithm.
    If it does not forward, it will simply drop the message.
until the message reaches the sink.
```

Algorithm UPR-P: Updating Packet Loss Rate Periodically

```
Each node u updates the packet loss rate of each of its links with its K lower layer In-out-neighbors or Out-neighbors every T time period.
Suppose node u sends out N_s messages to node v during T time period. At the end of T, node u sends a message to v asking "How many messages out of N_s have you received?".
After v receives the inquiry, it replies directly or through the reverse path with the answer "N_d". Also it attaches to the message its layer number for u to adjust its layer number.
After u receives the answer, it updates the packet loss rate of link uv to 1 − N_d/N_s. Also u may adjust its layer number based on v's layer number.
```

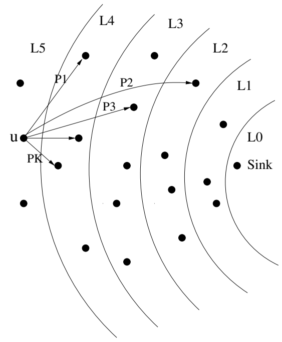


Fig. 2. A data forwarding scenario

Based on the packet loss rates of links to these neighbors, u broadcasts the message H times calculated by formula (2) in Section V. This guarantees that at least one of these neighbors will receive the message with a high probability. Then each of these receivers will decide the probability Γ to forward the message by formula (3) in Section V. The purpose of the forwarding probability is to avoid flooding the network with replicated packets. If a node chooses to forward, it becomes the new source and reapplies the routing protocol. Every T period of time, a sender will update the packet loss rates of its links so that the calculations of H and Γ can be more accurate next time.

V. Analysis

In this section, we provide an analysis of LayHet to show that if H and Γ are properly selected, there is a high chance that the routing can achieve an assured delivery rate Δ and reduce the number of replicated messages in the network. In order to explain that, it is easier to do it reversely, that is: Given the assured delivery rate Δ, decide the number of broadcasts H and the forwarding probability Γ to meet the assured delivery rate Δ and reduce the number of replicated messages.

For a node u at layer i, it forwards data to the lower layer nodes by broadcasting. Node u may need to broadcast several times so that at least one node in the lower layers receives the message. The number of broadcasts H depends on the link qualities from u to its lower layer In-out-neighbors or Out-neighbors. When a lower layer node v receives the message from u, it will forward the message with probability Γ, or in other words, with probability $1-\Gamma$, it will drop the message.

If node u in layer i in Fig. 2 needs to send a message to the sink. In the worst case, the message needs to travel i hops to reach the sink. Assume transmission in each layer is the same. To guarantee the overall assured delivery rate Δ, in each layer, we should guarantee the success rate of the transmission to be at least $\Delta^{\frac{1}{i}}$. We first decide the number of broadcasts H to satisfy the requirement. We assume the packet loss rate of the link from u to its j-th In-out-neighbor or Out-neighbor is p_j. A transmission is successful if at least one of the lower layer nodes receives the message. The probability is:

$$\begin{aligned}&Pr\{at\ least\ one\ lower\ layer\ node\ receives\ the\\&message\ after\ H\ transmissions\}\\=\ &1-(\prod_{j=1}^{K}p_j)^H\end{aligned}$$

$$\text{Let}\qquad 1-(\prod_{j=1}^{K}p_j)^H \geq \Delta^{\frac{1}{i}}$$

$$\text{We have}\qquad H \geq \frac{ln(1-\Delta^{\frac{1}{i}})}{\sum_{j=1}^{K} ln(p_j)} \tag{2}$$

After node u broadcasts the message H times, the message is transmitted to one or more lower layer nodes with high probability. To reduce the number of replicated messages, not all the nodes receiving the message will forward the message to a lower layer. A receiver only forwards the message with probability Γ. Given that a message has been received by some lower layer nodes, we should make sure that at least one node will forward the message. That probability is:

$$\begin{aligned}&Pr\{at\ least\ one\ lower\ layer\ node\ will\ forward\\&the\ message\}\\=\ &1-Pr\{no\ one\ will\ forward\ the\ message\}\\=\ &1-\prod_{j=1}^{K}(Pr\{the\ jth\ node\ does\ not\ receive\ the\\&message\}+Pr\{the\ jth\ node\ receives\ the\\&message\}\cdot Pr\{the\ jth\ node\ doesn't\ forward\\&the\ message\})\\=\ &1-\prod_{j=1}^{K}(p_j^H+(1-p_j^H)(1-\Gamma))\\=\ &1-\prod_{j=1}^{K}(1-\Gamma+p_j^H\Gamma)\end{aligned}$$

$$\text{Let}\qquad 1-\prod_{j=1}^{K}(1-\Gamma+p_j^H\Gamma) \geq \Delta^{\frac{1}{i}}$$

Then, $1-\Delta^{\frac{1}{i}} \geq \prod_{j=1}^{K}(1-\Gamma+p_j^H\Gamma) \geq (1-\Gamma+p_{min}^H\Gamma)^K$,

in which p_{min} is the minimum value of $p_j, (1 \leq j \leq K)$. Solving this inequality yields

$$\Gamma \geq \frac{1-(1-\Delta^{\frac{1}{i}})^{\frac{1}{k}}}{1-p_{min}^H} \tag{3}$$

Since H and Γ are obtained based on our preset value Δ, it is guaranteed that our routing algorithm LayHet can deliver messages with an assured delivery rate Δ and reduce the number of replicated messages.

VI. Simulation

We compare the performance of LayHet with our previous protocol ProHet in [4] using a self-written simulator in Java language. We only compare LayHet with ProHet because ProHet achieves the best performance in dealing with asymmetric links and has been proved in [4] to outperform other routing schemes in HWSNs.

The following metrics are used:

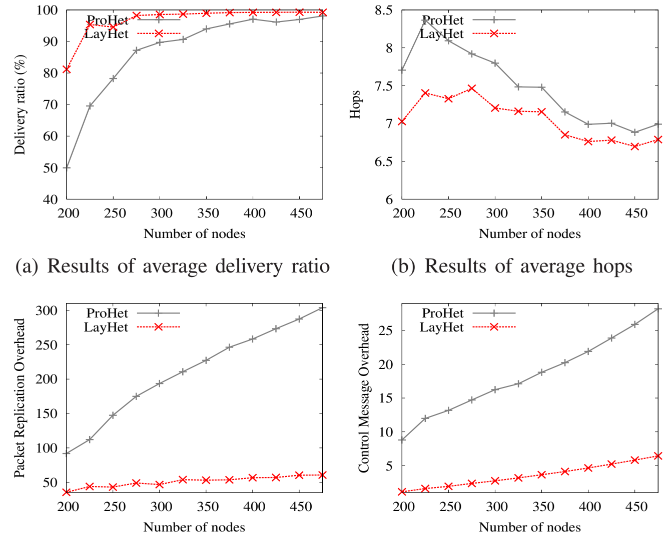
- Average delivery ratio: the average ratio of the number of packets successfully delivered to the total number of packets generated.
- Average hops: the average number of hops to deliver a packet from a source to the sink.
- Average packet replication overhead: the average number of packets replicated to successfully deliver a packet.
- Average control message overhead: the average number of control messages needed to successfully deliver a packet.

In our experiments, nodes are randomly deployed in a $500m \times 500m$ area. To diversify the transmission ranges of nodes, we use the idea in [8] to let a node randomly have one of the three transmission ranges: the *minimum* ($40m$), the *normal* ($50m$), and the *maximum* ($60m$). We randomly set the link packet loss rates between 0% and 20% initially, randomly generate 30 different deployments of heterogeneous sensors, set the assured delivery rate Δ to 99% and vary the number of nodes from 200 to 475 with a step of 25.

The simulation results are shown in Figs. 3(a)(b)(c)(d). In delivery ratio, LayHet is better than ProHet. When the number of nodes increases, the delivery ratio of both algorithms gets closer to 100%. This is because a denser HWSN helps a message to get more chances to be delivered to the sink. The average hops of LayHet are lower than those of ProHet because LayHet embeds the shortest path between a source and the sink in its design. The packet replication overhead of LayHet is only $20\% - 39\%$ of that of ProHet and the control message overhead of LayHet is only $12\% - 23\%$ of that of ProHet. This means unlike ProHet looking for the route in each relay, LayHet can save a lot of effort by identifying node layers at the beginning. However, LayHet has the layer initialization overhead that ProHet does not have. But the layer initialization of all the network nodes only happens once at the beginning. The later node layer adjustment is done locally whenever there is a change.

VII. Conclusion

In this paper, we proposed LayHet, a layer-based routing protocol which takes advantage of the asymmetric links to deliver messages to the sink with an assured delivery rate. Simulation results showed that LayHet outperforms previous routing methods in terms of average delivery ratio, average hops, average packet replication and average control message overhead. Our future work will consider energy consumption and network lifetime using the protocol.



(a) Results of average delivery ratio

(b) Results of average hops

(c) Results of average packet replication overhead

(d) Results of average control message overhead

Fig. 3. Comparison of ProHet and LayHet

References

[1] X. Chen, W. Y. Qu, H. L. Ma, and K. Q. Li, "A geography-based heterogeneous hierarchy routing protocol for wireless sensor networks," *Proc. of the 10th IEEE International Conference on High Performance Computing and Communications(HPCC)*, 2008, pp. 767-774.

[2] X. Du, M. Guizani, X. Yang, H. H. Chen, "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 9, 2007, pp. 3395-3401.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proc. of the 33rd Hawaii International Conference on System Sciences (HICSS)*, Jan. 2008.

[4] Y. F. Hu, W. Z. Li, X. Chen, X. Chen, S. L. Lu, and J. Wu, "A probabilistic routing protocol for heterogeneous sensor networks," *Proc. of IEEE International Conference on Networking, Architecture, and Storage (IEEE NAS 2010)*, July 2010.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proc. of ACM MobiCom*, Aug., 2000, pp. 56-67.

[6] A. Manjeshwar and D. P. Agrawal, "Teen: A routing protocol for enhanced efficiency in wireless sensor networks," *Parallel and Distributed Processing Symposium, International*, Apr. 2001, pp. 2009-2015.

[7] J. Newsome and D. X. Song, "Gem: graph embedding for routing and data-centric storage in sensor networks without geographic information," *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2003, pp. 76-88.

[8] V. Ramasubramanian and D. Mosse, "Bra: a bidirectional routing abstraction for asymmetric mobile ad hoc networks," *IEEE/ACM Transaction on Networking(TON)*, vol. 16, no. 1, 2008, pp. 116-129.

[9] D. Tian and N. D. Georganas, "Energy efficient routing with guaranteed delivery in wireless sensor networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, Mar. 2003, pp. 1923-1929.

[10] M. D. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," *IEEE INFOCOM*, Mar. 2005, pp. 878-890.

[11] F. Ye, H. Y. Luo, J. Cheng, S. W. Lu, and L. X. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," *Proc. of ACM MOBICOM*, Sept. 2002, pp. 148-159.

[12] Q. Zhang and W. G. Chang, "A power efficiency routing protocol for heterogeneous sensor networks," *4th International Conference on Wireless Communications, Networking and Mobile Computing(WiCOM)*, Oct. 2008, pp. 1-4.

[13] W. Y. Zhang, X. J. Du, J. Wu, S. D. Soysa, and Y. Liu, "Near-minimum-energy routing in heterogeneous wireless sensor networks," *IEEE GLOBECOM*, 2010, pp. 1-5.