

EgyHet: An Energy-Saving Routing Protocol for Wireless Heterogeneous Sensor Networks

Xiao Chen, Zanzun Dai, Hongchi Shi

Department of Computer Science, Texas State University, San Marcos, TX, USA

Email: xc10@txstate.edu, zd1020@txstate.edu, hs15@txstate.edu

Abstract—Due to different requirements in application environment, wireless heterogeneous sensor networks (WHSNs) formed by sensors with various capacities are built. Data routing in WHSNs poses special challenges: First, it should be redesigned because the existing ones may not be directly used due to asymmetric links caused by diverse sensor transmission ranges. Second, it should guarantee an assured delivery rate because data is routed through lossy links. Third, it should be energy-efficient due to the limitation of sensor batteries and the difficulty of replacing them after deployment. To address these issues, we propose *EgyHet*: an Energy-saving routing protocol for Heterogeneous sensor networks. *EgyHet* deals with asymmetric links by establishing reverse paths. It saves energy by taking the shortest path, considering the remaining energy in sensors and reducing the number of forwarding nodes while guarantees an assured delivery rate. Simulation results show that *EgyHet* can save more energy yet keep the similar delivery ratio and latency to those of the existing routing protocol for WHSNs.

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

I. INTRODUCTION

Wireless sensor networks (WSNs) are projected to have a significant impact on our daily lives. Most research considers homogeneous sensor networks where sensors have the same capabilities in terms of computation, communication, memory, power supply, reliability, etc. However, when properly deployed, heterogeneous sensors can triple the average delivery rate and provide a five-fold increase in the network lifetime [7]. The wireless networks with heterogeneous sensors are called *Wireless Heterogeneous Sensor Networks* (WHSNs).

Routing in WHSNs poses special challenges: First, because of the asymmetric links caused by different transmission ranges of sensors, e.g. sensor A can reach sensor B, but B may not reach A, the routing protocols developed for regular WSNs may not be directly used in HWSNs or may incur higher overhead [6]. Second, data transmission in WHSNs goes through lossy links, which makes delivery rate important, especially for mission critical systems. Third, due to the limitation of power in sensors, routing algorithms need to be energy-efficient.

Most of the existing routing protocols [2], [4], [8], [9] for WHSNs are coarse-grained in that they roughly divide sensors into two categories: powerful ones and less powerful ones and ignore individual sensor capabilities. Therefore, asymmetric links are not fully utilized. To better address the above challenges, we take sensors' diverse transmission ranges brought about by their heterogeneity into account and put

forward a routing protocol **EgyHet**: an Energy-saving routing protocol for wireless **Heterogeneous** sensor networks. *EgyHet* is a layer-based protocol whose layer information of a sensor embeds the number of hops of the shortest path between the sensor and the sink. The purpose of using layer information is to avoid longer paths to save energy and improve the delivery ratio of packets. The initialization of layer numbers for all nodes in the network only happens once before routing. The later adjustment of layer numbers is done locally. Unlike regular routing protocols, *EgyHet* takes advantage of the asymmetric links in routing by establishing reverse paths. It guarantees an assured delivery rate by letting the source node broadcast H times so that at least one neighbor can receive the message. It saves energy by considering the remaining energy in sensors, reducing the number of forwarding nodes, and letting each receiving node decide the probability Γ to forward the packet to avoid flooding the network with replicated packets. *EgyHet* is an enhancement to our previous protocol *LayHet* in [3]. The major advance is the energy consideration in *EgyHet*. Simulation results show that *EgyHet* can save more energy than *LayHet* yet keep the similar delivery ratio and latency to those of *LayHet*.

The rest of the paper is organized as follows: Section II references the related work. Section III presents the preliminaries. Section IV proposes the *EgyHet* protocol. Section V calculates H and Γ . Section VI shows the simulation results. And conclusion is drawn in Section VII.

II. RELATED WORK

Due to the limitation of power in sensors, many routing protocols have been specially designed for WSNs where energy awareness is an essential design issue [1]. Most of these energy aware routing protocols address sensors with the same sensing and transmission ranges, battery power, etc. Only the methods using the cluster tactic to save energy [2], [4], [8], [9] 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 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. However, the capability of each individual sensor is not distinguished and the asymmetric links are not utilized. In [3],

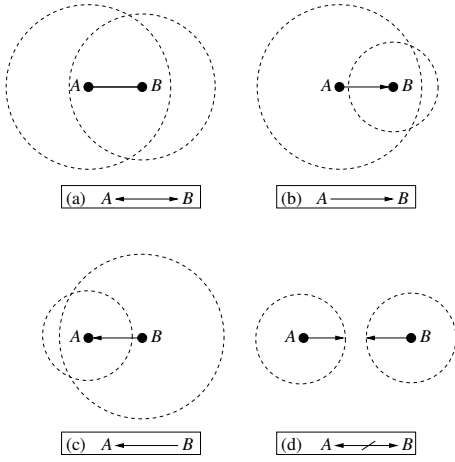


Fig. 1. The neighbor relationships between two nodes A and B . (a) A and B are each other's In-out-neighbor; (b) A is the In-neighbor of B and B is the Out-neighbor of A ; (c) B is the In-neighbor of A and A is the Out-neighbor of B ; (d) A and B are Non-neighbors

we put forward a protocol LayHet that takes sensors' diverse transmission ranges into account and utilizes asymmetric links in routing while achieves an assured delivery rate. In this paper, we enhance LayHet to make it more energy-efficient.

III. PRELIMINARIES

A. Definition of nodes' neighbor relationships

A WHSN 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 sensor network is a strongly-connected directed graph. Therefore, the sensor network is also strongly-connected.

We divide 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 $A \rightarrow B$ and $B \rightarrow A$, then A and B are In-out-neighbors of each other. If only $A \rightarrow B$ (or $B \rightarrow A$), 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 $A \rightarrow B$ nor $B \rightarrow A$, they are Non-neighbors of each other.

B. Definition of packet loss rate of a link

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 and the total number of packets N_s sent by u . It can be expressed as:

$$P_{lossrate} = 1 - N_d/N_s \quad (1)$$

IV. ENERGY-SAVING ROUTING PROTOCOL

In this section, we present EgyHet which has two parts: The preparation part which includes finding a reverse paths for each

asymmetric link, assigning layer numbers to the nodes and adjusting the layer numbers periodically; And the routing part which includes the sender broadcasting H times based on the remaining energy of sensors 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 [5], which is omitted here because of space limitation, to establish a reverse path between a node and each of its In-neighbors. The idea is as follows: If a node u receives another node v 's "Hello" message but not the "Ack" of its own "Hello" message, then u knows that v is its In-neighbor. Then u will broadcast a "Find" message containing the source ID (" u "), the destination ID (the ID of the In-neighbor to which it wants to find the reverse path (e.g. " v ")), and an expiration length of 3 hops. The expiration length is set to 3 because reverse paths can be successfully found for 97% of the asymmetric links [6] within 3 hops. If the receiver of the "Find" message is the destination, it will send back a "Path" message containing the reverse route to the source u . If the receiver is not the destination but is still within 3 hops, it will rebroadcast the message. In all other cases, the message will be dropped.

2) *Deciding initial layer numbers*: Each node needs to find out its layer number which represents its shortest hop count to the sink. The process is as follows: a node u broadcasts an exploration packet EP containing a hop-count 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 EP s 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 hop-count c from u to the sink. Then the sink sends back an Ack of EP containing c to u via all of the forwarding nodes on the path. Because of the asymmetric links, the reverse paths may be used when the sink sends back an Ack. Fig. 2 shows that all of the sensors are virtually put into different layers relative to the sink after this process.

3) *Adjusting layer numbers*: In deciding nodes' initial layer numbers, because of lossy links, some nodes may not get the right layer numbers. So their layer numbers need to be adjusted. To reduce the overhead, the adjustment of layer numbers is done locally and can be embedded in Algorithm UPR-P in subsection IV-B.

B. Routing

After the preparation, the routing part of EgyHet that is directed by the remaining energy of sensors to achieve an assured delivery rate using local information can be carried

out. The assured delivery rate is preset to Δ . The routing part contains three phases: Broadcasting H times, Forwarding messages, and Updating packet loss rate periodically which are presented in Algorithms BRD-H, FWD-M and UPR-P, respectively. The H value in Algorithm BRD-H is calculated by Algorithm CAL-H based on the remaining energy of sensors.

In Algorithm BRD-H, energy-efficiency is reflected in two ways: use nodes with high remaining energy and reduce the number of forwarding nodes. 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.

Algorithm BRD-H calls algorithm CAL-H to calculate the H value based on the remaining energy of sensors. In algorithm CAL-H, S represents the selected forwarding nodes with the highest remaining energy. First, the K lower layer In-out-neighbors or Out-neighbors of node u are ordered in non-increasing order based on their remaining energy levels. Then starting from the node with the highest remaining energy, we add node one by one to S . After adding a new node, we use formula (2) in Section V to calculate H . The H value may be reduced with the increase of the node number in S . The algorithm stops if the new added node does not reduce H any more or all of the K nodes are added. After the H value is known, u broadcasts the message containing the packet it wants to send to the sink, the selected forwarding nodes in S and its link loss rates to the forwarding nodes H times so that at least one of them can receive the message with a high probability in order to achieve the assured delivery rate Δ .

Next in the FWD-M algorithm, a receiving node v , if it is selected as a forwarding node, will forward the message at a probability of Γ ($0 \leq \Gamma \leq 1$). The purpose of the forwarding probability is to avoid flooding the network with unnecessary messages to save energy but at the same time guarantees that at least one of the forwarding nodes forwards the message with a high probability so as to reach the assured delivery rate. The explanation of the formulas to calculate H and Γ is presented in Section V.

The UPR-P algorithm is used to update the packet loss rate of each link periodically so that the next routing can be guided by more accurate information in the network. The layer numbers of nodes are also updated in UPR-P by piggybacking.

Now we use an example in Fig. 2 to explain the EgyHet protocol. In the figure, each black dot represents a sensor and the right most node is the sink. All sensors will send collected data to the sink.

In the preparation part, a reverse path is set up between a node and each of its In-neighbors. Each node is virtually put into a different layer relative to the sink after knowing its layer number. Even if it is not in the right layer because of the lossy links, it can still update its layer number later by algorithm UPR-P in the routing process.

In the routing part, suppose a source node u in Layer L_5

Algorithm BRD-H: Broadcasting H times

- 1: Source node u finds out the packet loss rates p_1, p_2, \dots, p_K of links 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.)
 - 2: Node u calculates the value of H and selects the forwarding set S out of the K In-out-neighbors or Out-neighbors using Algorithm CAL-H.
 - 3: Node u broadcasts the message containing the packet it wants to send to the sink, the selected forwarding nodes in S and its link loss rates to them H times to its In-out-neighbors or Out-neighbors.
-

Algorithm CAL-H: Calculating H value

- 1: Order node u 's K lower layer In-out-neighbors or Out-neighbors in non-increasing order according to their remaining energy levels. Here we use a node's remaining energy level to represent the node. Suppose the sequence $E = \{E_1, E_2, \dots, E_K\}$.
 - 2: $S = \{E_1\}, i = 0$.
 - 3: Calculate H based on sequence S using formula (2) in Section V.
 - 4: **repeat**
 - 5: $H_{pre} = H, i = i + 1$.
 - 6: **if** $i = K + 1$ **then**
 - 7: return H and S
 - 8: **end if**
 - 9: $S = S \cup \{E_i\}$.
 - 10: Calculate H based on sequence S using formula (2) in Section V.
 - 11: **until** $H == H_{pre}$
 - 12: return H and $S = S - \{E_i\}$.
-

Algorithm FWD-M: Forwarding Messages

- 1: **if** a node v receives a message from a higher layer neighbor u **then**
 - 2: **if** it is selected as a forwarding node **then**
 - 3: it uses formula (3) in Section V to decide its probability Γ to forward the message.
 - 4: **if** it forwards **then**
 - 5: it becomes the new source and applies the BRD-H algorithm.
 - 6: **end if**
 - 7: **end if**
 - 8: It simply drops the message.
 - 9: **end if**
-

wants to send a packet to the sink. Suppose 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

Algorithm UPR-P: Updating Packet loss Rate Periodically

- 1: Each node u updates the packet loss rate of each of its links with its In-out-neighbors or Out-neighbors every T time period.
 - 2: 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?”.
 - 3: 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.
 - 4: After u receives the answer, it updates the packet loss rate of link uv to $1 - \frac{N_d}{N_s}$. Also if u 's layer number is at least 2 more than v 's layer number, u adjusts its layer number to v 's layer number +1.
-

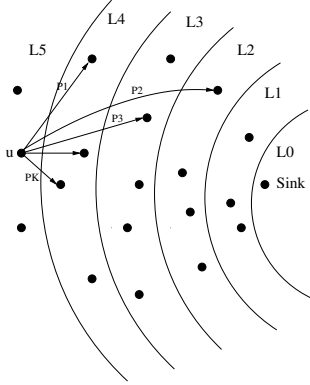


Fig. 2. A data forwarding scenario

the nature of broadcast. Based on the remaining energy and the link packet loss rates of its In-out-neighbors or Out-neighbors, H is calculated and a forwarding set is chosen. Then u broadcasts the message H times, which guarantees that at least one of the forwarding nodes will receive the message with a high probability. Then each of these forwarding nodes will decide the probability Γ to forward the message if it receives the message. If a node chooses to forward, it becomes the new source and the EgyHet routing protocol will be reapplied. Then hop by hop, the message will be relayed to the sink. Every T period of time, a sender will update the packet loss rates of its links to its In-out-neighbors or Out-neighbors so that the calculations of H and Γ can be more accurate next time.

V. CALCULATION OF H AND Γ

In this section, we show how to calculate H and Γ given assured delivery rate Δ and the packet loss rates of links between a sending node and its selected forwarding nodes.

Assume node u at layer i (see Fig. 2) has a total of K lower layer In-out-neighbors or Out-neighbors. It wants to broadcast messages to k ($k \leq K$) selected lower layer forwarding nodes with whom the packet loss rates p_1, p_2, \dots, p_k are known

through random setting initially or periodic message exchange in Algorithm UPR-P later. Because of lossy links, it may need to broadcast multiple times to make sure that at least one node in the forwarding set can receive the message. Normally the message needs to travel i hops to reach the sink. Assume the transmission in each layer is the same. To guarantee the overall delivery rate Δ , in each layer, we should guarantee the success rate of transmission to be at least $\Delta^{\frac{1}{i}}$. We assume the packet loss rate of the link from u to its j -th forwarding node is p_j . A transmission is successful if at least one of the forwarding nodes receives the message. That is, the probability

$$\begin{aligned} & Pr\{\text{at least one forwarding node receives the} \\ & \text{message after } H \text{ transmissions}\} \\ &= 1 - \left(\prod_{j=1}^k p_j\right)^H \geq \Delta^{\frac{1}{i}} \end{aligned}$$

We have

$$H \geq \frac{\ln(1 - \Delta^{\frac{1}{i}})}{\sum_{j=1}^k \ln(p_j)} \quad (2)$$

We set H to the minimum integer that can satisfy formula (2). After node u broadcasts the message H times, the message is transmitted to one or more forwarding nodes with high probability. To reduce redundancy and save energy, not all of the nodes receiving the message will forward the message. The receivers only forward the message with probability Γ and drop the message with probability $1 - \Gamma$. Given that a message has been received by a few forwarding nodes, we should make sure at least one node will forward the message. That probability

$$\begin{aligned} & Pr\{\text{at least one forwarding node will forward} \\ & \text{the message}\} \\ &= 1 - Pr\{\text{no one will forward the message}\} \\ &= 1 - \prod_{j=1}^k (Pr\{\text{the } j\text{th node does not receive the} \\ & \text{message}\} + Pr\{\text{the } j\text{th node receives the} \\ & \text{message}\} \cdot Pr\{\text{the } j\text{th node doesn't forward} \\ & \text{the message}\}) \\ &= 1 - \prod_{j=1}^k (p_j^H + (1 - p_j^H)(1 - \Gamma)) \\ &= 1 - \prod_{j=1}^k (1 - \Gamma + p_j^H \Gamma) \geq \Delta^{\frac{1}{i}} \end{aligned}$$

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

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

Now Γ can be set to the minimum value that satisfies formula (3).

VI. SIMULATION

The most relevant routing protocol that we can compare EGYHet with is the LayHet protocol in [3] since others do not guarantee an assured delivery rate, utilize asymmetric links and differentiate diverse sensor transmission ranges. We conduct our experiments using a self-written simulator in Java language and the performance of the two protocols is evaluated by the following metrics:

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

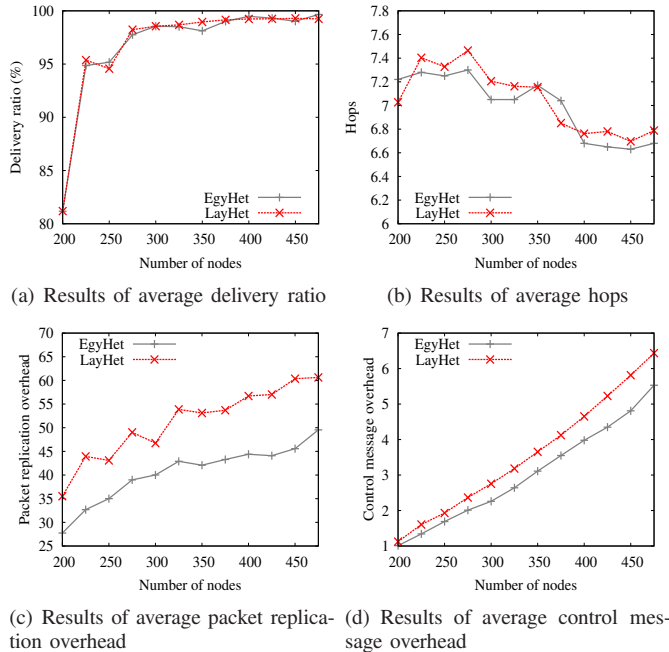


Fig. 3. Comparison of EGYHet and LayHet with assured delivery rate 99%

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 [6] to let a node have one of the three transmission ranges: the *minimum* (40m), the *normal* (50m), and the *maximum* (60m). A transmission range is

selected randomly out of the three for a node in simulation. We randomly set link loss rates to be between 0% and 20%. To implement message sending and receiving, a virtual concept of *time slots* is used. In each time slot, we randomly choose a sensor to generate a new message and let it send the message to the sink. Each node uses a buffer to cache packets from other nodes. Assume all packets in the buffer can be transmitted to the next hop node within one time slot. The simulation time is set to 1000 time slots. During the experiments, we randomly generate 30 different deployments of heterogeneous sensor nodes, set the assured delivery rates to 99%, and vary the number of nodes from 200 to 475 with a step of 25 and calculate the average performance in the simulation results.

The simulation results are shown in Fig. 3. EGYHet has lower average packet replication overhead and control overhead than those of LayHet. This is because EGYHet just selected a subset of K neighbors to route packets. Yet EGYHet achieves similar delivery ratio and average hops.

VII. CONCLUSION

In this paper, we proposed *EGYHet*: an energy-saving routing protocol for heterogeneous sensor networks, which saves energy by taking the shortest paths, considering the remaining energy of sensors and reducing the number of forwarding nodes while takes advantages of asymmetric links and guarantees an assured delivery rate. The comparison of EGYHet with our previous protocol LayHet shows that EGYHet can save more energy yet keep similar delivery ratio and latency. In the future, we will work on more energy-efficient routing protocols for WHSNs.

REFERENCES

- [1] J. N. Al-Karaki, A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Transactions on Wireless Communications*, Vol. 11, Issue 6, Dec. 2004, pp. 6-28.
- [2] 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.
- [3] X. Chen, Z. X. Dai, W. Z. Li, and H. C. Shi, "A layer-based routing protocol for heterogeneous wireless sensor networks," *Proc. of IEEE ICC*, June 2012, pp. 233-237.
- [4] 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.
- [5] 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.
- [6] V. Ramasubramanian and D. Mosse, "Bra: a bidirectional routing abstraction for asymmetric mobile ad hoc networks," *IEEE/ACM Transactions on Networking (TON)*, Vol. 16, No. 1, 2008, pp. 116-129.
- [7] M. D. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," *Proc. of IEEE INFOCOM*, Miami, Florida, USA, Mar. 2005, pp. 878-890.
- [8] Q. Zhang and W. G. Chang, "A power efficiency routing protocol for heterogeneous sensor networks," *Proc. of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Dalian, China, Oct. 2008, pp. 1-4.
- [9] W. Y. Zhang, X. J. Du, J. Wu, S. D. Soysa, and Y. Liu, "Near-minimum-energy routing in heterogeneous wireless sensor networks," *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, 2010, pp. 1-5.