A Novel Clustering Topology Control for Reliable Multi-hop Routing in Wireless Sensor Networks

Ruiying Du[1,2], Chunyu Ai[3], Longjiang Guo[4,5], Jing Chen[1,2], Jianwei Liu[6], Jing He[5], and Yingshu Li[5]
[1]School of Computer science, Wuhan University, Wuhan,China
[2]State Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan, China
[3]Department of Mathematics, Physics, Computer Science, and Geomatics, Troy University, Troy, AL 36082, USA
[4]Department of Computer Science, Heilongjiang University, Harbin, China
[5]Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
[6]School of Electronics Information Engineering, Beihang University,Beijing,China
Email: duraying@gmail.com

***Abstract*— The reliability of wireless sensor networks is significant in certain applications, especially the reliable routing. Most existing routing protocols use multi-paths to improve routing reliability. However, multi-paths waste a large amount of energy to obtain redundancy. This is not an optimal option for sensor nodes with limited energy. In this paper, a novel clustering-based reliable multi-hop routing algorithm ($CRMR$) is proposed. The algorithm adopts a mechanism of multiple backup cluster heads efficiently to extend time of stable period of clusters and to decrease energy consumption for reconstructing clusters. The local reconstruction of clusters is addressed for improving coverage, maintaining connectivity, and extending the network lifetime. While the algorithm overcomes the randomicity of selecting cluster heads and ensures well proportioned clusters. Employing backup cluster heads and gateways can ensure reliability of routing and overcome disadvantages of most existing reliable routing protocols, which is to preserve multiple backup paths. The algorithm adopts query driving data transmission mode for finding routes and bypassing unavailable routing nodes for backtracking to ensure the speediness of data transmissions and the reliability. The simulation results show that the algorithm can achieve good performance on both routing reliability and energy consumption.**

***Index Terms*— wireless sensor networks, reliable routing, energy efficiency**

I. INTRODUCTION

It is well known that the reliability of a Wireless Sensor Network (WSN) is restricted by the limited energy, memory space, and computation ability of sensor nodes. The affection of surrounding environment changes or low residual energy could result in sensor nodes not working properly, even invalidation. Also, it is not easy to maintain and replace sensor nodes after deployment [1]. Most applications require WSNs to have a high level of fault-tolerance, on both hardware and software [2]. Consequently, the system should automatically adapt or correct faults so as to keep the entire network working continuously and properly. For a certain WSN which has an extra requirement of the fault-tolerance, the routing protocol design must consider the adoption of certain mechanisms to ensure the stability of routing [3].

Most current existing reliable routing protocols are based on the planar routing mechanism [4], [5], [6], [7], [8], and [21]. For ensuring reliability they use multiple paths to transmit data simultaneously. However, these algorithms consume more energy and/or require storage of multiple paths in sensor nodes.

In the hierarchical model of clustering, the reliability of clusters mainly includes two aspects, the cluster-heads' reliability and gateways' reliability. In the civil environment, reliable cluster-heads and gateways lie in whether their energy is sufficient. Since cluster-heads and gateways consume more energy than other sensor nodes, we periodically cluster to replace these nodes so as to balance energy consumption among nodes, e.g., algorithm LEACH [9]. However, LEACH does not consider the stability of clusters. Also, it does not care a node's residual energy when it is chosen to be the cluster-head. Moreover, consecutive clustering must consume a large quantity of energy, so the network lifetime is shortened rapidly.

In this paper, we propose a valid solution to address this issue. For prolonging the network lifetime, sensor nodes are divided into two groups, working nodes and backup nodes. We let backup nodes go to sleep to conserve energy. A backup node is activated to replace the working node which cannot work properly caused by using up energy or being damaged. The entire network is divided into clusters by using the clustering algorithm. During the clustering phase, one cluster head and several backup cluster heads are generated in each cluster. Gateway nodes and backup gateway nodes which can connect adjacent cluster heads are generated too. When a small area of the network cannot work properly, we use local cluster reconstruction algorithm to regenerate clusters for

Manuscript received October 29, 2009; revised February 13, 2010; accepted May 31, 2010.

doi:10.4304/jcm.5.9.654-664

that area. If most of cluster heads, gateways, and their backups have low energy, the sink starts a global cluster reconstruction. When the sink sends a data request, the routing algorithm generates a shortest path from the sink to an target area using query-driven data transmission mode based on node locations. This route only consists of the cluster-heads and gateways. When the acquired data is sent back to the sink, it is transmitted along the above path. In order to ensure the data transmission reliability and efficiency, the shortest path bypassing algorithm is utilized when a node of the path is failed.

Our novel scheme has the following contributions and characteristics:

1) Multiple backup cluster-heads and gateways can improve the stability of clusters and the reliability of routes and decrease energy consumption for clustering.
2) Sensor nodes are divided into working nodes and backup nodes to prolong the network lifetime by letting backup nodes go to sleep.
3) The network stability can be maintained by the local cluster reconstruction instead of global cluster reconstruction for most of situations, thus conserving energy for clustering.
4) The mechanism, multiple backup cluster-heads and gateways can enhance the routing fault-tolerance.
5) The algorithm adopts the query-driven data transmission mode and bypasses unavailable sensor nodes to ensure the reliability and speediness of data transmissions.

The rest of this paper is organized as follows. Section II presents related works. The related theory is introduced in Section III. The detailed working scenario of the proposed framework is illustrated in Section IV. The simulation results are shown in Section V. Section VI concludes this paper.

II. Related work

The Flooding routing algorithm is a traditional method, which has an extremely high reliability. Normally, it is employed in the military activities. The Flooding routing algorithm guarantees the fault-tolerant ability and reliability by using the redundancy mechanism. As we know, communication consumes most energy consumption of WSNs. Since the Flooding routing causes a large amount of messages, it is not suitable for WSNs which have limited energy.

A lot of applications do not require high reliability, so some works incline to improve the Flooding routing algorithm by reducing the energy consumption. Directed Flooding [6] is a descendant of the Flooding routing protocol. It is a fault-tolerant and energy efficient routing protocol compared to the Flooding algorithm. In stead of broadcasting used by the traditional Flood protocol, messages are sent or forwarded in a specific directional virtual aperture. This routing mechanism can reduce the energy consumption efficiently. As a consequence, it can prolong the network lifetime. However, the energy consumption performance of Directed Flooding is still not good enough for WSNs with limited energy.

The work in [4]proposed a multiple path routing mechanism. It establishes and maintains a group of routing paths in advance. This brings an advantage that the network routing paths could be refreshed without periodically Flooding. Firstly, it creates an optimal primary path and multiple backup pathes from the source to destination. The message is delivered through the primary path, and backup paths are used when the primary path fails. To maintain the backup paths, a low-rate data is delivered through backup paths. When the primary path fails, the algorithm will choose the secondary optimal path as the primary path as soon as possible. For establishing multiple paths, two different methods are addressed, the disjoint multiple path routing and the braided multiple path routing. In the disjoint multiple path method, the backup paths might be longer than the primary path. The braided multiple path routing can conquer the single fault problem. The ideal case is that the backup paths are node disjoint with the primary path. However, this needs to consume sensors' resource to compute and maintain multiple backup paths. It implies the network lifetime is shortened due to extra energy consumption for maintaining multiple paths.

$ReInForM$ [5] proposed a reliable routing protocol that data can be delivered at desired levels of reliability at proportional cost in spite of significant channel errors. It uses the concept of dynamic packet state to control the number of paths required by the desired reliability. It only uses local knowledge of channel error rates and does not require any prior computation or maintenance of these multiple paths. The basic process of $ReInForM$ has two steps. Firstly, the source node computes the number of paths required by the transmission reliability. Then, it chooses the next hop nodes among its neighbors and distributes the number of paths to them. When a node receives the particular data package from a source node, it will consider itself as the source node and repeats the above process to forward the data packet. $ReInForM$ considers both the system demand and he signal path quality, and it chooses the number of paths dynamically. The number of paths is adjusted according to the signal quality of paths. As a result, it still can achieve high successful delivery rate when the signal path quality is low and conserve energy when the quality is high by reducing the number of paths. The simulation results show that $ReInForM$ algorithm provides a tradeoff between reliability and energy consumption, and the overall energy consumption is much lower than the Flooding. However, $ReInForM$ demands complicated computation at every node, thus requiring a high computing ability for a node. Moreover, the $ReInForM$ method cannot adjust routing in real-time manner according to the changes of sinal quality. Although, the $ReInForM$ routing can satisfy the system demand, it is a passive tolerant method. Also, the $ReInForM$ cannot detect failed nodes.

For improving the $ReInForM$, the work in [**?**]

proposes a multi path tolerance mechanism based on none-intersect multi path routing which aims to enhance the system reliability. This model combines the positive tolerance and the passive tolerance control technology to guarantee the reliability of routing. It assigns a reliable level to each path. The mechanism adapts the multi path transmission according to the system demand and changes the path reliable level dynamically according to the success rate of data transmission. Consequently, this method can maximize the transmission ability and guarantee the reliable level. These efforts can reduce the computation and transmission requirement to nodes compared to $ReInForM$. Also, it can save a considerable amount of energy and eventually prolong the network lifetime. This method also needs to consume sensors' resource to compute and maintain multiple paths, and utilizes several paths to transmit data simultaneously. The network lifetime is shortened due to the plentiful energy consumption too.

Although a lot of routing protocols do not focus on routing reliability issue, they use the redundant mechanism in data transmission which can obtain some routing tolerant ability, e.g., LEACH. Because the process of cluster head choosing and establishing clusters is dynamic, a failed node does not participate in the cluster establishing process. Obviously, if the cluster head works well, the data forwarding process can go on smoothly. Even the cluster head loses its ability in a specific round, it only affects the data transmission of that round. The tolerance ability of the LEACH algorithm can be enhanced by reducing the stable period. However, the short stable period increases the cost of clustering. It is a challenge to reach the tradeoff between the network overhead and the tolerance ability. In addition, the algorithm proposed by [10], [11], [12],and [19] are also based on a hierarchical network.

In summary, for cluster structure, because the network reliability depends on the validity of cluster-heads and gateways, once these nodes are damaged or deplete their energy, the network performance will be influenced evidently. Moreover, for prolonging the network lifetime and efficiency, we should make the stable period of clusters much longer than the clustering phase, thus reducing the number of rounds of clustering.

III. PRELIMINARIES

In this section, we first introduce some concepts and definitions.

A. The coverage and connectivity of clusters

Lemma 1: There are three circles $D1$, $D2$, and $D3$ with the radius r, and the centers of these circles are $C1$, $C2$, and $C3$ respectively (as shown in Figure 1). When the circle D1, D2, and D3 intersect at the same point and $\triangle\, C1C2C3$ is an equilateral triangle, the maximum seamless area covered by $D1$, $D2$, and $D3$ is obtained, and the area is $\frac{4\pi+3\sqrt{3}}{2}\gamma^2$.

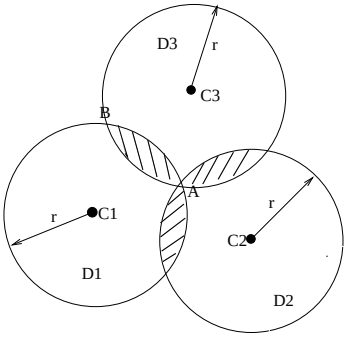


Figure 1. Three seamless topological circle field.

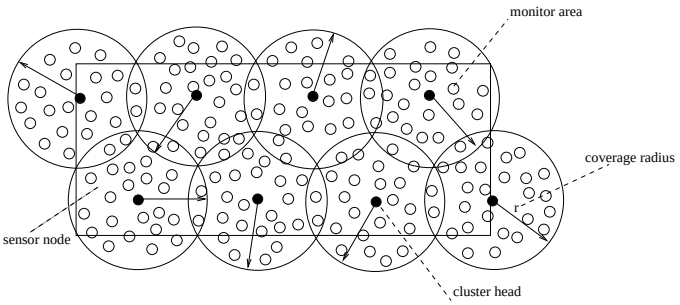


Figure 2. A cluster-based optimal WSN

The seamless connectivity ([14], [15], [16]) and [17]) refers to using the least number of circles to cover a rectangle area in which each point is covered by at least one circle. For wireless sensor networks, we consider a sensor's transmission coverage as a circle which is centered at the location of the sensor with the radius r. Here, r is a sensor's transmission range, and we assume all sensor nodes have the same transmission range r. If a sensor node is chosen as a **cluster head**, we define the coverage circle of the cluster head's transmission range as a **cluster**. All sensor nodes within cluster is the members of this cluster. It is possible a sensor node is a member of several clusters. Usually, we choose this kind of sensor nodes as gateway nodes to connect clusters. As shown in Figure 2, black sensor nodes are chosen as cluster heads, so a rectangle area is fully covered by these eight clusters. Minimizing the number of circles (clusters) is actually the issue of maximizing the coverage area of each circle. In other words, based on the condition of seamless connectivity, we maximize the valid coverage area of each circle and sufficiently utilize every circle area. This problem can be normalized as follows. We use F to indicate the monitored area, and there are N circles ($D = \{D_1, D_2, \cdots, D_N\}$) which can seamlessly cover F. That is, $F \subseteq \bigcup_{D_i \in D} D_i$, where $\cup$ is the coverage union operation of circles. Our purpose is to find a minimum subset M of D which can satisfy $F \subseteq \bigcup_{D_i \in M} D_i$.

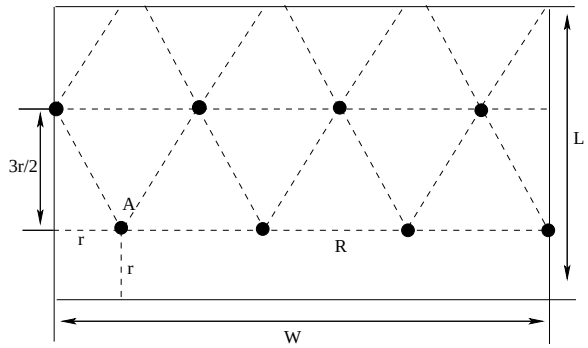


Figure 3. Grid division of WSN

B. The grid design on connectivity

Given a monitored field F with the length L and width W, the ideal cluster head deployment which covers the entire F is shown in Figure 3. We make every cluster head and two of its neighbor nodes to generate an equilateral triangle, thus achieving maximum coverage. Then, the sensor network achieves the seamless coverage. In Figure 3, R express the edge length of every equilateral triangle, it's valule is $\sqrt{3}r$, where r is the radius of a cluster . According to the above deployment, the minimum number of clusters to cover F is $M = [\frac{L-r}{3r/2}] \times [\frac{W-r}{\sqrt{3}r}]$.

C. Analysis

According to the above theory and model, we can make optimal clustering division for a particular network so as to minimize the number of clusters by using seamless topology. However, for most sensor networks it is not practical to construct clusters according to the above model. First of all, for most sensor networks, a sensor node does not know its geographical location. Even though the location information is known, sensor nodes cannot be deployed so perfectly to satisfy the equilateral triangle requirement.

In this paper, we propose an self-adapting cluster head selection method. In order to minimize the number of clusters, we limit the size of intersection area between any two adjacent clusters instead of trying to form an equilateral triangle. Moreover, for keeping the communication connectivity of networks, we choose a sensor node in the intersection of two clusters as a **gateway** to connect two cluster heads. To improve the reliability of routing, one or more backup cluster heads and gateways are chosen to replace the primary one when it fails. The facts affecting the number of backup cluster heads include: (1) The density of sensor node deployment; (2) The sensor node energy consumption of transmission; (3) The requirements of applications. We also can refer to the number of gateways. Since the cluster heads execute more tasks than gateways, the number of cluster heads should be greater than the number of gateways. To keep the connectivity among clusters, a backup cluster head should be as close as possible to its cluster head. If the density of sensor nodes in a network is d, according to Lemma 1, the intersection area of two adjacent clusters Sr is: $\frac{2\pi-3\sqrt{3}}{6}\gamma^2$. Then, the distance between a cluster head and its backups, Rr, is $\sqrt{\frac{2\pi-3\sqrt{3}}{6\pi}}\gamma \approx 0.24\gamma$.

IV. Redundancy-based Reliable Directional Clustering Multi-hop Routing Algorithm

In this section we describe the design of our Redundancy-based Reliable Directional Clustering Multi-hop Routing Algorithm ($RDMR$). We introduce the working scenario of our algorithm in Section IV-A. In Section IV-B how to build initial clusters is represented. Maintaining clusters is discussed in Section IV-C. Local and global cluster reconstructions are introduced in Section IV-D and IV-E respectively. How to build routing paths is described in Section IV-F, and Section IV-H introduces data transmission.

A. Working Scenario

After deploying sensor nodes, the initial cluster construction algorithm is invoked to build clusters for the entire network. The sink chooses a sensor node to start this process. Clusters are established in an in-network manner. Since a cluster is determined by its cluster head, building clusters is actually how to choose cluster heads. A sensor node with more residual energy and achieving more coverage has priority to be chosen. A new generated cluster head must connect with existing cluster through gateways, so we also generate gateways when building clusters. To guarantee the routing reliability, we also generate backups for cluster heads and gateways. When primary cluster heads or gateways fail or deplete energy, these backups can replace them. Other sensor nodes join in a closet cluster. If a sensor node is not a primary or backup cluster head or gateway and no specific task is assigned by users, it can go to sleep to conserve energy. After clusters are generated, the entire network can start working. We also introduce how to identify routes and transmit data based on our cluster topology.

Since cluster heads and gateways are in charge of collecting and forwarding data, they deplete energy quickly than other sensor nodes. If a cluster fails, it will affect the communication around that area. For this situation, an existing algorithm such as LEACH reconstructs clusters for the entire network. Frequent global cluster reconstructions cause too much energy consumption. We locally reconstruct clusters for the failed area. When most of cluster heads and gateways have low energy level, the sink starts a global cluster reconstruction.

B. Initial Cluster Construction

As we mentioned before, a cluster is determined by the location of its cluster. In this paper, a cluster is defined as the covered transmission range area of its cluster head. If the transmission range of a sensor node is r, the cluster is a circle centered at the cluster head with radius r. A cluster head is responsible for node management in its cluster, data fusion, and communication among clusters. According to optimal deployment mentioned in Section

III-B, the ideal distance between two adjacent clusters are $\sqrt{3}r$, so two adjacent cluster heads might not be in each other's transmission range. We choose a sensor node which is in the intersection area of two adjacent clusters as the gateway node to connect clusters. The gateway node is the communication bridge between two adjacent clusters. To improve the routing reliability, every cluster-head can choose multiple sensors (at least one) close to itself as backup cluster-heads. The sensor nodes except the gateway node in the intersection area of two clusters can be chosen as a backup gateway. These backups of cluster heads and gateways usually stay in a sleeping state, when cluster-heads or gateways cannot continue their normal work, the backups are activated to replace them.

In the process of cluster construction, we use the following states of sensor nodes: $idle$ means the node is in a initial state; ch means the node has been declared as a cluster-head; ch-r means the node is a backup cluster head; gw means this is a gateway; gw-r means the node is a backup gateway; $member$ means the node is a normal member node of a cluster.

The process of cluster construction is to choose cluster heads. If a sensor node is chosen as a cluster head, it has to satisfy the following conditions:

1) A cluster head cannot be a member of other clusters. In other words, if a sensor already joined a cluster, it cannot be chosen as a cluster head. Consequently, a candidate cluster head must be in $idle$ state. The requirement can make sure the distance between any two cluster heads is greater than r (transmission range of a sensor node).
2) If a sensor node is chosen as a cluster head, its cluster must have intersection with at least one existing cluster. Also, the number of sensor nodes in the intersection is another concern since gateways are generated among these sensor nodes. This restrict can make sure the new generated cluster is not isolated from others.
3) Sensor nodes which satisfy the above two conditions are candidate cluster heads. Define a function $f = N \times energy$, where N is the number of idle neighbor nodes and $energy$ is the residual energy of a candidate cluster head. Then, the candidate with the maximum f is chosen as the cluster head. As a result, a sensor node with more residual energy and

After deployment of sensor nodes, every sensor node broadcasts a HELLO message. A sensor node N_i inserts the sensor node ID of received HELLO message into its neighbor list. The initial cluster construction algorithm as shown in Algorithm IV-B is invoked on every sensor node. Before cluster construction, all sensor nodes' state is idle. First of all, the sink chooses a sensor node as the first cluster head. Then, this sensor node broadcasts a HEAD message to declare itself as a cluster head. If an idle sensor node hears the HEAD message, it sends a JOIN message to join this cluster as a member. If an idle sensor node already receives ng JOIN messages from the same cluster members, it can be a candidate cluster-head. Here, ng is the required number of gateways including the primary one and backups. This requirement can guarantee that a new generated cluster connects to the existing one and there are enough sensor nodes which can be gateways between them. Among these candidate cluster-heads, the one with the maximum f is chosen as the cluster-head. Since every candidate cluster head broadcasts a CANDIDATE_HEAD message with its f value, a candidate can compare its f with others it received. If it has the maximum f, it declares itself as a cluster-head and broadcasts a HEAD message. Also, it will pick the closest member as the backup cluster-head according to the signal strength of received JOIN messages. Sensor nodes which sent JOIN messages to up two the current cluster-heads will be chosen as a gateway node or backup gateway.

Algorithm 1 Initial Cluster Construction

1: $N_i.idleneighbor$= the number of N_i's neighbors.
2: $N_i.role = idle$. /*The current node is N_i.*/
3: **while** Waiting for receiving a message **do**
4: **if** The received message is HEAD AND $N_i.role\ is\ idleormember$ **then**
5: $N_i.role = member$.
6: Broadcast a JOIN N_i message.
7: **end if**
8: **if** The received message is JOIN N_k AND $N_i.role\ is\ idle$ **then**
9: $N_i.idleneighbor - -$.
10: $N_i.N_k.JOIN++$. /*Initial value is 0.*/
11: **if** $N_i.N_k.JOIN >= ng$ **then**
12: Broadcast a CANDIDATE_HEAD message with $N_i.f = N_i.idleneighbor * energy$.
13: Wait for t time. /*t is a predefined waiting delay.*/
14: **if** $N_i.f > any\ N_j.f$ of received CANDIDATE_HEAD message **then**
15: Broadcast a HEAD message to declare N_i as a cluster head.
16: Wait for JOIN messages from its non-member neighbors, pick a sensor node with the strongest signal intensity as the backup cluster head, and inform that sensor node.
17: Pick a primary GATEWAY and $ng - 1$ backup GATEWAYs from these sensor nodes which sent JOIN N_k messages and inform them.
18: **end if**
19: **end if**
20: **end if**
21: **if** The received message is JOIN N_k AND $N_i.role\ is\ ch$ **then**
22: $N_i.N_k.JOIN++$.
23: **if** $N_i.N_k.JOIN >= ng$ **then**
24: Pick a primary GATEWAY and $ng - 1$ backup GATEWAYs from these sensor nodes which sent JOIN N_k messages and inform them.
25: **end if**
26: **end if**
27: **end while**

An example of cluster construction is shown in Figure 4. Suppose only one backup cluster head and one backup gateway are required. The sink chooses node A to start, so A broadcasts a HEAD message. Node B, C, D, K, J, and X receive the HEAD message and send JOIN messages to join as members. When A receives JOIN messages,

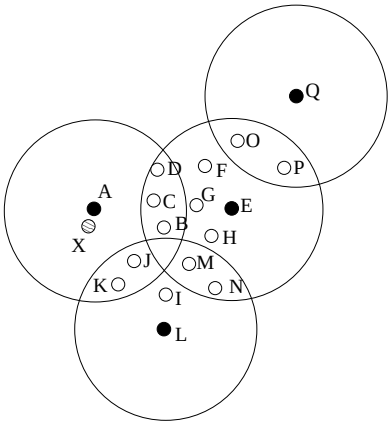


Figure 4. Example of identifying cluster heads

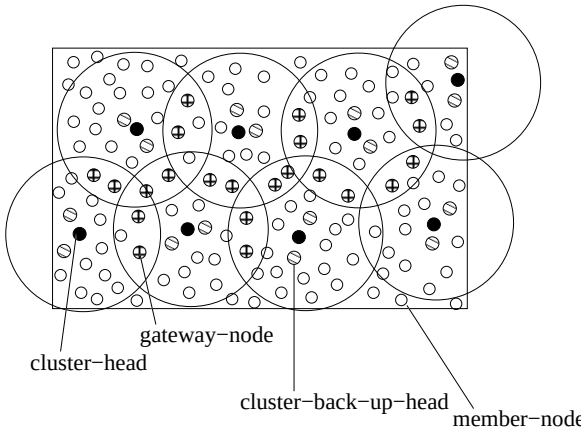


Figure 5. Cluster construction example

it will pick X as the backup cluster head since X is the closest one with the strongest signal strength. Node E, F, G, H, I, L, M, and N also receive JOIN messages. Since L satisfies all requirements of candidate cluster head and it has the maximum f around its neighbors, it declares itself as a cluster head. Then, L chooses I as the backup cluster head and K and J as the gateway and backup gateway respectively. Node E declares itself as a cluster head, and it picks G as its backup cluster head, B and C as gateways. As we can see, both cluster E and L are extended by cluster A, so they connect to cluster A through gateway B and K. However, cluster E and L also have an intersection and enough sensor nodes to be gateways. Suppose node M and J become cluster L's members firstly. Later when E declares itself as a cluster head, M and N will send a JOIN message again. Then, when I receives two JOIN messages from the same cluster E, it can specify M and N as gateways for cluster L and E.

A cluster construction result for a large network is shown in Figure 5. The communication cost of this clustering algorithm is more than the LEACH algorithm. However, once the clusters are established, the working phase is much longer than that of LEACH.

C. Cluster Maintenance

After the initial construction of clusters, in order to ensure that the stable phase is much longer, we need to maintain clusters dynamically. Our algorithm resolves two major issues of maintenance, inner reconstruction and local reconstruction.

Inner reconstruction is to activate the backup cluster-head or gateways to replace the failed cluster heads or gateways.

Local reconstruction is the process of constructing clusters in a local area when some cluster heads or gateways encountered energy shortage, with no ability to continue their work, and without any available backup to replace.

The two reconstruction processes are just performed in some local areas of the network, and only a few sensor nodes are involved in the process. The energy consumption cost is relatively small compared to reconstruct clusters globally.

Inner reconstruction includes cluster head replacement and gateway replacement. When residual energy of a cluster head is less than the user specified Threshold_Value, it will send a Replace_CH message to its backup cluster head. If there is no available backup cluster head, it will broadcast a Local_Reconstruction message to start a local reconstruction. When a backup cluster head receives a Replace_CH message, if it has the ability to be the cluster head, it must send a Replace_CH_Apply message to respond. Otherwise, it has to send a Refuse message the cluster head. If the cluster head receives Replace_CH_Apply message from its backup, it sends the cluster management information to the backup cluster head and informs the members of this cluster the node ID of the new cluster head. However, if the cluster head receives a Refuse respond, it will contact other backups. If no backup gives Replace_CH_Apply, it will broadcast a Local_Reconstruction message to start a local reconstruction. The gateway replacement is similar. Morever, all cluster heads which share the previous gateway should be informed.

D. Local Reconstruction

Usually, the work load for different areas are different. For example, clusters close to the sink will consume more energy since they are responsible for forwarding messages to the sink. If we reconstruct clusters for the entire network because of a minor topology change, it wastes too much energy. Also, global reconstruction might be very often. To solve this problem, we just reconstruct clusters for the local area which will be unavailable soon. The local reconstruction starts when a cluster head or gateway needs to be replaced but there are no available backups. The local reconstruction firstly identifies clusters which need to be updated and sets the members to idle state. Then, the initial cluster construction algorithmi invoked to establish clusters for this area.

Local Reconstruction Algorithm Description:

Step1: When a cluster head or gateway needs to be replaced because it encounters energy shortage but without available backup-nodes to replace, the cluster head broadcasts a local reconstruction message. The local reconstruction message includes the start time s of reconstruction and the hop which specifies the range for reconstruction. Then, it picks a neighbor with plenty energy to be the starter and informs it. The starter will declare itself as a cluster head at time s.

Step2: After receiving the local reconstruction message from a cluster head, its members set their states to idle and record the reconstruction start time. The Algorithm IV-B is invoked at time s on all idle sensor nodes. After receiving the local reconstruction message, the gateways set their states to idle and check if the hop number is more than zero. If so, the hop is reduced by one. Then, the local reconstruction message is forwarded to the nearby clusters with new hop.

Step3: Cluster-heads check if their clusters need to be reconstructed after receiving the local reconstruction message. If not, drop the message. Otherwise, we set all its members and itself's state to idle and check the hop in the message. If the hop is greater than zero, it would be reduced by one and the massage would be forwarded to the nearby gateways with new hop.

Figure 6(a) shows an example of before the local reconstruction. In cluster D, we assume that cluster-head D and backup cluster-head F have been severed as cluster-heads, backup cluster-head G is the current cluster-head. When G encounters energy shortage but without available backup nodes to replace, G broadcasts a local reconstruction message. After gateway-nodes L, M, N, O, P, Q, J, and K receive this message, they set their states to idle and reduce the hop number of this message by one, then forward updated message to neighbor cluster-heads A, B, C, and H (backup cluster-head H is the current cluster-head and cluster-head E and backup cluster-head H have been severed as cluster-heads). If cluster-head H agrees to participate local reconstruction but cluster-heads A, B and C not, cluster G and H participate the reconstruct process. In local reconstruction, bigger sensor's transmission range is needed because sensor nodes probably are sparse in that local area. Figure 6(b) shows the result after the local reconstruction is executed. D' and E' two clusters are generated in the original area.

E. Global Reconstruction

Global reconstruction will be launched by the sink when most of sensor nodes' energy is lower than the threshold value, or a considerable number of cluster heads and gateways almost lose their abilities. After the clusters work for a long term, frequent local reconstructions will happen. Since clusters generated by local reconstruction are not globally optimal, it will decrease the routing performance. At this time, a global reconstruction is better than frequent local reconstruction. The sink can launch a global reconstruction to replace the abundant local reconstructions, and this can keep an even clustering configuration and a better topology of the whole network. The global reconstruction should also be launched when a considerable number of sensor nodes lost their abilities. The communication radius should also be extended so as to avoid the situation of without gateways between adjacent clusters. In our simulation experiment, the global reconstruction would be launched when 80%of the sensor nodes' energy is lower than the threshold value or 30% or more sensor nodes died.

F. Routing

TABLE I.
LIST OF FUNCTIONS AND VARIABLES FOR ROUTING.

Ni:	i^{th}sensor node
N_i.cal_distance(dest):	return the distance from the node N_i to the destination node
Build_route():	node builds the route
N_i.next_node:	the next hop in routing table(from the destination area to the sink node)
N_i.pre_node:	the next hop in reverse routing table (from sink node to destination areas)
N_i.seek_ch_reverse(dest):	find its next hop cluster head in reverse routing
N_i.seek_gw_reverse(dest):	find its next hop gateway in reverse routing

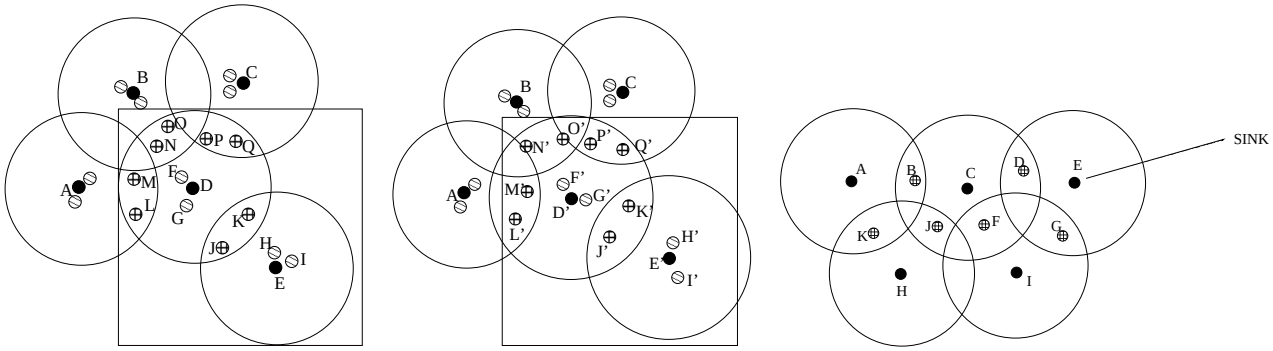
Algorithm IV-F describes the routing formation. The description of functions and variables are shown in Table I If there is no available route when the sink needs data of a destination area, it starts a route finding process. Firstly, the sink node broadcasts its own query message (including the coordinate information of the target cluster head). The receiver returns acknowledgement message to the sink. Then, the sink node calculates the distance between these cluster heads and the target cluster head and select the nearest one as the next hop (reverse routing). If no cluster head can communicate with the sink directly, it would broadcast the WANT_GW message and select the conjoint gateway as the next hop(reverse routing). The corresponding cluster would be elected as the next hop(reverse routing) of this gateway. Then, the cluster head finds its next hop gateway (reverse routing), and the gateway finds its next hop cluster head(reverse routing). Repeat like this until the message is sent to the destination node.

Some details are worth mentioning. The cluster head calculates the distances [18] between all its gateways and the destination node, finds the nearest gateway, and saves it as its next hop(reverse routing). The gateway calculates the distances between all its cluster heads and the destination node, finds the nearest cluster head, and saves it as its next hop(reverse routing).

G. The Route Maintaining Stage

1) The node replacement of a routing path.
   To ensure the successful data forwarding, the algorithm uses the reliable function to calculate the reliability of routing nodes. The function is:

$$R - reliability = \frac{(energr)^{\beta}}{(Er)^{\alpha} d^{\gamma}}$$



(a) Clusters before local reconstruction. (b) Clusters after local reconstruction. (c) Some routing nodes in our WSN.

Figure 6.

Algorithm 2 Routing Formation Algorithm

```
1: sink.broadcast(Interest_Msg).
2: while (Message ← Ni.receiveMessage())! = NULL do
3:    Message ← Ni.receiveMessage().
4:    Current_CH ← Message.source.
5:    if (Current_CH.cal_distance(destination_node) < min_distance then
6:       Ni.pre_node ← Current_CH.
7:       Current_CH.next_node ← Ni.
8:       min_distance ← Current_CH.cal_distance(destination_node)
9:    end if
10: end while
11: if sink.pre_node == NULL then
12:    sink.broadcast(WANT_GW).
13: end if
14: for all the nodes Ni receive the WANT_GW message do
15:    if (Message.source == sink) then
16:       Ni.next_node = sink.
17:       sink.pre_node = Ni.
18:       Ni.pre_node = Ni.ch.
19:    end if
20: end for
21: Currenr_node ← sink.pre_node.
22: while (Current_node! = detination_node) do
23:    if (Ni.role = CH) then
24:       Ni.seek_gw_reverse(destination_node)
25:    end if
26:    if Ni.role = GW then
27:       Ni.seek_ch_reverse(destination_node).
28:    end if
29:    Current_node ← Current_node.pre_node.
30: end while
```

R-reliability value is decided by residual energy, the energy cost in receiving data (per bit)Er, and the distance d between two nodes. α, β, γ are the balancing factors. Here, we use constant value [20], $\alpha, \beta, \gamma \epsilon (0, 1)$. When the reliable value is less than the threshold, the backup node would be launched. If there is no backup node for the particular node on the routing path, the node would claim its fault (no forwarding, no acknowledgement).

2) Avoiding fault nodes

Gateway fault: As seen in figure 6(c), if the current node is C, according to the established routing (A-B-C-D-E-Sink), its next hop is node D. If it cannot receive any reply from D, node C would choose a optimal path according to D's geographical information. For example, it would reach point F which is closer to the sink than itself. Then, we can find the route from F to the sink.

If the cluster head's next hop gateway node is unreliable, the cluster head finds another gateway as its next hop. Then, the new gateway finds its next hop. Repeat like this until the destination node is reached.

Cluster Head fault: As seen in figure 4, if the current node is B, its next hop is node C. If it cannot receive any reply from C, node B would return the message to A and send D's information to A. Then, A would choose an optimal path according to D's geographical information. This occurs when local reconstruct happens.

When part of the whole network's nodes consume too much energy while other fields do not have that much of consumption, the local reconstruction occurs. After that, some routing nodes become common nodes, and some clusters use the energy efficient backup nodes to replace the over burdened cluster heads. Consequently, we need to delete all the changing status nodes' information and initialize the next hop's ID. After the network's re-initialization, it would initial the routing process to find the next hop's ID and generate paths end to the sink. If the local reconstruction cannot satisfy the networks' performance demand, the network would begin its global reconstruction.

H. Data transmission

When the sink requires information of a specified region, it will send a request message to the cluster head of the target region. The cluster head collects the information from its member nodes in its cluster. When the data are sent back to the sink (reverse routing), we use the routing reliable function to choose a best path to ensure the reliability of data transmission.

Because of the heavy density of the sensor nodes' distribution and the relatively larger node sensing range, it

is not necessary that all sensor nodes sense data. However, in order to ensure the reliability of the monitoring data, we need to guarantee a certain number of sensor nodes are involved in monitoring task. Therefore, we set a threshold T (assuming 0.4). When a member node needs to collect data, it produces a random number $P \in [0, 1]$. A member node will participate the collection task if $P > T$, so this leads about 60% members to participate the work. The remaining members go to sleep to save energy. Thus, the energy consumption of the entire clusters is reduced significantly. Since 60% nodes are involved in the data collection, the reliability of monitoring data can also be guaranteed. The randomly selecting working nodes can balance the work load. As a result, it can also balance the energy consumption of the member nodes.

V. SIMULATION

In this section, we evaluate the performance of our Redundancy-based Reliable Directional Clustering Multi-hop Routing Algorithm ($CRMR$).

A. Settings of the parameters

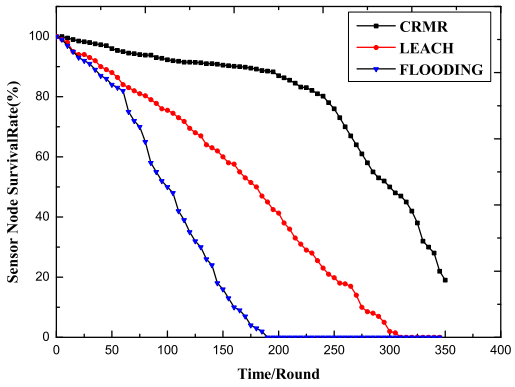
We distribute 300 nodes evenly in a 400m×200m area. Here, we assume that each node has $10J$ initial energy, and the energy consumption needed in sending or receiving is unified set as $Estatic$ = 50nJ/bit. In order to transmit data far enough, the amplifier's energy consumption is $Eamp$ = 100pJ/bit×m^2, and the size of each data packet is fixed to 64 bits. All nodes are static, and a node will die once its energy falls below the threshold. For better simulating the impact of the environment, the algorithm sets a very small probability of unexpected death when the nodes are working. The energy consumption of the sensor nodes for sending k bytes data is: $Esend$ = k×$Estatic$+k×Eamp×d^2; the energy consumption of the sensor nodes for receiving k bytes data is: $Ereceive$ = k×$Estatic$ [9]. In the simulation experiment of LEACH, we carry out some improvements on it: using the multi-hop routing mechanism; in order to avoid any blank areas which are not covered by any cluster, we set the generation rate of cluster heads T(n) to 0.08, which is a little higher than the optimum value.

B. The experimental results

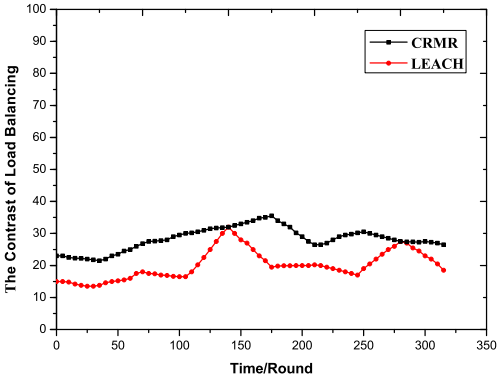
- The node survival rate.
  We respectively simulate our algorithm,LEACH and Flooding and obtain the nodes' survival rate with the working time. The results are shown in Figure 7(a). The horizontal axis in the figure is time, the vertical axis is the node survival ratio. The survival rate is decreasing with the increasing of network running time. Our CRMR is obviously better than LEACH and FLOODING. As seen in Figure 7(a), $CRMR$'s life cycle has been prolonged **30** compared with $LEACH$ and about 150 when compared with FLOODING. Moreover, $CRMR$'s death rate is smaller than $LEACH$ and FLOODING at the primal stage; however, at later stage, its death speed is faster than $LEACH$. This is because our algorithm relatively balances the energy consumption among sensor nodes, so only the nodes near the sink would die quicker than others because of fast energy consumption due to their heavy forwarding tasks. Other nodes' consumption status are similar, so there would be no big difference for most of the nodes' death time in the later period. This is why the figure shows the smooth curve and small slope in early stage, and the curve and slope become greater in the later stage.
- Load balancing.
  In hierarchical algorithms, beside the energy consumption for clustering, the average level of network coverage would also affect the network lifetime. The load balancing is one of the most significant criterion for evaluating the cluster's performance. The load balancing factor computing formula is:

$$[LBF] = [\frac{head - num}{\sum_{i=1}^{head-num} (x_i - \mu)^2}]$$

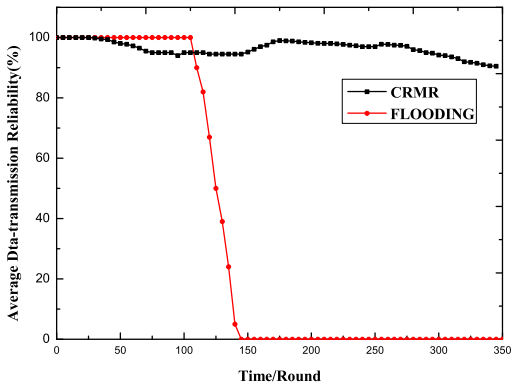
  In the formula, the $head - num$ denotes the number of clusters in this round, x_i denotes the node number in i cluster, and μ denotes the average number of nodes per cluster in this round. The larger the LBF, the better balance the load achieved. Figure 7(b) shows the load balance of $LEACH$ and $CRMR$. From Figure 7(b) we can see that the load balancing of $CRMR$ is better than $LEACH$, and the variation range is smaller. This means the load is more balanced in $CRMR$ since the topological structure is steady, so the LBF curve shape is more smooth. For LEACH, because the cluster organization process is repeated frequently, and the cluster head election occurs randomly, the LBF curve shape's extent is larger.
- Average reliability.
  Figure 7(c) shows the average reliability of the data transmission of flooding [6] and $CRMR$ when they work in the similar scenarios.
  As shown in Figure 7(c), the flooding algorithm is robust. The packet missing only occurs while the residual energy is too low. In the early period of data selection, since the energy is sufficient, we can hardly find the node fault, so that the packet missing hardly occurs as well. After a long while, flooding algorithm leads to a large energy consumption, so the packet missing rate grows sharply. When most of sensor nodes died, the nodes' data transmission rate turns into 0. The reliability of $CRMR$ algorithm keeps between *90-100*. In the process of data collection, the regional nodes' energy consumption is relatively higher because the nodes lose data in some ratio, so the successful rate of data transmission falls. However, along with the energy consumption, after the network topology reconstruction, the subsequent packets would transmit through other route which has sufficient energy. Consequently, the reliability



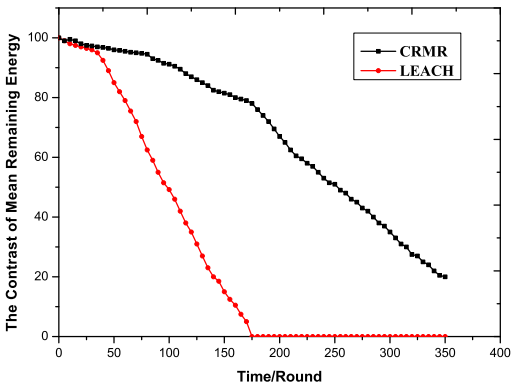
(a) Survival rate.



(b) Load balancing.



(c) Average data-transmission reliability.



(d) The contrast of mean remaining energy.

Figure 7. Simulation results.

level of data transmission might rally to a higher level, thus generating the performance curve like Figure 7(c).

- Average energy consumption.
  The average energy consumption denotes the average remaining energy of all sensor nodes in the network. The Figure 7(d) shows the comparison between flooding and $CRMR$. As shown in the figure, CRMR is much better then flooding. It is observed that there are a mount of repeating packets and transmission flow rate because the non-direction characteristics of flooding. This leads to a increasing energy consumption in the network. As a consequence, the performance of flooding is low.

VI. Conclusion

A novel clustering-based reliable multi-hop routing scheme ($CRMR$) is proposed in this paper. Clusters which can cover the entire network are constructed to support reliable routing. For fitting different network status, a local and global reconstruction algorithms are proposed to adjust clusters topology after initial cluster construction. Moreover, we also represent how to build routing tables and transmit data. Simulation results show that our scheme is better than LEACH and flooding algorithms.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under Grant No. 60903196, the Natural Science Foundation of Hubei Province of China under Grant No.2009CDB379, the Foundation of Jiangxi Educational Committee under Grant No.GJJ10661, and the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education Foundation under Grant No. AISTC2008_06 and AISTC2008_13.

References

[1] Limin Sun, Jianzhong Li, Yu Chen. Wireless Sensor Networks,Beijing: Tsinghua University Press 2005
[2] W. K, Bin Y, Wireless Sensor Networks:Architecture and protocol, Peking:E-industrual Press, 2007
[3] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica. Geographic Routing without Location Information,MobiCom'03. San Diego,California,USA.September14-19,2003

[4] Ganesan D, Govindan R, Shenker S, Estrin D. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. Mobile Computing and Communications Review (MC2R). 2002,1(2)
[5] Deb B, Bhatnagar S, Nath B. ReInForM: Reliable Information Forwarding using multiple paths in sensor networks, In: Proc 28th Annual IEEE Conf on Local Computer Networks (LCN), October 2003
[6] R. Farivar, M. Fazeli, S. G. Miremadi, "Directed Flooding: A Fault-Tolerant Routing Protocol for Wireless Sensor Networks", International Conference on Sensor Networks, Montreal, Canada, August 14, 2005
[7] M. Nassr, J. Jun, S. Eidenbenz , A. Hansson, and A. Mielke, Scalable and Reliable Sensor Network Routing: Performance Study from Field Deployment, Infocom, May 2007
[8] Stefan Pleisch, Mahesh Balakrishnan, Ken Birman, Robbert van Renesse, MISTRAL: efficient flooding in mobile ad-hoc networks, Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, May 2006
[9] Heinzelman W.R., Chandrakasan A., and Balakrishnan H., Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii International Conference on , vol., no., pp. 10 pp. vol.2-, 4-7 Jan. 2000.
[10] Yan Li, Xihuang Zhang, Yanzhong Li. LEACH-EE-Energy-Efficient clustering routing algor ithm based on LEACH, Computer Applications, 2007,05-1103-03
[11] M. P. Singh, M. M. Gore , "A New Energy-efficient Clustering Protocol for Wireless Sensor Networks", IEEE Communications Society, 2005
[12] Min Qin, Roger Zimmermann, "An Energy-Efficient Voting-Based Clustering Algorithm for Sensor Networks", the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel, 2005
[13] Shaoren Zheng, Haitao Wang, Zhifeng Zhao, Zhichao Mi, Ning Li. Adhoc network technology[M], Chapter V, Post and Telecom Press, 2005:972192, 10921111
[14] Zou Yi, Chakrabarty K.A distributed coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks[J].IEEE Transactions on Computers, 2005, 54(8) : 978- 990
[15] Huang H, Richa A W.Dynamic coverage in Ad- Hoc sensor networks[J].Mobile Networks and Applications, 2005, 10: 9- 17
[16] Ahmed N, Kanhere S S, Jha S.The holes problem in wireless sensor networks: a survey[J].Mobile Computing and Communications Review, 9(2) : 4- 18
[17] Xueqing Wang,Research on Connectivity and coverage of WSN,dissertation for the degree of PHD of Harbin Engineering University 2006
[18] Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In: International Conference on Mobile Computing and Networking Procee of the 6th annual international conference on Mobile computing and networking, New York: ACM Press, 2000
[19] Wen song, Du Rui-ying, A Hierarchical and Heterogeneous Reliable Routing Scheme in Wireless Sensor Network, Journal of Harbin Institute of Technology, Jan.2007
[20] VoB S. Steiner's problem in graphs: heuristic method. Discrete Applied Mathematics, 1992,40(1): 45-72
[21] D. Tian and N.D. Georganas, Energy Efficient Routing with Guaranteed Delivery in Wireless Sensor Networks, WCNC 2003, March 2003

Ruiying Du received her Ph.D. in information security from Wuhan University, China, and M.S. and B.S. degrees in computer application technology from Wuhan Technical University of Surveying and mapping, China, in 1994 and 1987, respectively. She is currently a Wuhan University Professor. Her main research interests include wireless communication network security and computer network security.

Chunyu Ai received her Ph.D. in Computer Science from Georgia State University in 2010, and Bachelor and Master of Computer Science from Heilongjiang University in 2001 and 2004, respectively. She is currently an assistant professor of Troy University. Her research interests include wireless sensor networks, database, and data security.

Longjiang Guo is an Associate Professor in the School of Computer Science and Technology at Heilongjiang University, China. He is now working in the Department of Computer Science at the Georgia State University with Dr. Yingshu Li and Dr. Raheem A. Beyah as a Research Scholar. His research interests include wireless sensor networks, data stream processing and data mining. He received his Ph.D. in School of Computer Science and Technology at the Harbin Institute of Technology in China advised by Prof. Jianzhong Li. Dr. Guo is the recipient of the Second Award of National Technical Advancement of China in 2005.

Jing Chen received his Ph.D. in information security from Huazhong University of Science and Technology, China, and B.S. degrees in thermal energy and power engineering from Wuhan University of Technology, China. He is currently a Wuhan University Assistant Professor. His technical research interests include wireless communication networks, sensor networks, and network security.

Jianwei Liu received his Ph.D. in communication engineering from Xidian University, China, in 1998, his M.S. and B.S. degrees in electronic engineering from Shandong University, China, in 1988 and 1985, respectively. He is currently a Professor of School of Electronic and Information Engineering of Beihang University. His current research interests include the security of wireless and mobile communication network and computer network. He is a senior member of the Chinese Institute of Electronics and director of the Chinese Association for Cryptologic Research.

Jing He is currently a Ph.D student in the Department of Computer Science at Georgia State University. She received her M.S. degrees from the Department of Computer Science at Utah State University. She received her B.S. degree from the Department of Electrical Engineering at Wuhan Institute of Technology, China. Her research interests include Optimization in Networks, Wireless Sensor Networks, Wireless Networking, Mobile Computing, and Artificial Intelligence.

Yingshu Li received her Ph.D. and M.S. degrees from the Department of Computer Science and Engineering at University of Minnesota-Twin Cities. She received her B.S. degree from the Department of Computer Science and Engineering at Beijing Institute of Technology, China. Dr. Li is currently an Assistant Professor in the Department of Computer Science at Georgia State University. Her research interests include Wireless Networking, Cyber-Physical Systems, and Phylogenetic Analyses.