REMUDA: A Practical Topology Control and Data Forwarding Mechanism for Wireless Sensor Networks[1)]

SUN Li-Min[1,2] YAN Ting-Xin[1,2] BI Yan-Zhong[1,2]

[1](*Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

[2](*Graduate School of the Chinese Academy of Sciences, Chinese Academy of Sciences, Beijing* 100039)

(E-mail: {sunlimin, tingxin03, yanzhong02}@ios.cn)

Abstract In wireless sensor networks, topology control plays an important role for data forwarding efficiency in the data gathering applications. In this paper, we present a novel topology control and data forwarding mechanism called REMUDA, which is designed for a practical indoor parking lot management system. REMUDA forms a tree-based hierarchical network topology which brings as many nodes as possible to be leaf nodes and constructs a virtual cluster structure. Meanwhile, it takes the reliability, stability and path length into account in the tree construction process. Through an experiment in a network of 30 real sensor nodes, we evaluate the performance of REMUDA and compare it with LEPS which is also a practical routing protocol in TinyOS. Experiment results show that REMUDA can achieve better performance than LEPS.

Key words Data forwarding mechanism, tree-based hierarchical topology, virtual cluster

1 Introduction

Data gathering is one of the most important application patterns in wireless sensor networks. The applications such as environment monitoring, military field surveillance, and accident alarming system can be treated as data gathering applications[1~3]. In these applications, two kinds of data reporting schemes are included: periodical data reporting and urgent events reporting. The former is delivered continuously with data aggregation while the latter should be reported to the destination as quickly as possible.

Wireless sensor network is a self-organizing network, in which sensor nodes form a network topology soon after they are deployed. Since it has a strong influence on the data delivery efficiency, an appropriate network topology is beneficial to the optimization of data delivery and the in-network data processing. In some application scenarios, routing protocols are very simple so that data gathering relies on network topologies to a large extent. Till now, the topology control mechanisms and routing protocols are always discussed separately. More attention shall be paid on a combined strategy.

We propose and implement a topology control and data delivery mechanism, which is called reliable multi-hop and unbalanced data gathering tree (REMUDA tree), for a wireless sensor network based large-scale indoor parking management system. A tree-like topology has the advantage that the paths are optimized. A cluster topology helps to deal with in-network data aggregation. REMUDA mechanism forms a topology which has both the advantages of tree topology and cluster topology in a self-organizing way.

In a parking lot management system, sensor nodes monitor occupation of each parking space and report events that vehicles get into or leave the parking spaces. The sensor nodes are deployed regularly and densely in the parking lot and the sink node is the gateway of the sensor network to the outside systems. This kind of parking lot management systems requires the topology control mechanism to be real-time, reliable and scalable. In REMUDA mechanism, one node has a stable link to its parent and has almost the least hop count towards the sink node. REMUDA brings as many nodes as possible to be leaf nodes and let them have a common parent node in order to construct a cluster-like topology for data aggregation. In addition, REMUDA can balance the size of the clusters by selecting the parent nodes dynamically.

The rest of this paper is organized as follows. Section 2 presents the previous works about data forwarding and topology control mechanisms for wireless sensor networks. Section 3 discusses REMUDA protocol in detail. Section 4 describes the implementation of REMUDA and Section 5 gives the

1) Supported by National Natural Science Foundation of P. R. China (60673178), National Basic Research Program of P.R. China (2006 CB 303000)

Received December 2, 2005; in revised form March 19, 2006

experimentation results and comparison with other protocol. Finally, we conclude this paper in Section 6.

2 Related works

In recent years, there have been extensive researches on routing and data delivery mechanisms for wireless sensor networks.

Directed diffusion[4] is a typical gradient-based data forwarding mechanism which can form a data-reporting tree in the network by flooding a query command message all over the network. The tree is optimized for faster data transferring from data sources to the sink node. Directed diffusion has a relatively heavy burden on communications as it not only needs to rebuild the topology periodically to maintain the gradient, but also has to maintain more than one path to sink node for each data source for transferring multi-copy message. GPSR[5] is a typical kind of geographic routing protocol, in which nodes use location rather than node ID as their identification. Each node chooses the neighbor node which has the minimum distance to the target position as the next hop. It is designed for general mobile ad hoc networks, and requires a location service to map locations. Some other routing protocols, such as SPEED[6], take the QoS problem into account. But in practical applications, the communication model is far more complicated than theoretical model, so these protocols cannot be utilized in practice directly. Multi-path routing is another approach to ensure the end-to-end reliability of data transferring. [7] presents a mechanism to generate several disjoint data paths from data source to the sink node. But the protocol cost of maintaining these data paths is considerably expensive in a multi-source and densely deployed network.

The topology control and maintenance is another critical aspect in wireless sensor network to enhance the efficiency of data transferring. GAF[8] is a topology control algorithm for ad hoc and sensor networks by generating a virtual grid architecture. The network area is divided into fixed zones and forms virtual grids. Inside each zone, nodes collaborate with each other to play different roles. GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity. So GAF needs many redundant nodes for replacement. Each node uses its location information to associate itself with a point in the virtual grid. LEACH[9] introduced a hierarchical clustering algorithm for wireless sensor networks. It randomly selects a few sensor nodes as cluster headers which compress data arriving from nodes belonging to respective cluster, and send an aggregated packet to the sink node in order to reduce the amount of information that must be transmitted. LEACH does not take the multi-hop data transferring into account, and the cost to maintain the cluster is relatively high. Some other topology control mechanisms, such as ASCENT[10], TopDisc[11], DRNG[12], mainly focus on the topology generation while the data forwarding is not in consideration. Besides, they have some strong assumptions which are hard to satisfy in practice.

Most of the protocols mentioned above still stay in the lab. The actual communication model also brings forward challenges to the protocols. In TinyOS, it implements a multi-hop data forwarding mechanism called LEPS. LEPS can build a minimum spanning tree in the network by periodically broadcasting a local topology information message. Each node chooses its parent by considering the least hop to the sink node first and the link quality secondly. When implemented in a densely deployed network, the topology generated by LEPS will keep changing fiercely and the protocol needs a large amount of message exchanging among sensor nodes.

3 Protocol description

REMUDA consists of three processes, which are topology construction, data forwarding and topology maintenance. The sink node generates and broadcasts a topology construction message to start the construction process. The message is spread out over the network to establish a data gathering tree, of which the sink node is the root. Each node chooses one of its neighbors with minimum hop and minimum node ID as the parent node. In the data forwarding process, data is sent from source node to its parent, and then forwarded to the sink node along the data gathering tree. In this process, some non-urgent data is aggregated during the forwarding process to decrease the traffic burden. In data forwarding process, each node checks the quality of link to its parent periodically to maintain the network topology. If the link quality is below some certain threshold, the node will re-select a new parent node.

3.1 Topology construction

1) Diffusion of the topology construction message

A topology construction message includes the fields of node ID, hop count to the sink node, and parent node ID. The hop count field of the message generated by the sink node is filled with zero. The sink node broadcasts a new construction message to start the topology construction process. Without hearing any construction message in a period, the sink node will rebroadcast the message again.

When receiving the first topology construction message, the sensor node waits for a certain time to receive more messages and records the senders as its neighbors. After that, it selects one of the neighbors as its parent and sets its own hop count as its parent′s plus one. Then the node generates a new construction message with its own information and broadcasts it. The parent selection algorithm is described in the subsection below.

2) Parent selection

In REMUDA, we have three criteria in parent choosing. They are listed below in descendent order of priority.

a) Reliability of the bidirectional communication link. The link between a node and its parent should have a satisfying communication quality to enhance the reliability in data transmission.

b) Hop count to the sink node. We tend to choose the neighbor which has the least hop count as parent, so the data gathering tree have the least depth to reduce the data delivery delay.

c) Node ID. We bring forward node ID as one of the criteria because it can make multiple nodes choose the same node as the parent node. This method can increase the out-degree of the non-leaf nodes and make more nodes as the leaf nodes. This structure is beneficial to data aggregation.

During the topology construction process, hop count and node ID play more important roles than link quality as it has not been fairly evaluated at this stage. After the topology tree has been constituted, the link quality becomes the most important factor to the re-selection of parents.

After the topology construction process, a data gathering tree is formed in the network. The depth of nodes on the tree is its least hop count to the sink node. As shown in Fig. 1, most nodes become leaf-nodes and multiple nodes in a local area choose the same node as their parent. This makes the network topology have the character cluster structure.

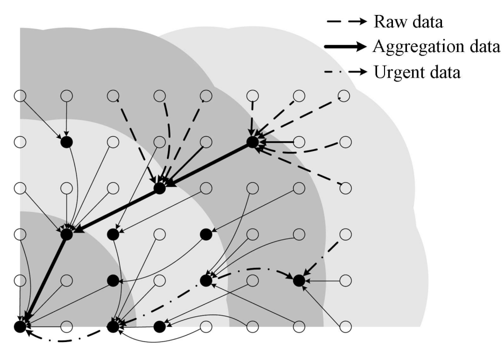


Fig. 1 Data delivery in a data gathering sensor network

3.2 Topology maintenance

1) Link checking

As the wireless channel is unstable and sensor nodes are easy to fail, the communication quality may degrade or even cause the link failure on the data gathering paths. In order to keep the communication reliability, each node carries out a periodical link checking procedure to monitor the quality of link to its parent. It selects a new parent to ensure the topology connectivity as soon as the link failure occurs.

The periodical link checking procedure is as follows. A node sends five checking messages to its parent at regular intervals and waits for the acknowledgements from its parent. Different checking messages are assigned with different weights. For example, the weight of the first message can be 1 and the weight of the fifth message can be 5 as it is fresher than the first one. If it fails to receive ACK from parent node, the weight value of the massage is zero. The sum of the five weight values will be considered as the link evaluation. If the evaluation is less than a threshold τ, the link is regarded as a failed one and the node will start a parent re-selection procedure described in next section.

If any data message is sent to the parent node during the checking interval, the data message can be considered as a check message. This can reduce the cost of the link checking mechanism. As the data traffic is heavier, the checking message in the network can be fewer.

2) Parent re-selection

If a node has no available parent node, it will re-select parent node by periodically broadcasting a Help message which includes its own node ID. When the node receives the first reply message, it starts a timer to wait for more replies. When the timer is fired, the node chooses one of the replied nodes as its new parent. The parent choosing mechanism is the same as that of parent selection described above. After that, the node broadcasts a notification message to inform its child nodes.

When a node receives a Help message not from its parent, it will send back a reply message including its node ID and hop count. If the Help message is from the parent node, it will mark a warning flag indicating that its parent turns to a parent re-selection process. If the node receives another Help message from its parent, it will give up the current parent node and perform a parent re-selection process. Otherwise, if it receives a notification message from its parent, it can conclude that its parent has found a new parent so it clears the warning flag. If its parent′s hop count becomes smaller at present, the node will only update its own hop count. It will start a parent re-selection procedure if its parent′s hop count is increased.

If one node joins a built network, it is just like a node loses the link to the parent. The new nodes will start a timer to wait for topology construction messages. If they can receive the messages before timeout, they will attach themselves to the topology construction process which is described in Section 3.1, or else they will carry out a parent re-selection process.

3.3 Data gathering

The messages reported by the sensor nodes in the parking lot consist of two kinds of data: urgent report and periodical report. When a car gets into or leaves from its parking space, an urgent message will be sent to the sink node. The occupation status of all the parking spaces is reported by the periodical message.

The data delivery in our sensor network is shown in Fig. 1. When one node generates or receives urgent report data, it will transmit the data to the parent directly. For the periodical report, the data will be aggregated into one packet in a virtual cluster before being delivered towards the sink node. A virtual cluster includes a non-leaf node as the cluster header and several leaf nodes that are the child nodes of the cluster head. At the beginning of the report period, leaf nodes in one virtual cluster send the occupation status to their parent separately. The cluster header waits for a certain time to collect all the report data in its cluster, aggregates all the data into one report packet and sends it out to the sink node. The aggregated packet will not be aggregated any more along its forwarding path.

3.4 Discussion

Energy efficiency is not the dominant factor in the indoor parking management system, but the protocol we proposed can also be utilized in many other energy-sensitive applications with a few modifications. For instance, when establishing the network topology, the topology construction message can carry an additional field of residual energy of the sender node and preferentially consider the residual energy while selecting its parent. Correspondingly, when nodes re-select their parents, an energy evaluation mechanism may be introduced to make nodes consume the energy of their parent candidates equally. Furthermore, periodical topology rebuilding mechanism can be added into the current protocol to optimize the data gathering paths.

4 Protocol implementation

We realize the REMUDA Protocol in TinyOS which is developed by U.C. Berkeley and is the most widely used software platform in wireless sensor networks. The hardware platform is developed by us and compatible with U.C. Berkeley MICA2 motes.

The software architecture is shown in Fig. 2. Three main components are REMUDA Protocol, Topology Report and Source Routing.

1) The REMUDA protocol component realizes the topology control and data forwarding mechanism in our protocol and also provides interfaces for other components to acquire the routing information. For example, when Data Gathering wants to send out its sensed data to the sink node, it will check this component to specify the node address of next hop.

2) Topology report component gathers local topology information and transfers it to the sink node. The topology information contains local node id, parent node id and hop count. With this information, the sink node can form the topology structure of the network completely. When node change its parent node or hop count, the REMUDA protocol component will signal a New Topology Info event to notify this component and it sends a topology report with the latest topology information.

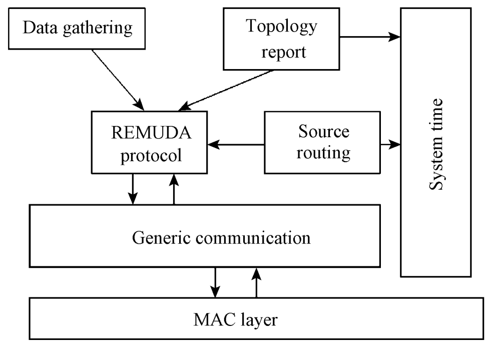


Fig. 2 Software components architecture in TinyOS with REMUDA protocol

3) Source routing component is used to measure the end-to-end delay between the sink node and certain node. The sink node sends out a command message to that node and it will give a reply immediately when receiving the command. Once the sink node sends out a command, it will record the time through System Time component. When it receives the reply, it will record the system time again. The time difference of the two time spots is regarded as the end-to-end delay from sink to that node.

5 Performance evaluation

5.1 Experiment environment

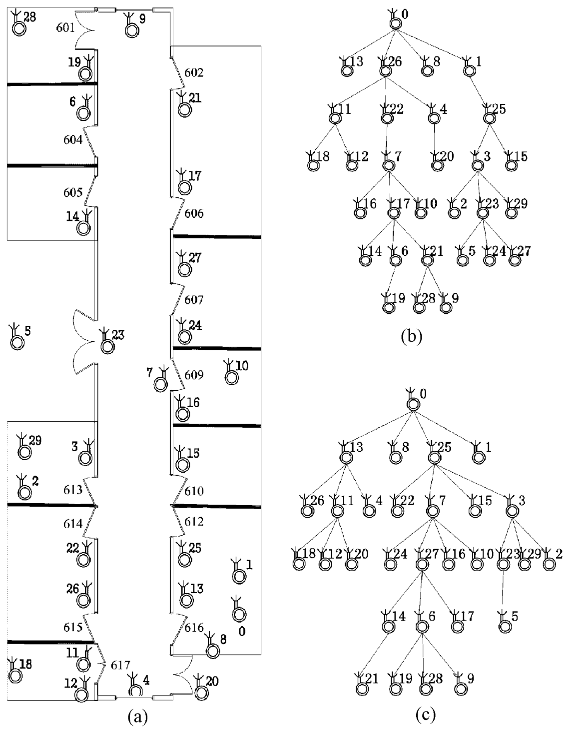
Our experiment is for the indoor parking lot applications based on a wireless sensor network where the nodes are densely deployed. We build up a testing network which consists of 30 sensor nodes and deployed in one floor of our lab building. With the obstacles like walls and separations, these nodes can form a multi-hop network. As the real communication model is much more complicated than the theoretical ones, the comparability is rare for our protocol to some ideal algorithms. In this section, we compare our REMUDA protocol with LEPS, which are both practical protocols and implemented in TinyOS.

Fig. 3 (a) illustrates the actual deployment of nodes in our lab building. Figs. 3 (b) and (c) are the logical network hierarchy generated in the experiment of REMUDA and LEPS separately.

5.2 Experiment result and analysis

As both REMUDA and LEPS are tree-based topology, we can use the parent change time to evaluate the frequency of the topology change, or the stability of the topology. As shown in Fig. 4, at the beginning of the experiment, both protocols have a fiercely changing topology, but REMUDA can quickly turn to a stable one and adjust it slightly from then on because in REMUDA, one node will not change its parent if it can communicate with the parent node stably.

Average data losing rate is to evaluate the reliability of the protocol. As shown in Fig. 5, at the beginning of the experiment when the topology is still under construction, the data losing rate is almost the same as both REMUDA and LEPS. As it is getting stable, the data losing rate of REMUDA is getting lower, but the LEPS still has the high losing rate due to instable topology.



(a) Actual nodes deployment (b) Logical structure in REMUDA (c) Logical structure in LEPS

Fig. 3 The deployment of nodes in our experiment

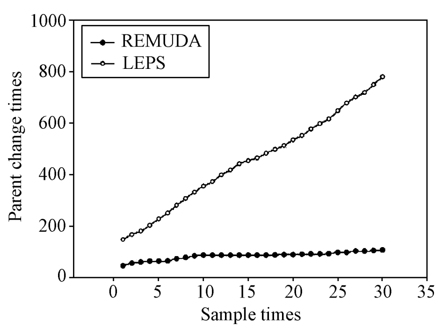


Fig. 4 Parent change time

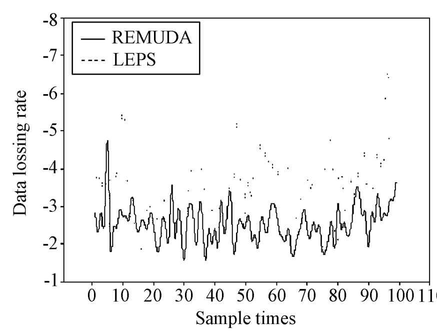


Fig. 5 Average data losing rate

Protocol cost is another important aspect to evaluate the efficiency of topology control protocols. In LEPS, every node periodically exchanges its neighbor information to other nodes by broadcasting mechanism, so the protocol cost increases in an almost constant speed as time goes by as shown in Fig. 6. But in REMUDA, the cost of check will be piggybacked by data reports, so the protocol cost is sharply cut down as the traffic burden increases.

The network topology construction time is a critical parameter to evaluate the convergence time of a topology generating algorithm. In our experiment, sensor nodes will notify the sink node as soon as it is connected to the network. We run the experiment 10 times and record the maximum, minimum and average construction time (Fig. 7). As REMUDA has a unique and active construction process, the construction time is much more stable and the average time is much shorter than LEPS′s.

Average Hop count is used to evaluate the distance from sensor nodes to the sink node, which denotes the data forwarding delay from data source to the sink node. As LEPS is the shortest path routing, it has the optimized hop count. However, the smallest hop count may lead to some unstable topology. As a result, REMUDA tends to find a stable topology with a little compromise on hop count, so it will increase its hop count in a stage and then goes to a stable situation. The experiment result of hop count sum is shown in Fig. 8.

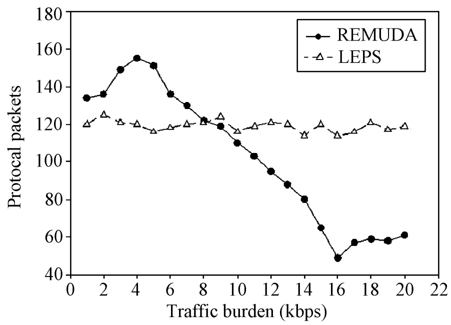


Fig. 6 Protocol cost

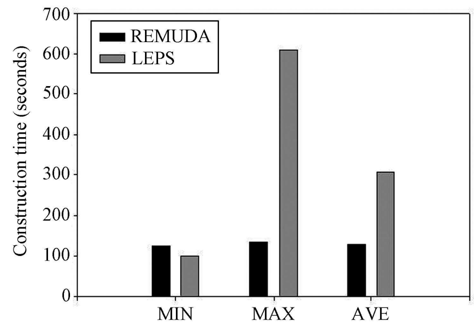


Fig. 7 Topology construction time

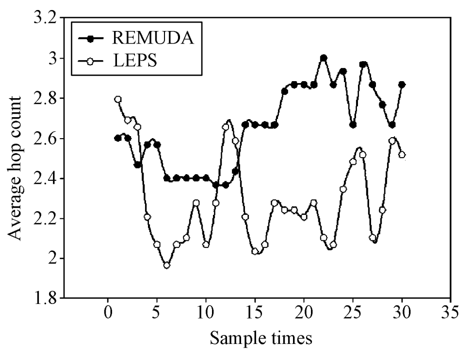


Fig. 8 Average hop count of the nodes in the network

6 Conclusion

In this paper, we present a practical topology controlling and data forwarding mechanism for wireless sensor networks. It is suitable for the data reporting applications such as environmental monitoring and parking lot, especially fit for the network with high density. It forms and maintains a spanning tree in the network with the consideration of communication reliability and topology stability. The nodes on the tree have the shortest paths to the sink node. And the clustered network topology is beneficial to in-network data aggregation.

REMUDA protocol is implemented as an independent component in TinyOS and can replace the LEPS routing protocol which is provided by TinyOS itself. Through the experiment on a network of 30 real sensor nodes, we evaluate the performance of REMUDA and LEPS. The results demonstrate that REMUDA can achieve higher reliability and stability than that of LEPS with a little compromise on the hop count. REMUDA also has a fast construction time and lower protocol cost than LEPS.

References

1 Akkaya K, Younis M. A Survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2005, **3**(3): 325∼349

2 Al-Karaki J N, Kamal A E. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 2004, **11**(6): 6∼28

3 Niculescu D, Americ N L. Communication paradigms for sensor networks. *IEEE Communications Magazine*, 2005, **43**(3): 116∼122

4 Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking. Boston, MA, USA: IEEE Press, 2000. 56∼67

5 Karp B, Kung H T. GPSR: Greedy perimeter stateless routing for wireless networks. In: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000). Boston, MA, USA: IEEE Press, 2000. 243∼254

6 He T, Stankovic J A, Lu C T F. SPEED: A stateless protocol for real-time communication in sensor networks. In: Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS 2003). Providence, RI, USA: IEEE Press, 2003. 46∼58

7 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): 295∼298

8 Xu Y, Heidemann J, Estrin D. Geography-informed energy conservation for *ad hoc* routing. In: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM). Rome, Italy: IEEE Press, 2001. 70∼84

9 Heinzelman W R, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 2002, **1**(1): 660∼670

10 Cerpa A, Estrin D. ASCENT: Adaptive self-configuring sensor networks topologies. In: Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies. Piscataway, USA: IEEE Press, 2002. 101∼111

11 Deb B, Bhatnagar S, Nath B. A topology discovery algorithm for sensor networks with applications to network management. DCS Technical Report DCS-TR-441. Pasadena, USA: Rutgers University, 2001

12 Li N, Hou J C. Topology control in heterogeneous wireless networks: Problems and solutions. In: Proceedings of IEEE INFOCOM, 2004. Hong Kong, China: IEEE Press, 2004. 232∼243

SUN Li-Min Professor at Institute of Software, Chinese Academy of Sciences. His research interests include wireless sensor networks and mobile IP networks.

YAN Ting-Xin Master student at Institute of Software, Chinese Academy of Sciences. His research interest includes communication protocols of wireless sensor networks.

BI Yan-Zhong Ph. D. candidate at Institute of Software, Chinese Academy of Sciences. His research interests include routing and MAC techniques of wireless sensor networks.