6

Routing Protocol with Unavailable Nodes in Wireless Sensor Networks

Deyun Gao, Linjuan Zhang and Yingying Gong
National Engineering Laboratory for Next Generation Internet Interconnection Devices, School of Electronic and Information Engineering, Beijing Jiaotong University Beijing 100044, P.R.China

1. Introduction

With the rapid development of modern microelectronic technology, wireless communication technology, signal processing technology, and computer network technology, wireless sensor networks (WSNs) has become one of the most important and the most basic technologies of information access (Jennifer Yick, 2008). WSNs have been widely used in military, environment monitoring, medicine care and transportation control. Routing protocol is one of the key support technologies in WSNs and the performance of routing protocols significantly impact the performance of the entire network (Khan & Javed, 2008).

In wireless sensor networks (WSNs), some unavailable areas often are formed because some sensor nodes become unavailable due to energy exhausted, congestion, or disaster (Fang et al., 2006; Jafarian & Jaseemuddin, 2008). Multi-path routing protocol is one of the mechanisms to solve or alleviate the above problems. Data delivery over multiple paths can help balance network load and extend the life time of entire network. Generally, multiple paths in the routing protocols can be classified into two categories: disjoint multiple paths and joint multiple paths (Ganesan et al., 2001). Disjoint multiple paths can be furthermore classified into node-disjoint multiple paths and link-disjoint multiple paths. In the node-disjoint multiple paths, each path is independent and has no affect on each other. Apparently, it is better to choose node-disjoint multiple paths for data delivery in the designed routing protocol if possible.

Most of multi-path routing protocols in wireless ad hoc networks are extended from classical single path routing protocols. For example, split multi-path routing (SMR) is based on the dynamic source routing (DSR) protocol and ad hoc on demand multi-path distance vector routing (AOMDV) extends the ad hoc on-demand distance vector routing (AODV) protocol (Lee & Gerla, 2001; Marina & Das, 2001). Similarly, as its special type, most of multi-path routing protocols in WSNs are extended from the ones in wireless ad hoc networks and at the same time take account of different factors such as energy, QoS, security, congestion, and etc. There are many papers to consider energy efficiency when designing multi-path routing protocols in WSNs (KIM et al., 2008). They mainly select multiple paths based on the link cost function consisting of both the node residual energy level and hop count. In (Huang & Fang, 2008), Xiaoxia Huang and Yuguang Fang proposed a probabilistic modeling of link state for wireless sensor networks. Based on this model, an approximation of local multi-path routing algorithm is explored to provide soft-QoS under multiple constraints, such as delay and reliability. Yunfeng Chen and Nidal Nasser proposed to select multiple paths between one

sink and multiple sources with the consideration of reducing collision occurred at nodes that are receiving and forwarding packets on behalf of the source nodes in order to improve QoS (Chen & Nasser, 2008). The same authors proposed an secure and energy-efficient multi-path routing protocol (SEER) (Nasser & Chen, 2007). Besides of using multiple paths alternately for communication between two nodes to prolong the lifetime of the network, SEER is resistive some specific attacks that have the character of pulling all traffic through the malicious nodes by advertising an attractive route to the destination. In (Toledo & Wang, 2006), Alberto Lopez Toledo and Xiaodong Wang proposed to use network coding to achieve an adaptive equivalent solution to the construction of disjoint multi-path routes from a source to a destination. It exploits both the low cost mesh-topology construction, such as those obtained by diffusion algorithms, and the capacity achieving capability of linear network coding. Jenn-Yue Teo, Yajun Ha, and Chen-Khong Tham proposed a heuristics-based interference-minimized multi-path routing (I2MR) protocol that increases throughput by discovering and using maximally zone-disjoint shortest paths for load balancing and a congestion control scheme that is able to adjust the loading rate of the source dynamically (Teo et al., 2008).

However, the existed multi-path routing protocols can not provide mechanisms to cross around the unavailable areas particularly during the routing building procedure or later data delivering procedure. Because in WSNs the states of sensor nodes or areas are changing due to many factors, it is important to consider all of these factors and situations when designing the routing protocols. In this chapter, we propose a new micro sensor multi-path routing protocol (MSMRP) to avoid crossing the unavailable areas based on the micro sensor routing protocol (MSRP) previously developed by us (Gao et al., 2009). We firstly define the unavailable areas that may be formed due to kinds of reasons such as energy exhausted, disaster and so on, which can be detected by kinds of sensors through some predefined settings. Then we design several new routing packets and routing tables to help building multiple paths based on the MSRP. In particularly, we propose a neighbor node table exchanging mechanism that can help build an alternate route around the unavailable areas and try to avoid the multiple paths intersect. When a sensor node becomes unavailable during the route reply (RREP) forwarding procedure, its precursor node will try to find the alternate route to forward the RREP to the destination with the help of above mechanism. It also can help balance the network load, improve the transmission efficiency and routing stability with multi-path transmission, which furthermore decreases the unavailable areas' forming and enlarging. Finally, we implement the proposed protocol in the real sensor nodes and set up a testbed to conduct detail experiments. The experimental results show that MSMRP can perform well to build up multiple paths to avoid the unavailable areas.

This chapter is organized as follows. Section 2 describes the MSRP routing protocol. Section 3 introduces the definitions of unavailable and available areas, and presents the details of the MSMRP including new added message formats and operation mechanisms. Section 4 introduces our developed sensor node's hardware architecture. Section 5 presents the software architecture, operation mechanisms of some standard interfaces of the connector module and adaptive data processing scheme. Section 6 shows the experimental performance results of WSNs implementing MSMRP. Some important conclusions are drawn in Section 7.

2. Micro Sensor Routing Protocol

Based on AODV, we designed Micro Sensor Routing Protocol (MSRP)for IEEE802.15.4 based sensor network. In the following, we firstly describe the protocol stacks of IPv6 sensor node designed. Then, we present the details of MSRP.

2.1 Protocol Stack of IPv6 sensor node

Fig. 1 shows protocol stack of IPv6 sensor node designed by us. We divide the protocols into fiver layers including application layer, network layer, adaptation layer, data link layer and physical layer. Considering scare resources we simplify the traditional transportation layer (TCP and UDP) and merge them into network layer. Also, we put our MSRP routing protocol into network layer. Specially, we add a new adaption layer. For other layers, it is easy to understand their functions and we do not need to introduce them. Here, we just describe the adaptation layer.

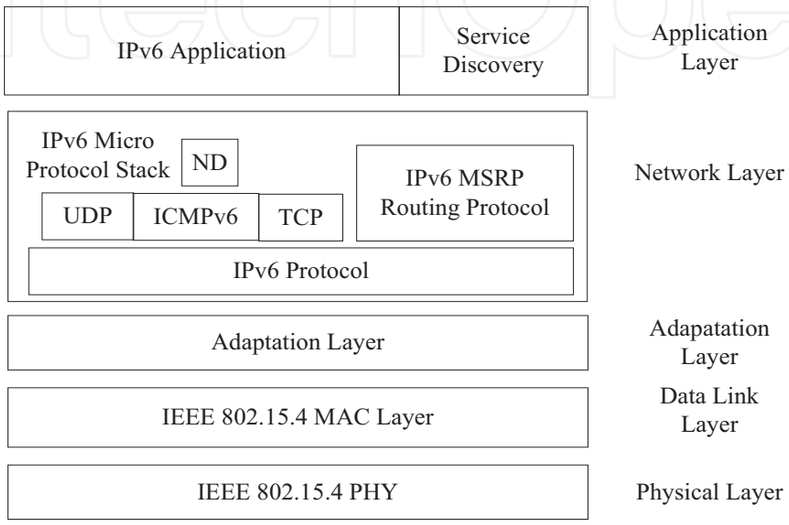


Fig. 1. Architecture of IPv6 Wireless Sensor Node

The adaptation layer lies between IEEE 802.15.4 MAC layer and the network layer. Adaptation layer is used mainly for fragmentation and reassembly. As we use IPv6 in the network layer, the maximum transmission unit (MTU) size for IPv6 packets over IEEE802.15.4 is 1280 octets. However, a full IPv6 packet does not fit in an IEEE802.15.4 frame. IEEE802.15.4 protocol data units have different sizes depending on how much overhead is present. Starting from a maximum physical layer packet size of 127 octets and a maximum frame overhead of 25, the resultant maximum frame size at the media access control layer is 102 octets. Link-layer security imposes further overhead, which in the maximum case (21 octets of overhead in the AES-CCM-128 case, versus 9 and 13 for AES-CCM-32 and AES-CCM-64, respectively) leaves only 81 octets available. This is obviously far below the maximum IPv6 packet size of 1280 octets, and in keeping with Section 5 of the IPv6 specification (Deering & Hinden, 1998), a fragmentation and reassembly adaptation layer must be provided at the layer below IP. Furthermore, since the IPv6 header is 40 octets long, this leaves only 41 octets for upper-layer protocols, like UDP. The latter uses 8 octets in the header which leaves only 33 octets for application data. Thus, there is a need for a fragmentation and reassembly layer.

2.2 Micro Sensor Routing Protocol Packet Format

In order to reduce low-speed IPv6 WSN equipment energy consumption, it is very important to design efficient and streamlined routing protocol packet formats. Considering low-speed wireless network characteristics, we designed our routing protocol with three routing packet formats including routing request (RREQ), routing reply (RREP) and routing error (RERR). We

do not use a Hello mechanism for route maintenance, thereby reducing the routing packet size sent in establishing new routes and maintaining them, which will reduce energy consumption. In the following we take a more descriptive look at these three packet formats.
Fig. 2 and 3 show the Route request packet format and the Route reply packet format respectively.

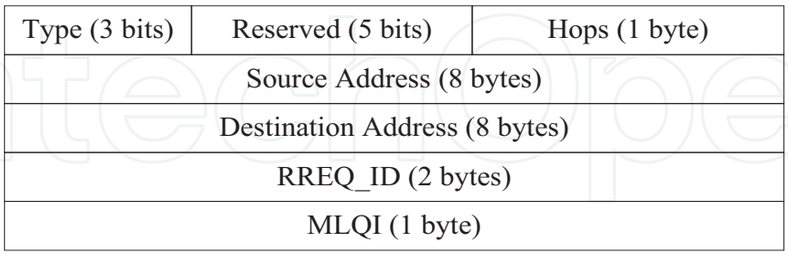


Fig. 2. RREQ Packet Format

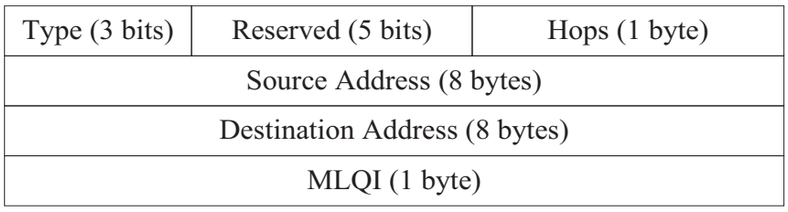


Fig. 3. RREP Packet Format

The fields in these two packets are the followings.

- Type: 000, 001 for RREQ and RREP message types respectively;
- Reserved: Reserved for future enhancements;
- Hops: Number of nodes RREQ or RREP messages passed from the corresponding source to current Node;
- RREQ_ ID: Unique identifier of RREQ message;
- Source Address: Address of the node which initiated RREQ or RREP;
- Destination Address: Requested route destination node address or Address of the node which initiated RREQ;
- MLQI: Minimum of the Link Quality Indicator (LQI) values between RREQ or RREP source to current node.

Fig. 4 illustrates the route error message format.

- Type: 010, for the Route Error (RERR) message format type;
- No. of Addresses: Number of neighbors which became unreachable as detected by the RERR originator node;
- Unreachable Destination Address n: Addresses of nodes unreachable (Number of addresses depend on "No. of Unreachable Addresses" field, in order to comply with IEEE802.15.4 standard, a IEEE802 .15.4 the size of data packets is not more than 128 bytes, hence one Route Error (RERR) message may Carry up to 4 unreachable addresses);

Type (3 bits)	No. of addresses (2 bytes)	Hops (1 byte)
Unreachable Destination Address 1 (8 bytes)		
Unreachable Destination Address 2 (8 bytes)		
...	...	

Fig. 4. RRER Packet Format

2.3 Routing Tables

Fig. 5 illustrates a routing table entry.

Type (1 bit)	Reserved (7bits)	PAN ID (2 bytes)	
Hop Limit (1 byte)		Time to Expire (1 byte)	Route LQI Value (1 byte)
Destination Address Interface ID (8 bytes)		Next-Hop Address Interface ID (8 bytes)	
Precursor Node Address Interface ID 1 (8 bytes)		Precursor Node Address Interface ID 2 (8 bytes)	
...		.	

Fig. 5. Routing Table Entry

- Type: Used for distinction between two types of equipments: Cluster head (0) and the cluster members (1);
- PAN ID: PAN (Personal Area Network) identifier;
- Hop Limit: No. of hops for this route;
- Time to Expire: The time of the expiration or deletion of this route entry;
- Route LQI value: Minimum LQI value of the Route;
- Destination Address Interface ID: Interface identifier(IEEE 64bit) of the destination node;
- Next-Hop Address Interface ID: Interface identifier(IEEE 64bit) of the next-hop of the route;
- Precursor Node Address Interface ID: Interface identifier(IEEE 64bit) of the previous node in the route (possibly more than one, used to send RERR messages);

In IPv6 WSN, routing protocol must avoid routing loops, reduce invalid data packets, effectively record routes and dynamically adapt to the changes in network topology and improve the information transmission efficiency. In our Micro Sensor Routing Protocol (MSRP) when a source needs to send data packet to unknown destination it will encapsulation and broadcast a RREQ packet. But the intermediate nodes will receive multiple instances of this RREQ packet through multiple paths. If the intermediate node broadcasts each time when this type of RREQ is received, this will create broadcast storms, which will affect the network performance and by increasing energy consumption of nodes it will decrease the network life time. Therefore we use a mechanism which involves a duplicate routing table. Dupe table will be inserted with the RREQ message information with the unique RREQ_ID. If another RREQ

message arrived from the same source (through a different path) with the same RREQ_ID before the entry expiration time, this new packet will be dropped. This mechanism effectively reduces overhead on the network at route establishment phase. Fig. 6 shows a dupe table entry.

RREQ Source Address (8 bytes)	RREQ_ID (2 bytes)	Time to Expire (1 byte)

Fig. 6. Dupe table entry

- RREQ Source Address: Address of the node which initiated one RREQ message;
- RREQ_ID: Unique identifier of RREQ message;
- Time to Expire: The time of the expiration or deletion of a route;

2.4 Route Selection and Decision Making Process of MSRP

MSRP is actually an on-demand routing protocol. When there is a need to send data to a destination, source node launches routing search process to find the corresponding route. This kind of on-demand routing protocol overhead is reduced and suitable to IPv6 WSN with energy saving requirements.

2.4.1 Sending RREQ

In IPv6 WSN, when a node needs to send data to another destination node, first search the local routing table, if no entry to the destination exists, cache current data packets and create RREQ packet, then broadcast the RREQ.

2.4.2 When intermediate nodes receive a RREQ

First when a intermediate node receives a RREQ message, it checks if the destination address is itself, if not, first check its dupe table. If there already exists a similar entry, that means this node received a RREQ from the same source with the same RREQ_ID, in order to reduce LR-WPAN energy consumption and broadcast storms, this duplicate RREQ is dropped. If no entry exists in the duplicate table, route to the source is added to routing table, then if there exists a route to the source, compare the two routes and store the optimum route. If there is a route to RREQ destination then unicast the RREQ to its destination, otherwise broadcast the packet.

2.4.3 When the destination node receive a RREQ

If the node detects that the destination address of RREQ equals its own, then it enters route reply process:
First of all node will put RREQ message in a cache table. Because in the RREQ path determining process, RREQ messages are broadcasted through network and hence the node might receive multiple RREQ messages through multiple paths, as a result it is necessary to wait for a reasonable period of time T, afterwards we apply the route determining function $f(m,h,n)$.

$$f(m,h,n) = Am + Bh + Cn \tag{1}$$

where m is the number of nodes with insufficient energy from source to destination, h is the number of hops from source to destination, n is the number of links with weak LQI between

source to destination. A, B, C parameters are to be determined under different network environments. A, B and C are values produced by non negative integer powers of 2, and must meet the condition $A >> C > B$. For example, in the open area network environment, we can use $A = 256$, $B = 1$, $C = 2$.
Destination node will calculate f value for each RREQ received through different routes from source. Then it will compare f value for current route to source if there is an entry in the routing table, then choose the entry with the lowest f value and start the reply process. Afterwards it unicasts a RREP through optimum path to RREQ source.
Fig. 7 describes the receive route request process.

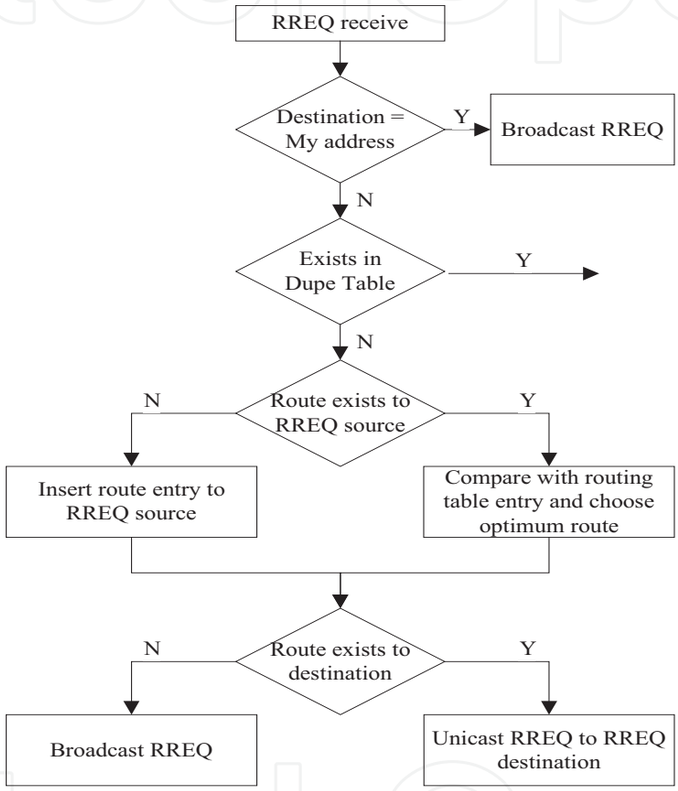


Fig. 7. Receive routing request

2.4.4 Receive RREP

When nodes receive RREP message, the first objective is to determine RREP destination address is itself. If it has established a routing entry to RREP source in the routing table, send the data items in the transmission buffer. Otherwise, it inserts or updates the RREP source address route entry, searchs routing table for the route to RREP destination, and then unicasts RREP towards its destination.

2.5 Route Maintenance and Error Handling Process

Micro Sensor Routing Protocol does not use the traditional maintenance methods like AODV Hello messages. Furthermore IEEE802.15.4 standard uses ACK frames to determine neighbor node reliability, if you do not receive an ACK in certain period of time after sending data,

this means that the neighbors nodes had expired, then save current data to a buffer, once again start the RREQ process, at the same time send a RERR to the Precursor node. It has the advantage of reducing energy consumption and network resource usage of sending and receiving Hello messages, on the other hand low-rate WPAN equipment usually are not very delay sensitive. As an on-demand routing protocol MSRP more effectively performs route maintenance.
When a neighbor node failure is detected, first find routing entries with that node address as the next-hop address. Then get their precursor node address and encapsulate a RERR message, unicast the precursor nodes with RERR, then delete the Routing table entries with that node as the next-hop address from the routing table. When a precursor node receives a RERR message, similarly process unreachable entries in the routing table, until all precursor nodes in this route has been informed about the route expiration.

3. Micro Sensor Multi-Path Routing Protocol

In this section, we firstly define unavailable areas that are formed due to the occurrence of unavailable sensor nodes, which can not provide data forwarding any more. Then, we introduce the MSMRP operation procedures and key modules.

3.1 Available and Unavailable Areas in Wireless Sensor Networks

In wireless sensor networks, the data delivery over some areas are unfeasible maybe because in this area the energy of sensor nodes are exhausted, or there are serious network congestion, or blind spots in the coverage area, or there are sudden disasters where even some nodes are destroyed. We can define such an area as **unavailable area**. Otherwise, the area where the data delivery can be completed feasibly can be defined as **available area**. Fig. 8 shows an example. In the figure, the red area is marked as unavailable area and the remain area is available area.

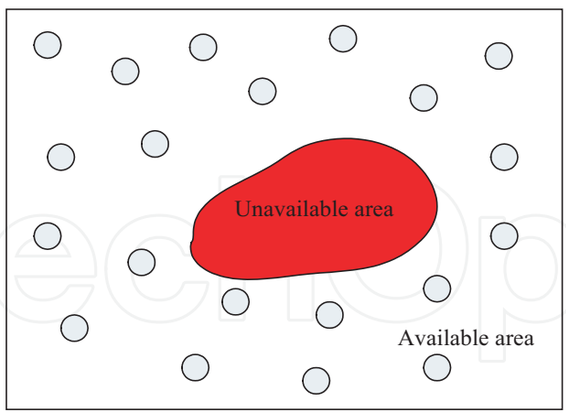


Fig. 8. Available and unavailable areas in wireless sensor networks

These unavailable areas forms because the sensor nodes in these areas becomes unavailable for data delivery. Furthermore, we can divide these unavailable sensor nodes into two categories. **The first category of sensor nodes** are located in the unavailable area. **The second category of sensor nodes** are located in the boarder of the unavailable area and they only have one neighbor node. After it receives the data frame from its neighbor node, it can not

forward out this frame. Each sensor node has a flag bit that is set to "disable" when it becomes a unavailable node.

3.1.1 The First Category of Sensor Nodes

For the classification of the first category of sensor nodes, we can make judgement according to the sensor node's energy, sensory data, network congestion status, and so on under different situations.

For example, the battery voltage is 1.15∼3.7V in our developed sensor node. If the battery voltage of a sensor node is lower than 1.15V, it can not work. Thus, when a sensor node's battery voltage is lower than 1.5v, we should decrease their work load and set it to a unavailable node. In the sensor node, we use a 8-bit digit to represent the energy value and 1.15V∼3.7V can be converted to 0∼255 as shown in Eq. (3).

$$B = (A - 1.15)\frac{255}{3.7 - 1.15} \tag{2}$$

where, A is the battery voltage of a sensor node and B is the converted energy value. When $A = 1.5$V, B is 35, i.e., when the energy value of a sensor node is lower than 35, we set it to a unavailable node.

For the temperature, we assume that the temperature is T, the digital value representing the temperature is SO_T.

$$T = d_1 + d_2 * SO_T \tag{3}$$

where, d_1, d_2 and SO_T of the temperature sensor SHT11 are shown in Tabs. 1 and 2

VDD	$d1[^\circ C]$
5v	-40.00
3.5v	-39.66
3.3v	-39.636
2.5v	-39.55

Table 1. Relationship between VDD (voltage drain drain) and d_1

SO_T	$d2[^\circ C]$
14bit	0.01
12bit	0.04

Table 2. Relationship between SO_T and d_2

For example, we assume that there occurs a disaster such as fire if the temperature is larger than 60°C. If the current VDD is 3.5V, the 14-bit digital SO_T can be calculated as 9966, i.e., if the temperature value in the sensor node is larger than 9966, we set the sensor node as a unavailable node.

We use the number of route entries, queue length, and frame sending rate to judge whether the network load is heavy with three threshold values for them. If any one is larger than the corresponding threshold value, then we think the sensor node is congested and set it to a unavailable sensor node.

3.1.2 The Second Category of Sensor Nodes

After a unavailable sensor node judged with the above mechanism is set, it will immediately send a removing message to its neighbor nodes. The sensor nodes receiving the message will remove the corresponding entries in the neighbor node table and check the number of its neighbor nodes. If it has only one neighbor node in the table, then it will be set to an unavailable node. This kind of unavailable nodes is the second category of sensor nodes.

3.2 MSMRP Routing Protocol Packets and Routing Tables

In order to minimize the energy consumption of the sensor nodes, it is very important to design space efficient and concise routing protocol packet formats and routing tables.

3.2.1 Routing Protocol Packets

In MSMRP, the used routing packets include routing request (RREQ), routing reply (RREP), routing error (RERR), HELLO message, advertisement message of neighbor node table and delete message of neighbor node. Among them, the former three kinds of routing packets can be referred in the MSRP routing protocol (Gao et al., 2009). The latter three kind of messages are explained in the followings.

The first new added type of packet is HELLO message, which includes three fields and is shown in Tab. 3. The field "Type" distinguishes message type. Here it specifies "011" for the HELLO message. The second field "Reserved" is reserved for future enhancements. The third field "Address" is the address of the sensor node that sends out this HELLO message.

Type (3 bits)	Reserved (5 bits)	Address (2 bytes)

Table 3. HELLO message format

The second new added type of packet is the advertisement message of neighbor node table (NDAD), which includes three fields at least and is shown in Tab. 4. The field "Type" specifies "100" for the NDAD message. The second field "Reserved" is reserved for future enhancements. The third field "Address of neighbor node 1" is the address of the first neighbor node. If it includes one more neighbor nodes, their address can be included into the following fields.

Type (3 bits)	Reserved (5 bits)
Address neighbor node 1 (8 bytes)	
Address neighbor node 2 (8 bytes)	
...	

Table 4. Advertisement message of neighbor node table

The third new added type of packet is the delete message of neighbor node (NDDE), which includes three fields and is shown in Tab. 4. The field "Type" specifies "101" for the NDDE message. The second field "Reserved" is reserved for future enhancements. The third field "Address" is the address of the sensor node that sends out the NDDE message.

Type (3 bits)	Reserved (5 bits)	Address (8 bytes)

Table 5. Delete message of neighbor node

3.2.2 Routing Tables

Each sensor node maintains a local routing table (LRT) for packet forwarding and a duplicate (DUPE) routing table to detect duplicate RREQ messages to avoid excessive flooding or control messages during the route discovery process. DUPE table will be inserted with the RREQ message information of a unique RREQ_ID. If another RREQ message arrived from the same source through a different path with the same RREQ_ID before the entry expiration time, this packet will be dropped. LRT and DUPE can also be referred in the MSRP routing protocol (Gao et al., 2009). Another new added table is neighbor nodes table that includes the address of neighbor nodes and is shown in Tab. 6.

Address of neighbor node 1 (8 bytes)	Address of neighbor node 2 (8 bytes)	...

Table 6. Neighbor nodes table

3.3 Operation Procedure of MSMRP

The basic operation procedure of MSMRP includes the followings:

- When the node starts up, it will broadcast a HELLO message to the neighbor nodes. If one neighbor node receives the HELLO message, it will build up a neighbor nodes' table.
- When the source wants to send the collected information to the sink node, it will broadcast a RREQ message. The middle nodes, which received the RREQ message, will broadcast this message until it arrives at the sink node finally.
- After the sink node receives multiple RREQ messages from different nodes, it will select N paths with the minimal hops and save them to a route request table, then send back two route reply messages.
- When a middle node receives the first route reply, it will set a mark in the route entry, which means it is already a forwarding node along the first route.
- If the node receives the second RREP with the same destination node later, it will start the neighbor node table exchanging mechanism to find out a common neighbor node with the previous hop node. And it will set the neighbor node as the next hop, to continue forwarding the second RREP message.
- If the source node receives two RREP messages, it will randomly select a route to send the data or use two routes to balance the network load.

3.3.1 Route Discovery Procedure

As the MSRP, during the route query phase, the source node will broadcast a RREQ message as shown in Fig. 9. When the RREQ is forwarded to a unavailable node, it will be discarded directly. Thus, the inverse route to the source node is not built and later the RREP message sent from sink node will not traverse the unavailable node.

3.3.2 Route Reply Procedure

After some RREQ messages traversing different paths arrive at the sink node finally, the sink node will select two best optimal path to send out the RREP messages, which will arrive at the source node along the inverse route built in the route query procedure.

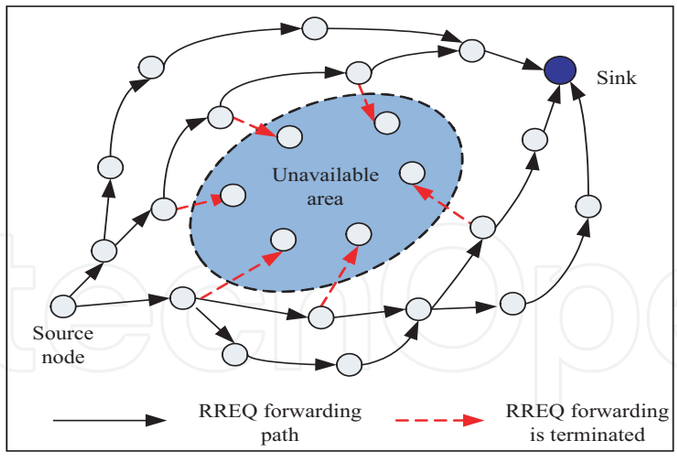


Fig. 9. RREQ message forwarding procedure

The problem that may arise is that there appears a unavailable node along the inverse path. The neighbor nodes of the unavailable node including the previous hop along the inverse path will remove its corresponding route entries, i.e., the inverse path is broken here. The sensor node on previous hop will cache the RREP and try other neighbor node one by one. If its neighbor node has a path to the source node, then it will send ACK to it and the RREP can be forwarded to the source node along a new inverse path. If there is no any neighbor node that has a path to the source node, then it will send NO message to the sink node. And the sink node will select another best path from the remaining paths to send out a RREP message again. The route reply procedure is shown in Fig. 10.

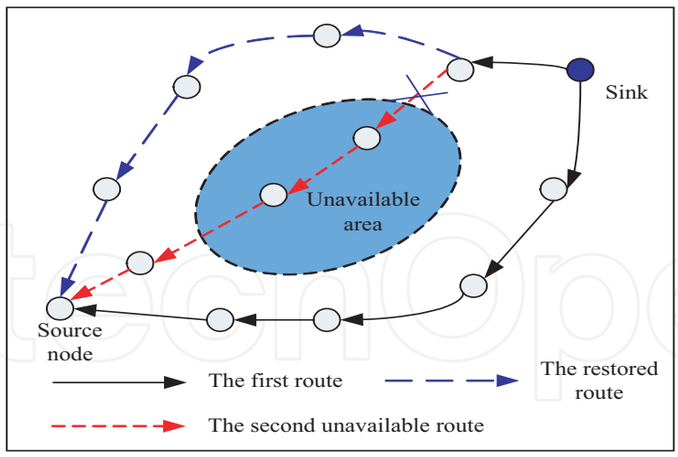


Fig. 10. Route Reply Procedure

3.3.3 Route Maintenance and Error Handling Process

After the route is built up, the source node can send the data along the route. If later some sensor nodes along the route become unavailable nodes, then the data forwarding can not be completed. Therefore, we need to design route maintenance and error handling mechanism to deal with it.

In the RERR process, when a node is notified by the neighbor unavailable node or knows some neighbor node becomes unavailable if its HELLO message dose not been received during a period, it first searches routing entries with that node address as the next-hop address in its LRT. Then get their precursor node address and encapsulate a RERR message, unicast the precursor nodes with RERR, then delete the routing table entries with that node as the next-hop address from the routing table. When a precursor node receives a RERR message, similarly process unreachable entries in the routing table, until all precursor nodes in this route has been informed about the route expiration. In the case where a particular precursor node that also becomes unavailable is detected, our design triggers no further RERR for energy conservation.

3.4 Neighbor Node Table Exchanging Mechanism

In the MSMRP, the multiple paths that can avoid crossing the unavailable area are built with the help of neighbor node table exchanging mechanism.

3.4.1 Neighbor Node Table Building

A node sets up its neighbor node table through the HELLO message broadcasting periodically. When a node starts up, it broadcasts a HELLO message to its neighbors and the node that receives the HELLO message will search its neighbor node table. If the node address information does not exist in its neighbor node table, then it adds the address into its table. If the address exists, then just ignores the HELLO message. This table can be used to help ensure multiple paths disjoint.

3.4.2 Neighbor Node Table Exchanging

After an intermediate node receives a RREP message, it will firstly check up its flag bit to see whether it is an effective node. If it is an unavailable node, it will discard the RREP message directly. Otherwise, it will check whether it is the first received RREP message. If so, it will build up an inverse route to the sink node. Otherwise, it becomes a joint node between two routes. At this time, it will start up the neighbor node table exchanging mechanism to deal with this situation.

When an intermediate node finds out that it is the joint node between two paths after it receives the second RREP message, it will send out its neighbor node table to the precursor node that forwarded the second RREP message to it. The precursor node will find out the common neighbor nodes of them through comparing the received neighbor node table and itself. Following that, the precursor node will send the RREP to a common neighbor node selecting from them. If the selected common neighbor node has a route to the destination node of the RREP message, it will send back an ACK to the precursor node and continue forwarding the RREP message. If the precursor node does not receive the ACK, then it will select another common neighbor node to try again with the repeated procedure. Finally, the second RREP message will arrive at the destination node along the second changed route. However, there maybe not has any common neighbor node that has a route to the destination node. Under this situation, the precursor node will notify the joint node and the joint node will forward the RREP message by itself. Thus, this kind of mechanism can try best to avoid the intersection between two paths.

4. Hardware of the Sensor Node

The designed sensor node should has strong extensibility and is adaptive to the new applications with slight secondary development work. It can improve node flexibility and computation performance.

4.1 Basic Architecture of the Hardware

Fig. 11 shows the hardware architecture of the designed sensor node. The sensor node is divided into main board and expansion board. Processor, wireless communication, power and connector modules, which are the common components for most of applications, are located in the main board. Sensors, memories and other modules, which are changed usually for different applications, are put into the expansion board. Thus, when we reconstruct the sensor node according to users' requirements, this kind of design can reduce the secondary development time and cost.

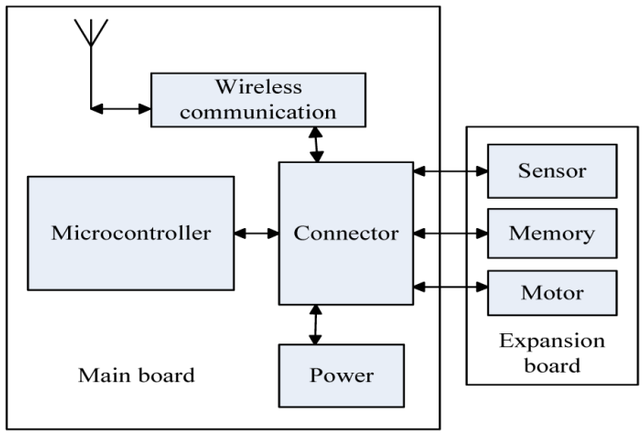


Fig. 11. Hardware architecture of the sensor node

In the following, we briefly introduce each module.

- In the general purpose multi-sensor node, the connector module is responsible to connect other modules and expansion board that is composed of some sensors, memory, and others. If we want to extend any module, then we just need to redesign this module and connect to the connector module. Thus, it is very convenient to extend the functions of sensor node for new applications.
- Microcontroller module is the key of the sensor node, which is in charge of controlling the node, sensor data processing, and etc. It can use the connector module to delivery commands to any module in the node.
- Wireless communication module sends or receives information over wireless link. It is connected to microcontroller through the connector module with SPI interface.
- The sensor module senses and collects the environment information with different type of sensors, and sends these collected information to the microcontroller through the connected module with some standard interfaces.
- The power module provides energy to all of the modules of sensor node, and computes the residual energy for furthermore schedule of the network operation.

4.2 The Connector Module

In our new designed sensor node, we particularly add a new module, connector module. The connector module mainly affects the extensibility of sensor node. The special requirements particularly for the sensors can be put in the expansion board. The sensor module can be re-designed to meet different demands and connect to the main board via the connector module. This kind of design significantly improves the extensibility of sensor networks.

In the connector module, we use a DF9-31 plug-in connection from HRS company. It has 9.9v, 3.3v interferences, SPI, ADC, serial port, and etc. With these connection ports, the sensor node can support many kinds of sensors, radio chips, memorizers, and other devices. The main connection circuit is shown in Fig. 12.

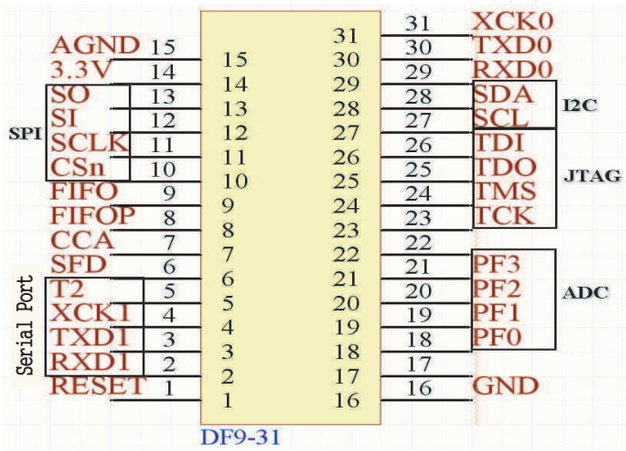


Fig. 12. Interface module schematic of the connector module

It mainly includes the following connection ports.

- SPI interface: It connects to the radio chip.
- Serial Port: It can connect to other terminal for the debugging and communication.
- ADC interface: It connects to some analog sensors for the A/D conversions.
- JTAG interface: It can be used for the software debugging and controlling.
- I2C interface: It connects to some digital sensors and memorizers.
- I/O interface: It also connects to some special digital sensors and memorizers through simulating time sequences.
- 3.3v and 9.9v power interface.

As shown in the figure, "SO, SI, SCLK, CSn" are used for the SPI interface, "TDI, TDO, TMS, TCK" are used for the JTAG interface, "ADC3, ADC2, ADC1, ADC0" are used for the ADC interface, "RXD1, TXD1, XCK1, T2" are used for the serial port, "SCL, ADA" are used for the I2C bus interface, respectively. "GND" and "AGND" are the ground pins. "9v" and "3.3v" provide the power for other components in the node. "RESET" is used for the reset of the microcontroller ATmega128. "FIFO, FIFOP, CCA, SFD" can be used for the interruption, which connects to the wireless communication module. "XCK0, TXD0, RXD0" are used for the serial port interface. By the way, all of the above ports can be used for I/O interface. Due to the space limitation, Fig. 13 only shows the serial port schematic as an example.

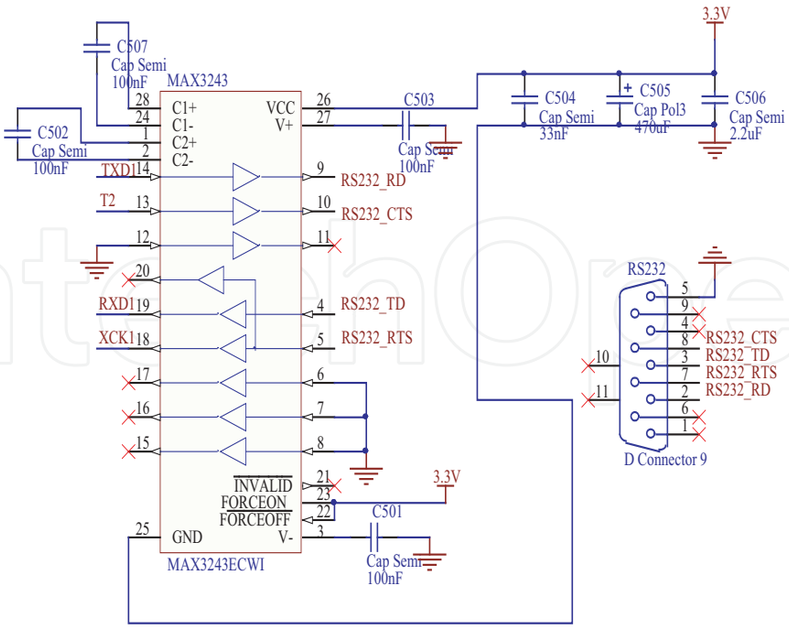


Fig. 13. Serial port schematic

With the above connector module, it improves the extensibility of sensor node. The sensor node can be configured with some common sensors such as temperature/humidity sensor, luminance sensor and accelerometer, support other analog and digital sensors with I2C bus interface and other ports, and connect to the memories that can keep the sensory data and node information. With the redesigned expansion board, the sensor node can easily implement many new functions with new added sensors.

4.3 The Implementation of Sensor node

Our developed sensor node is composed of two parts: main board and expansion board. These two boards are connected through the connector module. The microcontroller module, wireless communication module and power module locates in the main board. The sensor module and some JTAG debug interface locates in the expansion board. The main board is the core of sensor node, it can be connected to some other expansion boards developed according to new requirements. The main board is shown in Fig. 14. In the figure, we indicate the electromagnetic shield cover and the connector module. The electromagnetic shield cover can reduce the electromagnetic interference. The connector module can connect to the expansion board to extend the functions of sensor node.

In Fig. 15, we show out the complete sensor node. The expansion board is put on the right of main board. It has temperature/humidity sensor SHT11, luminance sensor TSL2561, and accelerometer. Through the second development to the expansion board, it is easy to implement the functions to other types of sensors. For example, if we want to add smog sensor into the node, we only need to redesign the expansion board based on the interface functions provided by the connector module, and does not need modify the main board.

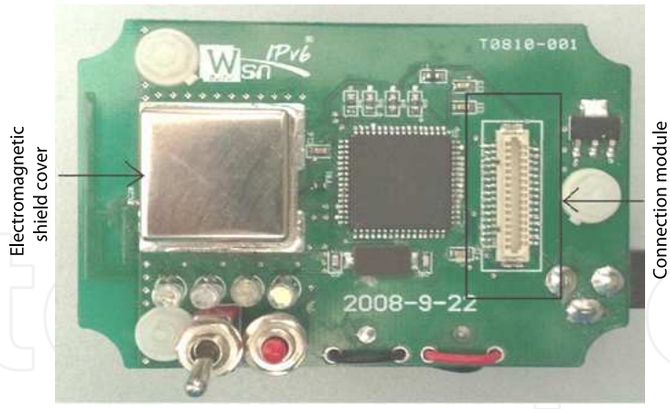


Fig. 14. Main board of the sensor node

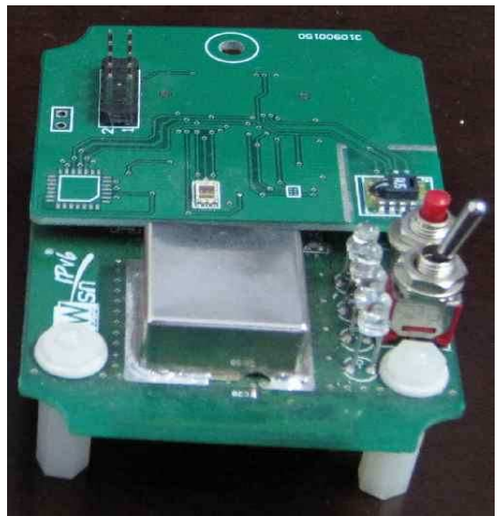


Fig. 15. The picture of sensor node

4.4 The Secondary Development Procedure of the Sensor Node

In our designed general purpose sensor node, it has been configured with temperature/humdity sensor, luminance sensor and accelerometer, which can satisfy the demands of many applications. However, if some applications have special requirements, we need redesign the expansion board according to the characteristics of hardware component and the layout of the connector module. In the designed connector module, it mainly supports SPI, I2C, serial port, I/O ports, with which most of sensors and radio chips can be supported. In the following, we briefly introduce the redesign procedure.

- Choose the hardware components according to the application requirements.
- Determine the interface according to the chosen components and layout of connector module.
- Design the schematic diagram according to the chosen components and interface.
- Design the PCB diagram according to the schematic diagram and the expansion board's requirements.
- Debug the new design modules.

For example, in order to adjust room temperature, we need to add the infrared component into the sensor node to control air conditioner. Firstly, we need to select the type of infrared transceiver according to the power and communication distance. After that, in order to keep the temperature/humidity sensors, we only can choose the free interface from the connector module, which can support connection to the infrared transceiver. In our designed node, we can choose the I/O port for this purpose. Then, we design the schematic diagram and PCB diagram accordingly. Finally, we debug the new design component.

5. Software Architecture of Sensor Node

5.1 The Software Framework of Sensor Node

The software framework of sensor node, which is shown in Fig. 16, is mainly composed of microcontroller software module, wireless communication software module, connector software module, sensor software module. Among them, microcontroller software module is the core of sensor node's software, which includes main process, communication protocol, and some application programs. It is responsible for all software modules' controlling and scheduling. The wireless communication software module is responsible for information exchanging and data delivery between sensor nodes. The connector software module is the key one to help implement the universality and reconfigurability. By the way, power module does not need software to control it so that we do not include it into the software framework. Sensor software module can configure with kinds of sensors to collect information according to application demands. In order to coordinate procedures of different sensors, we also design an adaptive data processing mechanism to work with multiple sensors.

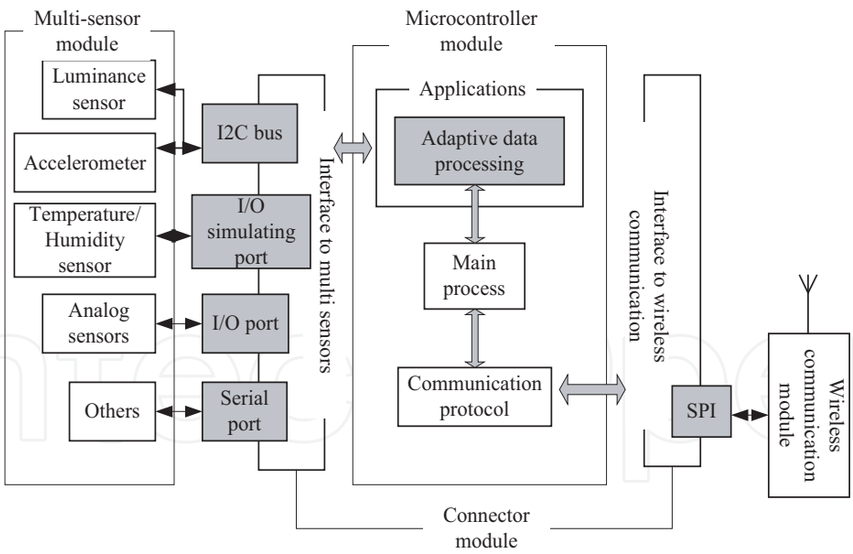


Fig. 16. Software framework of the sensor node

In the following, we firstly introduce some interfaces' operation mechanisms in the connector module. The details of other modules can be referred in Huo et al. (2006). Then, we explain the operation mechanisms of the adaptive data processing.

5.2 Interfaces in the Connector Module

We mainly introduce I2C interface, I/O simulating port, ADC interface, serial port and SPI interface in the connector module.

5.2.1 I2C Interface

In our designed node, the luminance sensor TSL2561 and accelerometer LIS3LV02DQ support I2C bus communications. Following the communication process of I2C bus, the sensor node can complete their sensory data collecting.

We give out the work flow procedure of the sensor TSL2561 as an example. After the sensor TSL2561 starts up, it initializes the I2C bus interface. To prevent the interference from LED lights, it needs to save the LEDs' work status and turn off the LEDs. Then, it sets the control register, time register, interruption register and threshold register of the TSL2561. And it measures the TSL2561 light intensities, compute the actual values of the light intensities. After that, it needs to restore the work status of LEDs. Please note that the control register setting and the light intensity reading is based on I2C time sequence in the communication one by one.

5.2.2 I/O Simulating Port

In our design sensor node, some sensors connect to the main board with same connection pins, but with with different interface technology. For example, the temperature/humidity sensor uses I/O communication technology, but the luminance sensor and accelerometer use I2C bus communication technology and they use the same pins of the connector module. Therefore, after the sensor node uses I2C bus technology to collect some sensory data, it needs to release I2C bus and restore the corresponding pins to I/O port for other type sensory data delivery. As above description, we need to simulate I/O port for the temperature/humidity information collecting. It needs to initialize the I/O port, sets them to three status "high impedance", "input" and "output", and uses the delay technology to simulate the communication time sequences of the sensors. In our sensor node, the data of temperature/humidity sensor SHT11 is communicated with the I/O simulating port. After it is powered, the data port and clock port are set to "output" status. The microcontoller sends the "start working" signal to the sensor SHT11. Then the microcontroller sends a "start temperature measuring" command to the sensor, and after some delay it starts reading "temperature data". Similarly, the sensor node can implement the humidity information collecting.

5.2.3 ADC Interface

For some analog sensors, the analog signals need to change to digital signals. In our sensor node, the input analog voltage in the ADC interface is changed to a 10-bit digital value. Its voltage reference (VREF) determines the ADC conversion range. The minimal value of VREF represents "GND", and the maximal one represents the value that the pin voltage subtracts 1LSB.

5.2.4 Serial Port

The serial port includes time clock generator, transmitter, receiver, and has three lines including "XCK1, TXD1" and "RXD1". "XCK1" is used for the synchronous transmitting mode. "TXD1" and "RXD1" are used for data sending and receiving respectively. After turning on the global interruption, it enables the serial port, enables the data receiving and saves the received data to the data register. Then, it collects the sensory data, writes them to the sending

register, enables the transmission function and sends out them. Finally, close the serial port and global interruption.

5.2.5 SPI interface

The microcontroller ATmega128L connects to radio communication CC2420 with SPI. In order to make sure the correct communication, master and slave devices have to operate in the same mode. In our sensor node, the microcontroller ATmega128L is the master device and the radio transceiver chip CC2420 is the slave device. Because the SPI communication time sequences of CC2420 has been fixed, we need to set the related register of ATmega128 accordingly. The SPI communication between ATmegal128L and CC2420 mainly involves writing and reading the related registers including control and data registers. Among them, it is most important to complete the writing operation to the sending register TxFIFO and the reading operation to the receiving register RxFIFO.

5.3 Adaptive Data Processing

The microcontroller software module is the core of sensor node, which is responsible for controlling, coordinating other modules. It controls the communication with other nodes, and deals with the sensory data locally. The on-site processing of sensory data, which mainly includes two sub-functions, makes adjustment to the data sensing according to environmental situations. One is that it deals with the sensory data considering the variations of measuring values. The second one is that it is able to deal with different sensory data with different priorities specially under some emergency situations.

5.3.1 Detection of Sensory Data Changing

Because normally the luminance and temperature' variation are smooth, it is unnecessary to collect these information frequently and we set the threshold values to decide the information collecting. It can also help save the hardware resource and the sensor node's energy because data delivery through wireless communication module consumes more energy than other modules in the sensor node. With the temperature/humidity sensor as an example, after the sensor nodes starts up, it will collect the sensory data in every one second. Each time, it will compares the current collected data with the last one, if the change is larger than the predefined threshold, it will transmit to the server and save the current one in the sensor node for the next comparison. However, if the sensor node does not send the sensory data up to two minutes, then it immediately sends and saves it without considering the sensory data's change.

5.3.2 Priority Setting for Different Sensors

In our designed node, there are multiple sensors that may collect the environment information at the same time. In this situation, the sensor with the highest priority will get the opportunity to delivery the sensory data to the microcontroller through the connector module. And other sensors will be delayed for some time. However, if one sensor detects an emergency incident, it will immediately get the resource to complete the data delivery in spite of its priority. For example, in our designed node, the accelerometer has the highest priority. But if the temperature/humdiity sensor detects that the temperature is changed significantly, then it will be set to the highest priority because this temperature significant change may indicate that there occurs emergency incidents. After the data is delivered, the priority can be reset to the initial ones.

6. Experiment Results and Analysis

In this chapter, we set up a test-bed to conduct the experiments with our developed sensor nodes. In the following, we firstly illustrate the multiple paths building procedure. Then, we set up a network scenario with an unavailable area simulating the fire disaster and give up the multiple paths building procedure under this situation.

6.1 Multi-Path Building Procedure

The position of sensor nodes is shown in Fig. 17. The communication range of each sensor node is illustrated in the circle with different color. From the figure, the sensor node "1945" can communicate directly with the sink node, the sensor nodes "1946" and "1949". The sensor node "1946" can communicate directly with the sink node, the sensor nodes "1945" and "1949". And the sensor node "1949" can communicate directly with the sensor nodes "1945" and "1946". In the following, we give out the detailed multi-route building procedure of the sensor node "1946" as an example.

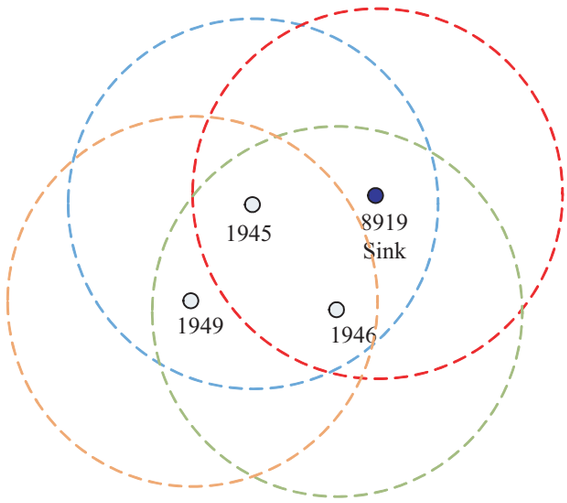


Fig. 17. Experiment scenario of multi-route building procedure

Before the sensor node "1946" wants to send data to the sink node, it needs to build the route to the sink node. Firstly, it broadcasts a RREQ message.

Time (us)	Length	Frame control field						Sequence number	Dest. PAN	Dest. Address	Source Address
		Type	Sec	Pnd	Ack req	Intra	PAN				
+3200											
=35240265	41	DATA	0	0	0	1		0x32	0x2420	0xFFFF	0x0000004792631946

MAC payload	LQI	FCS
0C 00 00 20 24 00 19 89 95 22 47 00 00 00 46 19 63 92 47 00 00 00 FF 01	52	OK

Fig. 18. Flow chart of intermediate nodes receiving RREP

Because the sink node is located within the communication range of the sensor node "1946", it receives the RREQ message and replies a RREP message to it. Fig. 19 shows our the RREP message sent from the sink node to the sensor node "'1946'. After the sensor node receives the RREP message. It will send back an ACK frame to the sink node and the direct communication path between them is built up.

Time (us)	Length	Frame control field							Sequence number	Dest. PAN	Dest. Address
		Type	Sec	Pnd	Ack req		Intra PAN				
+303403	47	DATA	0	0	1		1		0x1B	0x2420	0x0000004792631946
=35543668											

Source Address	MAC payload	LQI	FCS
0x0000004722958919	0C 00 01 00 20 24 00 19 89 95 22 47	160	OK
	00 00 00 46 19 63 92 47 00 00 00 FF		

Fig. 19. The RREP message sent from the sink node to the sensor node “1946”

Similarly, because the sensor node “1945” is also located in the communication range of the sensor node “1946”, it can receive the RREQ message. In this situation, the sensor node “1945” will forward the received RREQ to the sink node. The forwarded RREQ message is shown in Fig. 20. As shown in the figure, the fields “Type” and “Source address” are still “00” and “1946”, but the field “Number of hops” has been changed to “01”.

Time (us)	Length	Frame control field							Sequence number	Dest. PAN	Dest. Address	Source Address
		Type	Sec	Pnd	Ack req		Intra PAN					
+80116	41	DATA	0	0	0		1		0x9F	0x2420	0xFFFF	0x0000004792631945
=35625677												

MAC payload	LQI	FCS
0C 00 00 20 24 01 19 89 95 22 47 00	92	OK
00 00 46 19 63 92 47 00 00 00 04 01		

Type Number of hops Source address

Fig. 20. The RREQ message forwarded by the sensor node “1945”

After the sink node receives the RREQ message forwarded by the sensor node “1945”, it will reply to the sensor node “1945” a RREP message, which is shown in Fig. 21. In the figure, the fields “Type”, “Number of hops”, “Source address” are “01”, “00” and “1946” respectively.

Time (us)	Length	Frame control field							Sequence number	Dest. PAN	Dest. Address
		Type	Sec	Pnd	Ack req		Intra PAN				
+275426	47	DATA	0	0	1		1		0x1C	0x2420	0x0000004792631945
=35901103											

Source Address	MAC payload	LQI	FCS
0x0000004722958919	0C 00 01 00 20 24 00 19 89 95 22 47	160	OK
	00 00 00 46 19 63 92 47 00 00 00 FF		

Type Number of hops Source address

Fig. 21. The RREP message sent from the sink node to the sensor node “1945”

After the sensor node “1945” receives the RREP message, it will send back an ACK to the sink node and check the field “Source address”. It finds that it is not the RREP message’s final destination and will search the route to the sensor node “1946” in its routing table, which is built up during the RREQ forwarding procedure. After it finds out the corresponding route entry, it will continue forwarding out the RREP message, which is shown in Fig. 22. In the figure, the field “Number of hops” has become “01”, and the field “Source address” is still “1946”. Also, please note that the field “Address of forwarding node” is “1945”.
After the sensor node “1945” receives the forwarded RREP message, it will send back an ACK to the sensor node “1945” and finally build up the second route via the sensor node “1945” between the sensor node “1946” and the sink node. In fig. 23, we give out the final result of building up multiples routes for the sensor nodes “1946” and “1949”. The sensor node “1945” firstly starts up and builds a direct communication route to the sink node, which is not shown

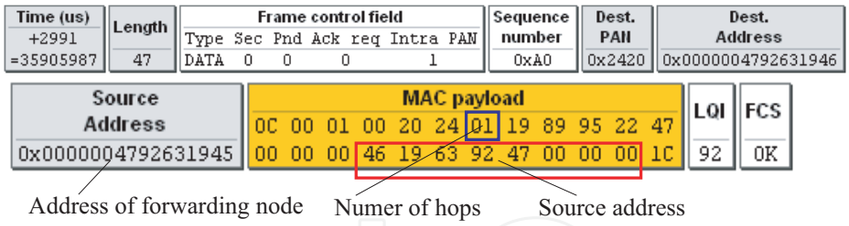


Fig. 22. Flow chart for stage of routing reply

in the figure. Then, the sensor node "1946" starts up and builds up two routes to the sink node, which are shown with the blue solid and dashed lines. Finally, the sensor node "1949" starts up and also builds up two routes to the sink node, which are shown with the red solid and dashed lines.

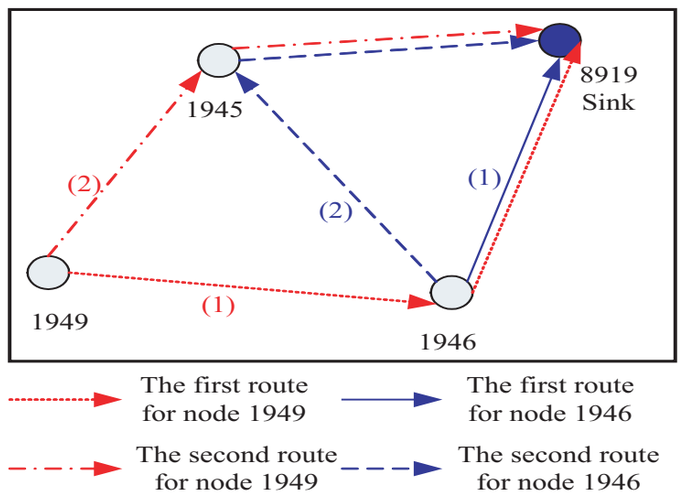


Fig. 23. The experiment result of building up multiple routes

6.2 Multi-Path Building Around the Unavailable Area

The experiment scenario of wireless sensor network is shown in Fig. 24. In the figure, the sensor node "8919" is the sink node. In order to simulate the fire disaster, we set the temperature threshold of some sensor nodes to a lower value. In our experiments, we change the temperature threshold of sensor nodes "1944, 1945, 1946" to 10°C, which is lower than current environmental temperature and the other sensor nodes' temperature thresholds are still 60°C. We start up the sensor nodes "8919, 1943, 1944, 1945, 1946, 1947" one bye one and among them the sensor nodes "1943"∼"1947" and the sink node can communicate with each other directly. After the sensor nodes "1944, 1945, 1946" start up and measure the environmental temperature, they will set them as unavailable sensor nodes because the measured temperature is larger than the predefine threshold value. Following that, they will notify their corresponding neighbor nodes "1943" and "1947". We set that the sensor nodes immediately send the data to the sink node after they start up, thus each sensor node will build their routes one bye one.

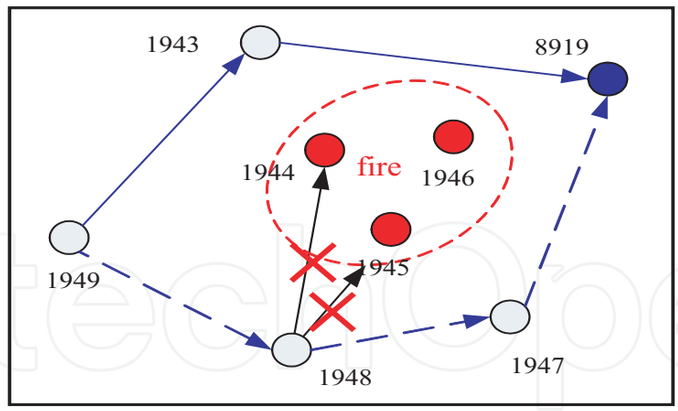


Fig. 24. Experiment scenario of wireless sensor network

The routing table and neighbor node table of sensor nodes “1943” and “1947” are shown in Tabs. 7, 8, 9 and 10. When the sensor node “1943” starts up, there is only the sink node that is working so that it only builds up one route in its routing table. And when the sensor node “1947” starts up, the sensor nodes “1943, 1944, 1945, 1946” and the sink node are working. But among them, the sensor nodes “1944, 1945, 1946” have been set to unavailable nodes so that they can not help forward the data from the sensor node “1947”. Thus, the sensor node “1947” only can build up the second route along the sensor nodes “1947⟶1943⟶8919” besides the direct communication route between it and the sink node.

Node’s ID	Available?
1944	No
1945	No
1946	No
1947	Yes

Table 7. Neighbor nodes table of the sensor node “1943”

Node’s ID	Available?
1943	Yes
1944	No
1945	No
1946	No

Table 8. Neighbor nodes table of the sensor node “1947”

In the following, we change the transmission power of the sensor nodes “1948” and “1949”. The sensor node “1948” can only communicate with the sensor nodes “1944, 1945, 1947”, and the sensor node “1949” can only communicate with the sensor nodes “1943, 1944, 1945, 1948”. And we start up the sensor nodes “1948” and “1949” one by one, their routing tables are shown in Tabs. 11 and 12. From their routing tables, the sensor nodes “1948” and “1949” successfully build the routes avoiding the unavailable areas.

Destination node's ID	Successor node's ID	Number of hops
8919	8919	1

Table 9. Routing table of the sensor node "1943"

Destination node's ID	Successor node's ID	Number of hops
8919	8919	1
8919	1943	2

Table 10. Routing table of the sensor node "1947"

Destination node's ID	Successor nodes's ID	Number of hops
8919	1947	2

Table 11. Routing table of the sensor node "1948"

Destination node's ID	Successor node's ID	Number of hops
8919	1943	2
8919	1948	3

Table 12. Routing table of the sensor node "1949"

7. Conclusions

In this chapter, we designed a new multi-path routing protocol, MSMRP, to cross around the unavailable areas based on our previously proposed MSRP routing protocol. In particularly, we design a neighbor node table exchanging mechanism that can help build an alternate route around the unavailable areas and try to avoid the multiple paths intersect. When a RREQ is arriving at some unavailable sensor nodes, they will not forward it so that these sensor nodes will not be included into the inverse routes from the sink to the source node. When a sensor node becomes unavailable during the RREP forwarding procedure, its precursor node will try to find the alternate route to forward the RREP to the destination. Finally, we implement the proposed protocol in the real sensor nodes and set up a testbed to conduct detail experiments. The experimental results show that MSMRP can perform well as we expect.

8. Acknowledgments

The authors gratefully acknowledge the support by "the Fundamental Research Funds for the Central Universities" under grant No. 2009JBM007, the support of the project-sponsored by SRF for ROCS, SEM under grant No. [2008]890, and the support of the National Natural Science Foundation of China (NSFC) under Grant No. 60802016, 60972010 and 60833002.

9. References

Chen, Y. & Nasser, N. (2008). Enabling QoS multipath routing protocol for wireless sensor networks, *Proc. of IEEE International Conference on Communications (ICC'08)*, Beijing, China, pp. 2421–2425.

Deering, S. & Hinden, R. (1998). Internet protocol, version 6 (IPv6), specification, RFC 2460.

Fang, Q., Gao, J. & Guibas, L. J. (2006). Locating and bypassing holes in sensor networks, *Springer Mobile Networks and Applications* **11**(2): 187–200.

Ganesan, D., Govindan, R., Shenker, S. & Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks, *ACM SIGMOBILE Mobile Computing and Communications Review* **5**(4): 11–25.

Gao, D., Niu, Y. & Zhang, H. (2009). Micro sensor routing protocol in IPv6 wireless sensor network, *Proc. of IEEE International Conference on Networking, Sensing and Control (ICNSC'09)*, Okayama, Japan, pp. 55–59.

Huang, X. & Fang, Y. (2008). Multiconstrained QoS multipath routing in wireless sensor networks, *Wireless Network* **14**(4): 465–478.

Huo, H., Zhang, H., Niu, Y., Gao, S., Li, Z. & Zhang, S. (2006). MSRLab6: An IPv6 wireless sensor networks testbed, *Proc. of the 8th International Conference on Signal Processing*, Vol. 4, Guilin, China.

Jafarian, M. & Jaseemuddin, M. (2008). Routing of emergency data in a wireless sensor network for mines, *Proc. of IEEE International Conference on Communications (ICC'08)*, Beijing, China, pp. 2813–2818.

Jennifer Yick, Biswanath Mukherjee, D. G. (2008). Wireless sensor network survey, *Computer Networks* **52**: 2292–2330.

Khan, I. & Javed, M. (2008). A survey on routing protocols and challenge of holes in wireless sensor networks, *Proc. of International Conference on Advanced Computer Theory and Engineering (ICACTE'08)*, Pukhet, Thailand, pp. 161–165.

KIM, M., JEONG, E., BANG, Y.-C., HWANG, S., SHIN, C., JIN, G.-J. & KIM, B. (2008). An energy-aware multipath routing algorithm in wireless sensor networks, *IEICE Transactions on Information and Systems* **E91-D**(10): 2419–2427.

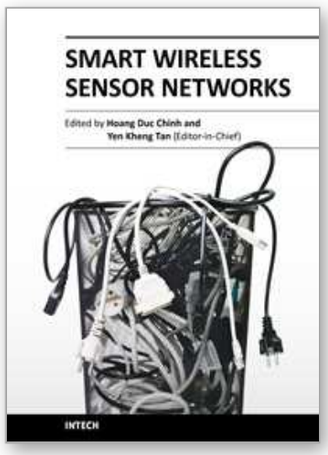
Lee, S.-J. & Gerla, M. (2001). Split multipath routing with maximally disjoint paths in ad hoc networks, *Proc. of IEEE International Conference on Communications (ICC'01)*, Helsinki, Finland, pp. 3201–3205.

Marina, M. & Das, S. (2001). On-demand multipath distance vector routing in ad hoc networks, *Proc. of the 9th International Conference on Network Protocols (ICNP'01)*, California, USA, pp. 14–23.

Nasser, N. & Chen, Y. (2007). Secure multipath routing protocol for wireless sensor networks, *Proc. of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, pp. 12–12.

Teo, J.-Y., Ha, Y. & Tham, C.-K. (2008). Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming, *IEEE Transactions on Mobile Computing* **7**(9): 1124–1137.

Toledo, A. & Wang, X. (2006). Efficient multipath in sensor networks using diffusion and network coding, *Proc. of the 40th Annual Conference on Information Sciences and Systems (CISS'06)*, Princeton, NJ, pp. 87–92.



Smart Wireless Sensor Networks

Edited by Yen Kheng Tan

ISBN 978-953-307-261-6
Hard cover, 418 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area â€“ wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodesâ€™ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of networkâ€™s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Deyun Gao (2010). Routing Protocol with Unavailable Nodes in Wireless Sensor Networks, Smart Wireless Sensor Networks, Yen Kheng Tan (Ed.), ISBN: 978-953-307-261-6, InTech, Available from: http://www.intechopen.com/books/smart-wireless-sensor-networks/routing-protocol-with-unavailable-nodes-in-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821

IntechOpen