Journal of Information Hiding and Multimedia Signal Processing ©2013 ISSN 2073-4212
Ubiquitous International Volume **4**, Number **1**, January **2013**

A Fault Management Protocol for Low-Energy and Efficient Wireless Sensor Networks

Tian-hua Liu

College of Software
Shenyang Normal University
253 Huanghe North Street, Shenyang City, Liaoning Province, China
liutianhua@sina.com

Si-chao Yi

College of Software
Shenyang Normal University
253 Huanghe North Street, Shenyang City, Liaoning Province, China
stevenysc@sina.com

Xiao-wei Wang

College of Software
Shenyang Normal University
253 Huanghe North Street, Shenyang City, Liaoning Province, China
wangxwvv@gmail.com

Received September 2012; revised November 2012

Abstract. *We propose a solution to fault management for Wireless sensor networks because of their own limitations and the scalability issue. By introducing new network equipments, one can improve the traditional distributed hierarchical management structure, the equipment can quickly locate the failure and analyze the cause of the failure, therefore can greatly improve the efficiency of network maintenance; we also propose a new low-energy fault management protocol, which can quickly respond to failures. The experimental results show that, compared with traditional protocols, this protocol can detect failures, responds quickly to failures and recover from failures at minimal costs, therefore reduce the impact of failures on networks.*
Keywords: Watermarking, Singular Value Decomposition, hacker attacks

1. **Introduction.** Wireless sensor networks (WSN) Sensor nodes are severely constrained in terms of storage resources, computational capabilities, communication bandwidth and power supply[1]. Relying on resource-constrained embedded devices for communication, processing, and sensing, WSNs can experience unexpected problems during deployment, due to hard-ware, software, or environmental anomalies[2]. So one of the research topics in wireless sensor networks is: how can people can manage WSN so that they run stably in the long term by making efficient use of energy.

Fault management protocols for WSN are primarily used for three architectures: centralized architecture, distributed architecture, hierarchical architecture [3]. MANNA, Sympathy, BOSS, MOTE-VIEW are proposed in [4-7]. The advantages of the centralized architecture are that it has a simple network structure, and has a higher accuracy in terms of fault management; drawback is that the management node has high resource

consumption. It costs energy to upload failure information, which may cause the "energy hole" phenomenon, costing more for the inner nodes close to the management node. In distributed architecture, there are multiple network management nodes, which divide the network into several regions. Each node is responsible for collecting and processing regional information, reduce the amount of information sent to the central node. Various distributed management architectures are proposed in [8-13]: LDACO,DSN, RM, NODE-energy level management, App-Sleep, sensor management optimization and WSN Diag. The advantage of distributed architecture is that failure is only reported to the core node when necessary, reducing the resource consumption of the core node as well as the energy consumption caused by frequent upload of the remaining nodes. The drawback is that the event management nodes have to have some processing capabilities which lead to more energy consumption by the management nodes. Hierarchical architecture uses middle-level management nodes, each management node is responsible for some regional nodes, and upload the information to higher-level nodes, each management region is independent. Some hierarchical management architectures are proposed in [14-17]: SNMP, STREAM, AppSleep and WSNMP. Hierarchical management architectures are widely used in recent years due to reliability and scalability issues.

This paper combines the distributed architecture and hierarchical architecture, introduces a new management device, which can locate the failure node and analyze the failure. Based on this new architecture, we propose LPS-FMP(Low Power the Speed-the Fault Management Protocol), a fault management protocol which can respond quickly to abnormal failures.

2. **Distributed hierarchical fault management architecture for WSN.**

Distributed hierarchical fault management architecture for WSN includes Management Station, Management Device, Agent Device, Gateway Device and Normal Senor Node.

2.1. **Management Station.**

The management station is responsible for the Internet connection, running fault management program and network maintenance program, providing a friendly user interface and stable energy for the management devices to ensure that the management devices run stably in the long term.

2.2. **Management Device.**

The management device is the core and the initiator of the entire network and is responsible for planning and controlling over the network. Its mainly responsible for collecting information from the network management entity and the ordinary nodes, send and it analyzes, process and schedule important events, it is the most important manager of the entire network.

2.3. **Agent Device.**

Agent device is mainly responsible for the communication between the ordinary sensor nodes and the management device, information and is central in the entire network. Agent devices make a decision on events uploaded from nodes below, and do failure detection, failure diagnosis and failure processing periodically, it is capable of recovering from failures and is a FFD(Full-Function Device). Agent devices are implemented as control centers and cluster head node in WSN management architecture.

2.4. **Gateway Device.**

Gateway devices are used to connect WSN and the Internet. Its mainly responsible for locating failure devices, reading failure information from failure nodes, analyzing failures and upload failure time, location, type and reasons to the management station.

2.5. **Ordinary Senor Node.** Ordinary sensor nodes are at the lowest level of the hierarchy. They are responsible for uploading to the cluster head node perceived information from the external environment, is a RFD(Reduced-function Device) node.

The architecture of Distributed hierarchical fault management system is shown in figure 1.

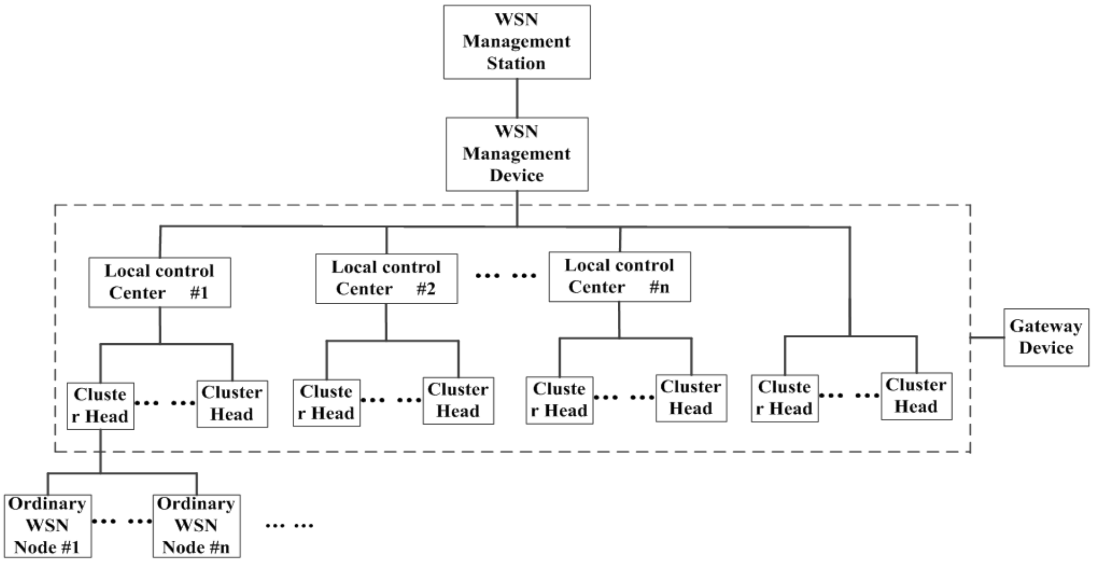

FIGURE 1. Distributed Hierarchical Fault Management Architecture based on WSN

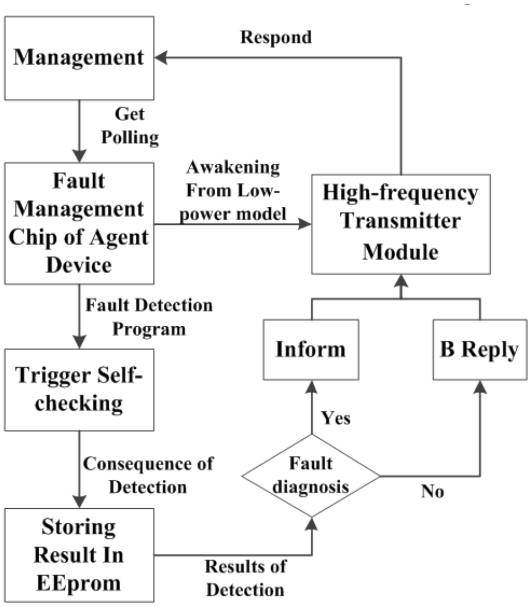

FIGURE 2. Management queries Agent

2.6. **Fault Management Process.** All types of nodes in the distributed hierarchical fault management architecture can be used as a fault management entity. In other words, the full-function device (FFD) and Reduced Function Device (RFD) all have features

such as fault detection, diagnosis and processing. The status and roles of each device in distributed hierarchical fault management architecture are different, but they differ only in the capabilities and implementation details of failure handling, the processes are essentially the same.

Agent devices in the fault management architecture are implemented as the control center and the Cluster head node. Each Agent device has two ways of making a query: Management devices regularly ask for information from agent device; Agent devices periodic do self-test and upload failure after detection. The workflow is shown in figure 2 and figure 3.

3. **Fault Management Protocol for WSN.**

3.1. **Distributed hierarchical fault management protocol for WSN(LPS-FMP).**

The LPS-FMP protocol is mainly designed for two management entities, Management and the Agent. To implement of the fault management service for WSN, one must ensure that management application link is established among the nodes in the network, this is different from the one defined in the protocol of the underlying WSN, so we call it fault management service link. In order to implement fault management service link, we define five communication primitives:

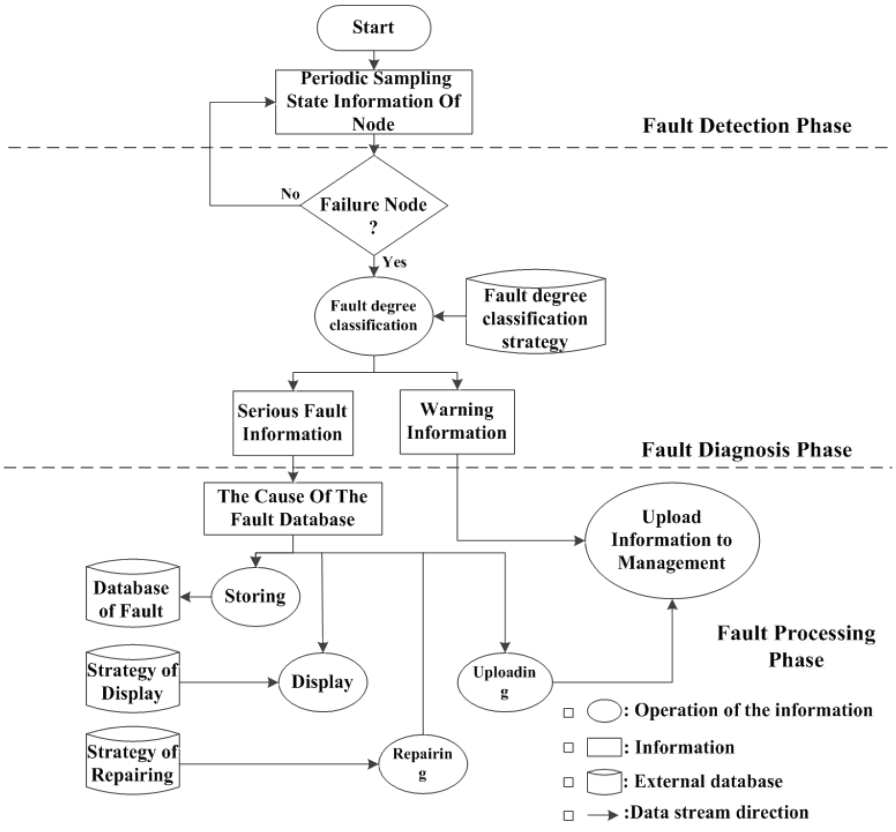

FIGURE 3. Agent uploads fault information

(1) Connection Request: used for communications and negotiations prior to the fault management operations
(2) Connection Reply: acknowledgement that agent sends back to management in response to link establishment

(3) Connection Reject: Agent rejects application from management to establish connections
(4) Disconnection Request: After fault management service is done, management sends out this primitive to disconnect the link.
(5) Disconnection Reply: Agent sends out this primitive in response to management to disconnect the link.

After the connection is established, we can use LPS-FMP to perform fault management service. We defines four kinds of operations: Get Request, Set Request, Inform, Cancel, and two kinds of response modes: B Reply, P Reply to complete the transmission of fault management information.

Get Request: Management sends this to Agent as a request for information, it operates on the properties of agents.

Set Request: Management uses this to modify properties of Agents, this includes resetting agents to their default values.

Inform: To pass around notice or event management among entities. It can be sent from Management to Agent, or vice versa. It is mainly used for the Agent to send fault information to the Management.

Cancel: To cancel the currently executing fault management tasks, and also to release the system resources being occupied.

B Reply: A simple response which does not contain too much information, it is simply a response to the management services operation.

P Reply: This is a detailed response which contains more information, including whether this service management is successful and response parameter values.

3.2. **LPS-FMP Fault Management Information Library.** This library contains information related to management entities involved and all the nodes, such as type information, parameter information, property information, services, operational information, module information and the type of failure. It has downward compatibility and can be extended to other wireless communications, such as RFID wireless communication.

Data in the library is organized as shown in figure 4. It gives a uniformed name and identifier to all the entities and services.

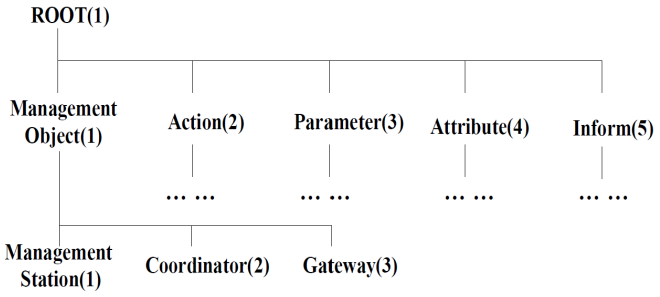

FIGURE 4. The registration tree in the fault management information library

Here are the steps of identification: Put all the numerical identification together, from the root node to the target object, putting a . between the numbers. For example, Management Station is on the third level in Figure 4, starting from the root, passing through Management Object, down to Management Station, so the identification is 1.1.1.

(1) the definition of managed objects in management
There are three types of managed objects: Management Station, Coordinator, Gateway, as shown in figure 5.

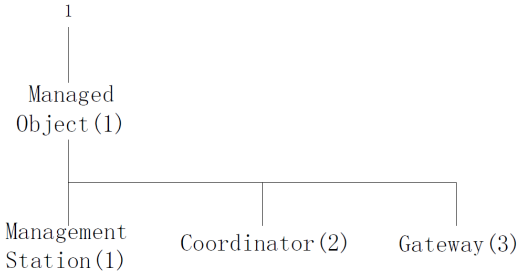

FIGURE 5. The registration tree of managed objects

(2) the definition of managed objects in management services
There are two types of objects in this category: Connect Serve and Operate Serve.

Connect Serve: It is used to establish fault management service link. Connect Serve defines five primitives: Connection Request, Connection Reply, Connection Reject, Disconnection Request and Disconnection Reply.

The Operate Serve: After fault management service link is established, this is used for fault management service operations. There are four types of operations: Get Request, Set Request, Inform, Cancel, and two modes of responses: B Reply and P Reply. Their job is to complete the transmission of fault management information. Their locations in the registration tree is shown in figure 6.

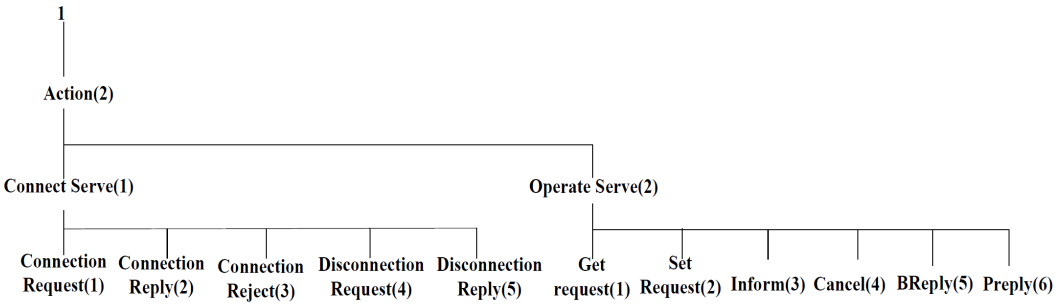

FIGURE 6. Registration tree of managed object for management service operations

(3) the definition of managed objects of node parameters
There are two types of objects in this category: Zigbee and RFID. Zigbee defines Hardware and Software, describes the hardware module information and software version information of the wireless sensor networks, mainly used for network maintenance and upload of fault information. RFID also defines Hardware and Software, mainly used for compatible RFID applications. The positions in the registration tree are shown in figure 7.

(4) the definition of managed objects of node properties
There are four types of objects in this category: Coordinator, Router Device, End Device. Describes the working mode and type of nodes in WSN.

Coordinator: it is the initiator of the network, it is also the aggregation node of the network.

Router: it is the relay node of the network, it is responsible for forwarding information.

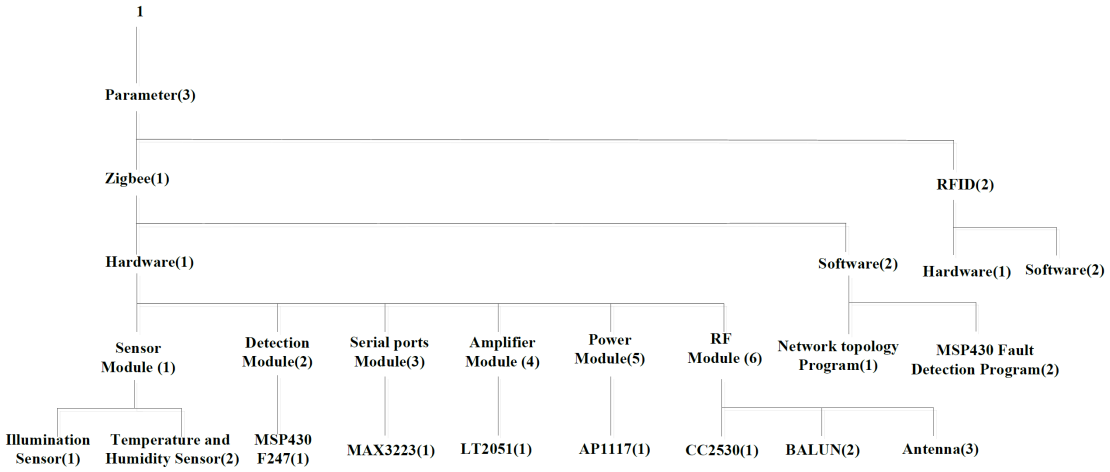

FIGURE 7. Registration tree of managed objects for node parameters

Device: it is useful for sensing and collecting environmental information, it belongs to FFD device.

End Device: it is on the lowest level of the hierarchy, is only used to collect environmental information, it has a unique function and it is not easily extensible, it is an RFD device, its location is shown in figure 8.

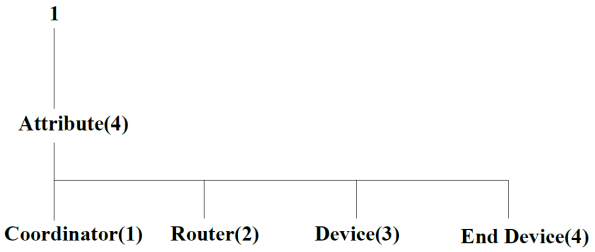

FIGURE 8. Registration tree of managed objects for node properties

(5) the definition of managed objects of notification
There are three types of objects in this category: Error Status, Enterprise, Community. Error Status: it defines the types of errors, it can be further divided into Node Fault and No Fault, namely node-failure and failure-free types. Battery Fault, Sensor Fault, Transmit Fault are defined in Node Fault.

Enterprise: defines the type of the data being sent. Enterprise is divided into two types: Send Data and Warning.

Community: it is used to ensure that information security of the network, it defines if data is a public or private. Its position in the registration tree is shown in figure 9.

3.3. **The entities in LPS-FMP fault management protocol.** The working mechanism of LPS-FMP fault management protocols is shown in figure 10. On one hand: Management sends operation requests to Agent regularly, the request can be routine operation requests to any node to collect perceived environmental information or operation requests to critical Agent periodically to check if Agent can respond to this request. In

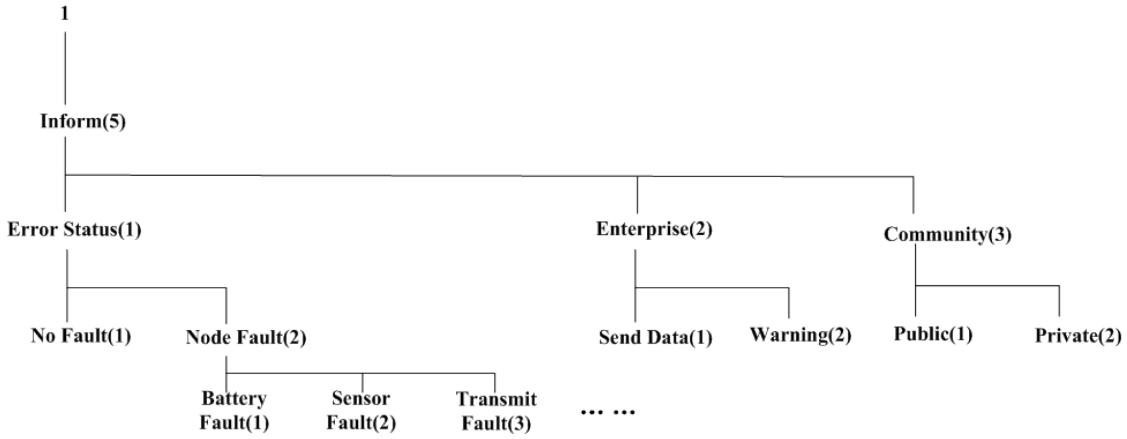

FIGURE 9. Registration tree of managed objects of notification

the case where Agent does not respond, Management resends the request. If the agent still does not respond, Management would consider that the Agent fails, Management will update its routing table and let Agent update their routing table by using Set Request primitive. If the Agent responds to either one of the two requests, Management would consider that the Agent is working normally.

On the other hand, Agent periodic self-detected failure and take the initiative to upload failure. Therefore, the Management can tell if Agent fails by taking advantage of Inform uploaded by Agent.

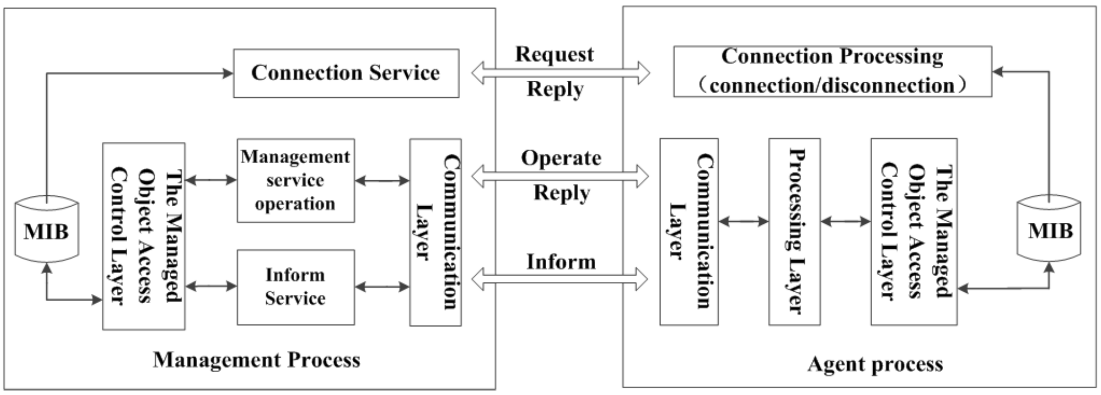

FIGURE 10. The working mechanism of entities in LPS-FMP fault management protocol

The data source of failure detection based on LPS-FMP: LPS-FMP Inform and the data obtained by querying Agent using Get.

(1) LPS-FMP Inform
LPS-FMP Inform is an asynchronous communication between Agent and Management. When the Agent node detect that some module exceeds a pre-defined security threshold, the Agent sends to Management LPS-FMP Inform containing failure information and failure address. Inform PDU structure is as follows: PDU Header refers to the node address of the entity sending out Inform; Time refers to the time

Inform PDU		
PDU Header	Time	Server ID
Node ID	Current Time	Error Status

of failure; Error Status refers to the prefix identification in LPS-FMP registration tree, represents the module that response to failure.

(2) LPS-FMP Get queries
LPS-FMP Get query happens when Management query agent to check if it is in normal state and the status of relevant modules and the perceived data, Agent respons to Management using Inform, whose structure is as follows: Object ID refers to

Get PDU	
PDU Header	Object ID

the prefix indentifier in WDHFMP registration tree, representing the target module of the node being queried. PDU Header consists of four parts: Server Connection

Server Connection TAG	Request ID	PDU Type	Agent ID

TAG is an identification that Management provides Agent, representing primitives establishing management application links; Request ID refers to ID of the requesting node, i.e. Management ID; PDU Type is used to identify the type of the operation; Agent ID is used to represent ID of the target node.

LPS-FMP Inform and LPS-FMP Get minimize the size of the frame so as to save energy.

4. **Design and Implementation.**

4.1. **Design and Implementation of System Model.**

Wireless sensor network fault detection device is fully compatible with wireless sensor nodes and perception of environmental information and network formation. In addition, it can detect important modules of the node, the structure is shown in figure 11. It consists mainly of the Agent fault detection devices, gateway fault detection devices and client software.

Agent fault detection devices consist of three functional components. The first is the network formation components, used to perceive environmental information, form the network and transmit information. The second is the failure detection component, which has the battery-powered module, the sensor module and the high frequency transmitter signal sampling module, used to determine whether the important modules in the network fail. The third is the failure-handling components, when failure arises, keep the node in working mode as long as possible and store failure information. First, Fault management collect information from power modules, sensor modules and high frequency transmitter module to decide whether failure exists. If so, the system enters the fault diagnosis stage, fault management chip will determine if it is just a warning or a real failure.

And then the system enters the failure-handling stage, if it is handling a warning, the system uses Inform in LPS-FMP to forward this information up in the hierarchy until it reaches the Management device. The way the system handles a real failure differs by modules, the commonality is that the fault management chip stores the failure information and the timing information into the failure information storage module. Failure related

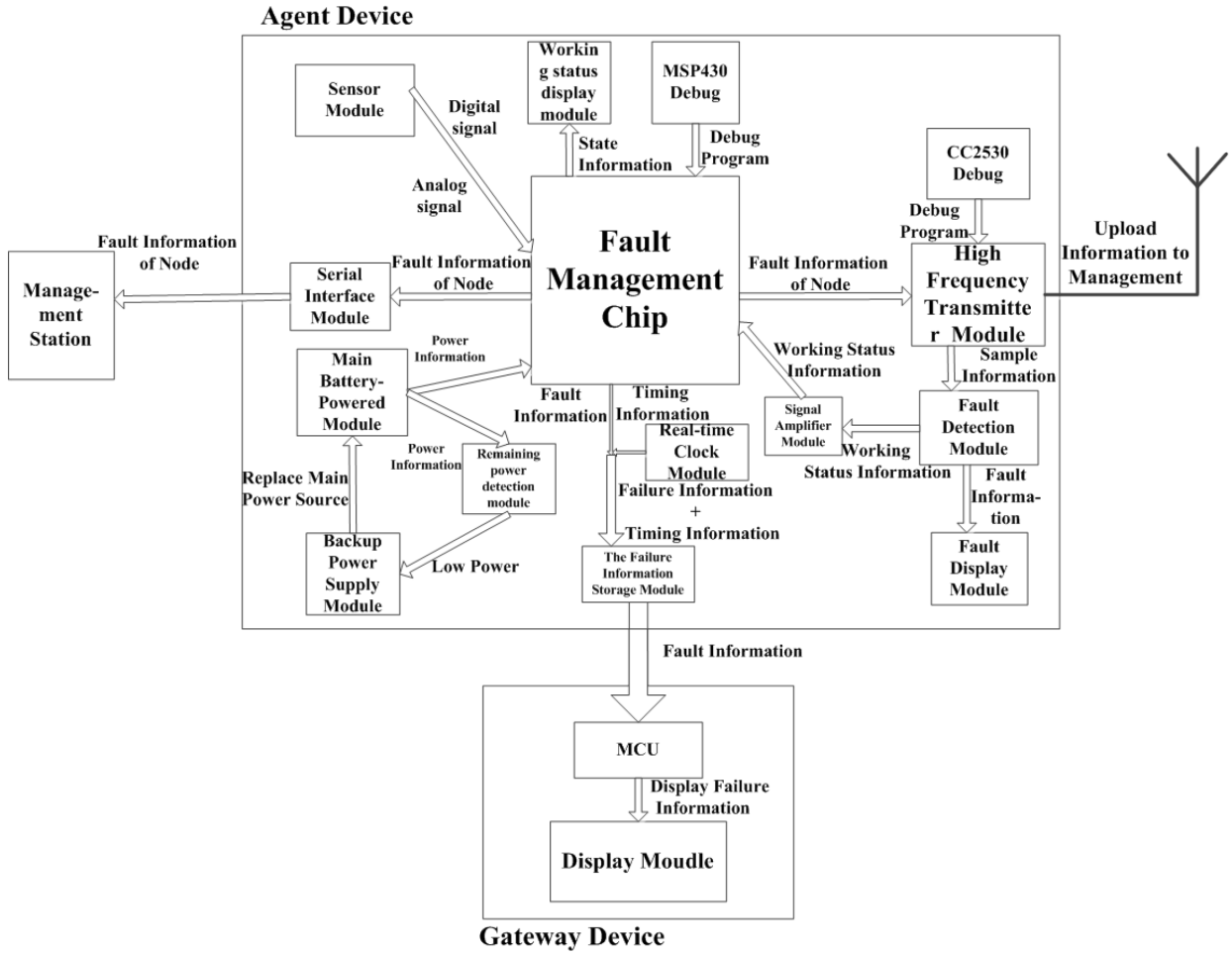

FIGURE 11. The diagram of system modules

to sensor module and power module is treated the same as the warning, it is forwarded up in the hierarchy until it reaches Management device and Management station, so as to alert the network administrator. If the failure is related to the power module, the fault management chip will start the backup power supply to replace the original power supply until the system recovers from this failure. The high frequency transmitter module failure will result in that a node cannot communicate with the outside world, therefore the only way to find this failure is that Management queries the device using Get in the LPS-FMP protocol, to update routing tables accordingly so that other nodes do not send data to this failure node.

Finally, the system enters the fault maintenance phase. Fault management software collect information in two ways: Management sends out queries, Agent uploads information. After failure occurs, maintenance module will use the gateway device to locate the failure, then analyze the failure according to fault information storage module, repair the failure as quickly as possible, improve the efficiency of network maintenance

5. **Performance Evaluations.** Performance evaluations were done using WSNMP with similar architectures. We consider fault response time and average energy consumption within a single cluster. We use Matlab, 100 wireless sensor nodes are deployed in a 200 x 200 region randomly, each node in the network launch radius is 25 meters, each node carries two joules of energy.

LPS-FMP protocol defines the frame format of Inform. When uploading failure information, we try to minimize the amount of data, under the constraint that the event is completely described; Agent detects if failure occurs through self-detecting the status of its own modules. Since each Agent is only responsible for its own modules, failure can be detected as quickly as possible, Agent uploads failure information to Management in a hierarchical manner within the fewest possible steps, effectively reduce the average energy consumption of network and minimizes the time between when the failure occurs and when Management gets informed. The ratio of fault response time over WSN nodes number is shown in figure 12. The ratio of average energy dissipation over agent nodes number is shown in figure 13.

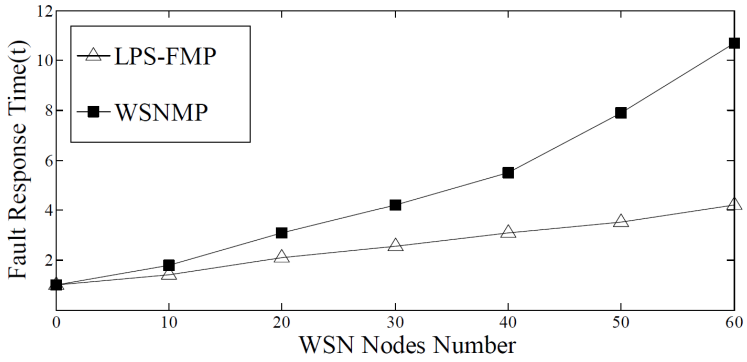

FIGURE 12. The relationship between Fault response time and WSN nodes number

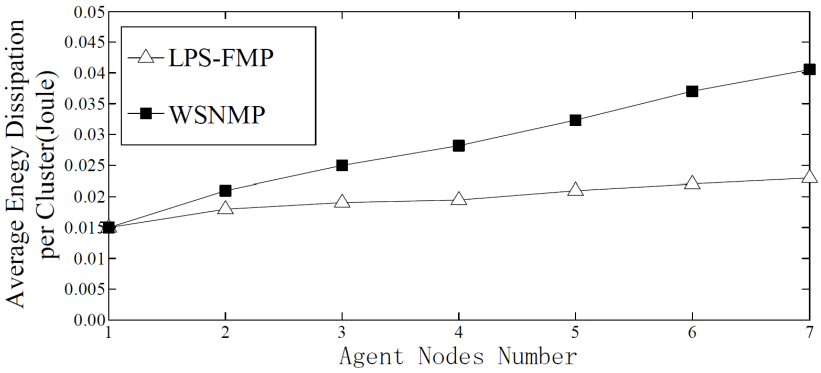

FIGURE 13. The relationship between average energy dissipation and agent nodes number

6. **Conclusion.** We propose a new fault management protocol LPS-FMP based on Distributed hierarchical wireless sensor network architecture. Using LPS-FMP, failure information can be obtained in two ways: Management sends out queries, or Agent uploads information using Inform. Compared with traditional WSN fault detection mechanism, we add gateway devices, which can locate and analyze failures. This greatly improves the efficiency of network maintenance and fault repairs. Finally, we implement a hardware platform and software for fault management protocol based on distributed hierarchical architecture. If some module fails in Agent, we can still take some actions to deal with this emergency. We extend the working time of the failure node as long as possible. Within

this amount of time, we try to repair the fault. We can avoid the cost of updating routing tables of the entire network caused by replacing the failure node.

Acknowledgment. Project supported by Liaoning Provincial Natural Science Foundation (Grant No. 201102201), Liaoning Baiqianwan Talents Program (No. 2011921046).

REFERENCES

[1] R. Jurdak, X. R. Wang, O. Obst, and P. Valencia, Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies, *Intelligence-Based Systems Engineering*, vol. 10, Springer, Berlin-Heidelberg, Germany, 2011.

[2] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, Computational Intelligence in Wireless Sensor Networks: A Survey, *Journal of IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68-96, 2011.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey, *Journal of Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[4] N. Ramanathan, E. Kohler, L. Girod, and D. Estrin, Sympathy: a debugging system for sensor networks, *Proc. of the 29th IEEE International Conference on Local Computer Networks*, pp. 554-555, 2004.

[5] W. L. Lee , A. Datta , and R. Cardell-oliver, WinMS: Wireless sensor network-management, system, an adaptive policy-based management for wireless sensor networks, *Proc. of CSSE Technical Report UWA-CSSE-06-001*, pp. 1-21, 2006.

[6] H. Song, D. Kim, K. Lee, and J. Sung, Upnp-based sensor network management architecture, *Proc. of the 2nd International Conference on Mobile Computing and Ubiquitous Networking*, pp. 21-26, 2005.

[7] M. Turon, Mote-view: a sensor network monitoring and management tool, *Proc. of the 2nd IEEE Workshop on Embedded Networked Sensors*, pp. 11-17, 2005.

[8] J. H. Hoa, H. C. Shihb, B. Y. Liaob, and S. C. Chuc, A ladder diffusion algorithm using ant colony optimization for wireless sensor networks, *Journal of Information Sciences*, vol. 192, pp. 204-212, 2012.

[9] E. C. Kulasekere, K. Premaratne, and P. H. Bauer, Resource management of task oriented distributed sensor networks, *Proc. of IEEE International Symposium on Circuits and Systems*, vol. 3, no. 2, pp. 513-516, 2001.

[10] A. Boulis, and M. Srivastava, Node-level energy management for sensor networks in the presence of multiple applications, *Journal of Wireless Networks - Special Issue: Pervasive Computing and Communications*, vol. 10, no. 6, pp. 737-746, 2004.

[11] N. Ramanathan, M. Yarvis, J. Chhabra, N. Kushalnagar, L. Krishnamurthy, and D. Estrin, A stream-oriented power management protocol for low duty cycle sensor network applications, *Proc. of the 2nd IEEE Workshop on Embedded Networked Sensors*, pp. 53-62, 2005.

[12] M. Perillo, and W. B. Heinzelman, Providing application QoS through intelligent sensor management, *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 93-101, 2003.

[13] A. K. Somani, and V. K. Agarwal, Distributed diagnosis algorithms for regular interconnected structures, *IEEE Trans. Computers*, Vol. 41, no. 7, pp. 899-906, 1992.

[14] B. Deb, S. Bhatnagar, and B. Nath, Stream: sensor topology retrieval at multiple resolutions, *Journal of Telecommunication Systems*, vol. 26, no. 2-4, pp. 285-320, 2004.

[15] S. Rajasegarar, C. Leckie, M. Palaniswami, and James C. Bezdek, Quarter sphere based distributed anomaly detection in wireless sensor networks, *Proc. of IEEE International Conference on Communications*, pp. 3864-3869 , 2007.

[16] M. Yu, H. Mokhtar, and M. Merabti, Self-managed fault management in wireless sensor networks, *Proc. of the 2nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 13-18, 2008.

[17] M. M. Alam, Md. Mamun-Or-Rashid, and C. S. Hong, Wsnmp: a network management protocol for wireless sensor networks, *Proc. of the 10th International Conference on Advanced Communication Technology*, pp. 742-747 ,2008.