

## An Adjustable Sleep Scheduling Framework for WSN

Yanheng Liu<sup>1,2</sup>, Lei Zhang<sup>1</sup>, Aimin Wang<sup>1,2</sup> and Dayang Sun<sup>3,4</sup>

<sup>1</sup>*College of Computer Science and Technology, Jilin University  
Changchun 130012, China*

<sup>2</sup>*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of  
Education, Jilin University  
Changchun 130012, China*

<sup>3</sup>*College of Communications and Engineering, Jilin University  
Changchun 130012, China*

<sup>4</sup>*Corresponding author: E-mail: www.sunday@gmail.com*

### **Abstract**

*This paper presents an adjustable sleep scheduling framework for a variety of application in wireless sensor networks (WSNs) so that the lifetime of the network could be prolonged. Varies of typical applications in WSNs are analyzed to identify the same characteristics with which unified interfaces can be designed for all applications, and to identify different characteristics for which interfaces are reserved. This framework can meet the needs of varies applications by simple configuration and adjust itself according to real-time aspects. It can also avoid wasteful duplication, improve work efficiency and provide a good solution to the issue of application independent of sleep scheduling in WSNs. The results of experiments have shown the rationality and practicality of this framework and it can be applied to forest fire monitoring, crop water status monitoring, gas meter reading and other fields.*

**Keywords:** wireless sensor networks, sleep scheduling framework, application independent

### **1. Introduction**

Wireless sensor networks(WSNs) provide unprecedented opportunities for monitoring and controlling homes, cities, and the environment because of its low cost, low power consumption, unmanned and large internet capacity [1-2]. But restrained by the working conditions, the energy supply of the sensor network nodes is usually dependent on the battery which is difficult to be replaced or recharged for supplement. So energy supply has become the greatest limitation of wireless sensor networks [3]. To minimize energy consumption and prolong network lifetime, an important technique is to allow nodes to sleep periodically, which can be called sleep scheduling [4-5].

Some researchers introduce sleep scheduling into the design for MAC layer protocol of WSN, which can be called active/sleep duty cycle. The static low duty cycle assignment in SMAC [6] has been efficient in terms of energy conservation. SMAC lets nodes periodically sleep. During the sleeping part, a node turns off its radio to preserve energy. During the active part, it can communicate with its neighbors and send any messages queued during the sleeping part. In exchange some reduction in both per-hop fairness and latency is inevitable. In order to further reduce the waste of energy, TMAC [7] introduces an adaptive duty cycle in a novel way: by dynamically ending the active part of it. In order to achieve a good tradeoff between energy and latency, DSMAC [8] is proposed to support multiple duty cycles automatically adjusted, based on the

measurement of the energy consumption level and delay. Xiaodong Wang *et al.*, decompose the end-to-end delay guarantee problem into a set of single-hop delay guarantee problems along each data flow in the network and then formulate the single-hop delay guarantee problem as a dynamic feedback control problem and design the controller rigorously, based on feedback control theory, for analytic assurance of control accuracy and system stability [9]. And then Heejung Byun and Junglok Yu propose an adaptive duty cycle control mechanism based on the queue management with the aims of power saving and delay reduction<sup>[10]</sup>. The proposed scheme does not require explicit state information from the neighboring nodes, but only uses the local queue length available at the node.

With the continuous development of the theory, putting sleep scheduling into the various practical application of WSN has significant meanings.

In recent years, when researchers study applications of WSN, a large number of them have thought about using the sleep scheduling mechanism and have had some achievements. Literature [11] adopted WSN as the fundamental network infrastructure and developed a traceability system for recirculation aquaculture with integrated decision support function. The system enables rapid deployment and can acquire water temperature, salinity, dissolved oxygen and pH and achieve real-time data transmission. In order to save power, the node only activates when data acquisition happens. Once the data has been sent, the radio is turned off, an interrupt is enabled and a timeout timer is reset. In Literature [12], a TDMA based MAC protocol was used to collect environmental data such as soil moisture and temperature of an irrigation system. The base station was collecting the data in a particular area using the sensor nodes. TDMA scheduler assigned the time slots for each node and those sensor nodes turn ON/OFF their radio according to the schedule to save energy. If the collected data was less than the threshold value, the base station would inform the controller by setting the corresponding bit to perform a motor control action. Literature [13-14] presented Lofar Agro project that concentrated on monitoring micro-climates in a crop field. The pilot project concerned the protection of a potato crop against phytophthora, a fungal disease that can spread easily among plants and destroy a complete harvest within a large region. The authors described a precision agriculture architecture based on WSN. They deployed a large-scale sensor network (about 100 nodes) and acquired valuable experiences. Literature [15] described the results of real deployment of Automated Agriculture System which consists of WSN to monitor and control the environments and a management sub-system to manage the WSN, and provided various and convenient services to consumers who live in farming village with hand-held devices such as PDA. To conserve power, order-based sleep scheme is used. Whenever the sensing schedule is set by the application server, the sink node keeps the schedule and it spreads the sleep order message over its network every sensing period.

Researches mentioned above chose sensors, designed sleep scheduling intervals, wake-up modes, etc, depending on different applications. However, they have two drawbacks. 1) There are still many reusable functions in system designing for each application even if WSNs are application independent. It would lead to duplicate work redesign for each application; 2) when designing for an application, interfaces are neglected to remain for altering scheduling interval or wake-up threshold in sleep scheduling mechanism. If the application needs to alter the parameters due to the change of requirements (*e.g.*, information increase/decrease, emergency division adjustment), it would be repeated work to redesign or reconfigure. These two factors

increase the difficulty of adding sleep scheduling to WSN applications and restrict the promotion of sleep scheduling in practical applications.

Therefore, this paper followed the trend of applying sleep scheduling in WSN applications, summarized various applications, identified the same characteristics to give reusable design, different characteristics with interfaces reserved, and offered a service-oriented framework for the sleep scheduling in WSN. Given that it is difficult to adjust the sleep scheduling in MAC layer directly, the framework encapsulates the entire process and provides a unified service to upper application. Through this framework, scheduling interval, wake-up mode, *etc*, in the applications can be set conveniently, and the parameters can be adjusted during operating time. These improvements would greatly reduce the workload of applying sleep scheduling into the practical applications of WSN, improve work efficiency, and make sleep scheduling meet the needs of various services.

The rest of this paper is organized as follows: Section 2 presents the classification of WSN applications, Section 3 illustrates the Design of a sleep scheduling framework, Section 4 describes the implementation of the sleep scheduling framework, and then Section 5 verifies the feasibility of the sleep scheduling with experimental results. Finally, the paper gives summary conclusion of our work in Section 6.

## 2. Classification

WSN application has bright prospect. It can be applied to military applications, environmental applications, health applications, home applications, *etc*.

**Table 1. Comparison between Different Applications**

practical application	the data collected		time interval
vineyard computing	temperature, humidity of soil		1 min
habitat monitoring	environmental information		15 min
greenhouses monitoring	the environment of the greenhouse		5 min
practical application	the data collected		fixed threshold
fire monitoring	temperature	yellow/red alarm:40°C/60°C	
Anti-Theft Alarm System	PV panels position	5 degree displacement	

Table 1 gives a comparison between different WSN applications [15-19]. From the table, it is obvious that the importance of real-time property for different applications is various and as a result, the scheduling intervals are set differently. In addition, considering the diversity of collected data, nodes need different kinds of sensors, At last, the differences in trigger conditions during data communication lead to the differences in node's wake-up modes. In spite of the differences among them, there still exist some same characteristics: 1) Even through scheduling intervals are different, all applications must have scheduling intervals; 2) Sensors required are different, but all applications obtain data collection from the sensors.

Based on the study and summary of these same and different characteristics, two types of sleep scheduling strategy can be proposed to meet the requirement of specific applications: Class A is a kind of scheduling strategy collecting data periodically. Class B is a kind of scheduling strategy using special event to trigger the alarm. Class A is

applied to the type of application, which has fixed scheduling interval, wake-up periodically, trigger data transmission and regular sleep. Class B is for the type of application which has a fixed threshold set for a particular sensor. When the collected data reach the threshold (emergency), the node is forced to be wakened and trigger emergency processing. After completing the process, the node restores to its previous state. The applications in the table above belong to the Class A application, which collect data periodically. What designers need to do is simply set the length of the cycle, and writing the corresponding conversational function for the special sensor so as to complete the conversion among them quickly. As for the application of Class B, it is designed to perform special operations when the collected data reaches a certain critical condition, and it is applied to the dangerous situation detection.

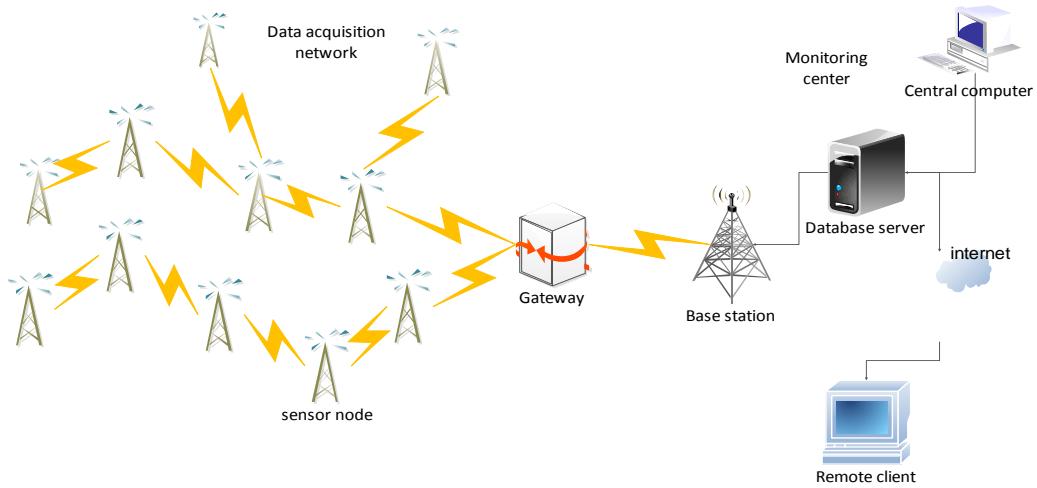
The two applications involve two different node wake-up modes: one is the regular wake-up mode which means that the length of the wake-up period is initialized firstly, and then nodes wake up and sleep periodically. The other one is mandatory wake-up mode, in which the nodes are forced to wake up when the collected data reaches the pre-set threshold.

Class A applications only use the regular wake-up mode. However, Class B applications use not only the regular wake-up mode but also the mandatory wake-up mode. Because the changing of requirements may lead to the modifications on threshold value and the wake-up interval length in class B applications, which requires the nodes not only to wake up periodically to see whether the new control command has arrived but also to wake up when threshold reached.

### **3. Design of Sleep Scheduling Framework**

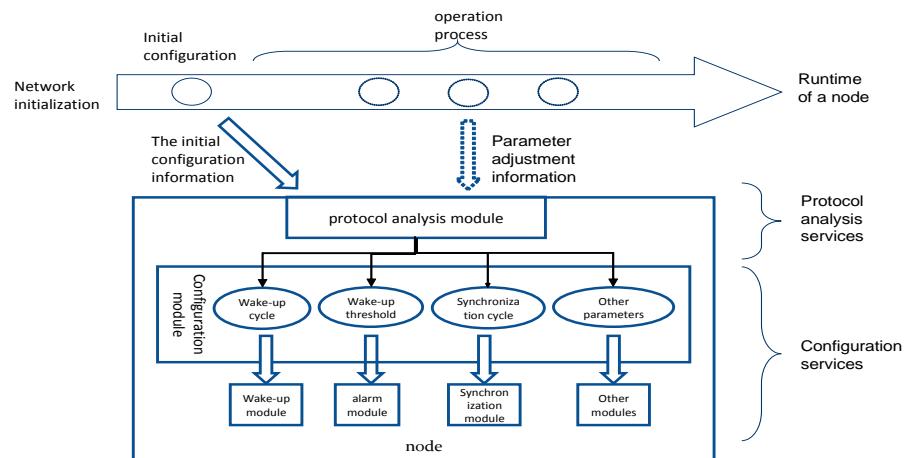
As it is mentioned above, sleep scheduling in WSN applications depends on the specific application, and there is no particular one method that can be applied to all applications. Because of different requirements in the same sensor network, sleep scheduling strategy is often different. It will be undoubtedly costly to design sensor network to design special sleep scheduling strategy for each application. In order to adapt to a variety of tasks, sensor networks need to choose a sleep scheduling strategy according to the application environment and requirements, and have the ability to adjust it with changing requirements. The sleep scheduling framework is introduced for this purpose.

The common practical applications of WSN mainly compose of two parts: data acquisition network and monitoring center. Data acquisition network is deployed in the monitoring region, and the monitoring center is located in a convenient place where people can access easily. The architecture and principle are shown in Figure 1.



**Figure 1. The General Architecture of the Practical Applications of WSNs**

According to the generic architecture of practical applications, the sleep scheduling framework is designed as in Figure 2.



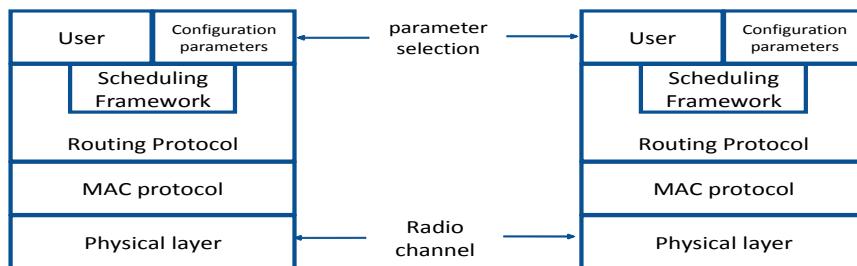
**Figure 2. Sleep Scheduling Framework in Nodes**

The arrow above is the timeline that indicates the whole process of a node, which is used to distinguish the initial configuration information from the later parameter adjustment information. After all nodes are started and the entire network is initialized, every collecting node will receive the initial configuration commands from the sink node. Then the protocol analysis module in node will parse the received commands, identify the configuration parameters needed for initialization and then configure the node. If a node receives a wake-up cycle parameters setting command, it will set the timer count parameter and then act on the wake-up module. If a node receives a wake-up threshold parameter setting command, it will set the threshold value and then act on the alarm module. If a node receives a synchronization cycle parameter setting command, it will set the synchronization cycle timer count and then act on the synchronization module. If a node receives other parameters setting command, it will set the corresponding variables and then act on the other modules.

The node will work well according to the initial configuration. If alterations on the parameters according to the change of requirements (*e.g.*, information increase/decrease, emergency division adjustment) are needed, the node will receive the parameter adjustment information from the sink node. Then the protocol analysis module in the node will parse the received commands, identify the configuration parameters which need to be adjusted and then do some corresponding work. This process is the same as the initialization process.

During this process, the wake-up module in the node will wake the node according to the wake-up cycle. The alarm module will compare the collected data with the threshold. When the collected data reach the threshold, the node is forced to be waked and the relevant work provided by the alarm module will be triggered. The synchronization module will update the node's schedule according to the synchronization cycle timer count to prevent long-time clock drift. The other modules and parameters provide interfaces for extending the sleep scheduling framework, and the users can customize the service to complete an additional function based on their requirements.

The sleep scheduling framework is used to set and adjust its parameters automatically according to the requirements. The framework encapsulates the entire process and provides a unified service to the upper application. The framework model between two nodes is designed as shown in Figure 3. The upper applications provide the parameters needed in sleep scheduling process and the sleep scheduling framework generates an appropriate sleep scheduling strategy based on this configuration and the change of network.



**Figure 3. Sleep Scheduling Framework between Nodes**

#### 4. Implementation of Sleep Scheduling Framework

Nowadays, Zigbee-compliant wireless sensor network platforms using TI CC2430 or CC2431 chipsets with Z-stack are widely used in practical applications. Paper [20] created a wireless healthcare monitoring platform using TI CC2430 or CC2431 chipsets with Z-stack. Paper [21] designed a Home Automation system based on the CC2430 chip. So according to the experimental requirements, this paper chooses choose TI CC2430 as the development tool, and provides an implemented program of the sleep scheduling framework.

The involved datagram in the entire sleep scheduling process are as follows: 0 real data, 1 adjust the wake-up cycle, 2 adjust the threshold, 3 keep awake, 4 confirm waked, 5 restore, 6 synchronization, 7 quit.

In order to facilitate programming, macro definition can be given as shown in Table 2.

**Table 2. Type Field**

#defi ne	DA TA	C_W AK	C_T HR	C_K EP	C_C ON	C_R ES	C_S YN	QUI T
	0	1	2	3	4	5	6	7

To meet the application, the format of a datagram can be designed as shown in Figure 4.

Destination address(16)	Source address(16)	Length(12)	Type(4)	Data	CRC(16)
-------------------------	--------------------	------------	---------	------	---------

**Figure 4. Datagram Format**

Destination address field and source address field belongs to the mechanisms in the application layer and they are used to distinguish different nodes. They can be physical addresses, network addresses or also node marks as long as different nodes can be distinguished using them.

The length field indicates the length of the data.

The type field indicates different data content, as shown in Table 3.

**Table 3. Type Field**

type	data content
DATA	real data
C_WAK	control commands which adjust the wake-up cycle
C_THR	control commands which adjust the threshold
C_KEP	control commands which keep awake
C_CON	control commands which confirm waked
C_RES	control commands which restore
C_SYN	control commands which maintain synchronization
QUIT	control commands which quit

The data field is determined by the type field, as shown in Table 4.

**Table 4. Data Field**

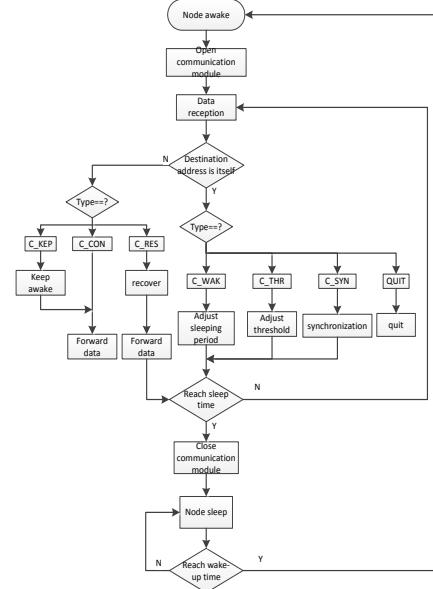
type	data field
DATA	the real data while communicating
C_WAK	the wake-up cycle timer count
C_THR	the wake-up threshold
C_KEP	address information of the node that is forced to wake up
C_CON	address information of the node that is forced to wake up
C_RES	address information of the node that is forced to wake up
C_SYN	the next-sleep time
QUIT	null

CRC field indicates the check sequence.

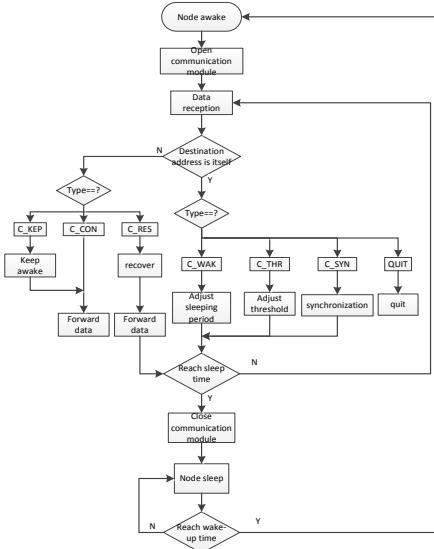
The parameters adjustment relates to two types of applications: A and B.

For type A, at beginning, the node wakes up and opens communication module simultaneously. Secondly, it opens the sensor module, collects data and sends out the data. Thirdly, the node does the receiving task and judges whether the destination address of the received datagram is itself. If not, the node transmits the datagram according to the routing table. If so, it does the work according to the type of the received datagram. If the type is DATA, it means that the datagram is a data datagram. If the type is C\_WAK, it means that the datagram is a control command which can adjust the wake-up cycle according to the datagram's parameter field. If the type is C\_SYN, which means that it is the time to maintain synchronization, the node should adjust the next-sleep time according to the datagram's parameter field. If the type is QUIT, which means quit, the node should stop working.

And then the node judges whether it is the time to sleep during this period. If so, the node closes the communication module, saves the uncompleted work in the storage device and sleeps until the next wake-up. If not, the node goes on working. The whole process of type A is shown in Figure 5.



**Figure 5. Type A Application Parameters Adjustment Flowchart**



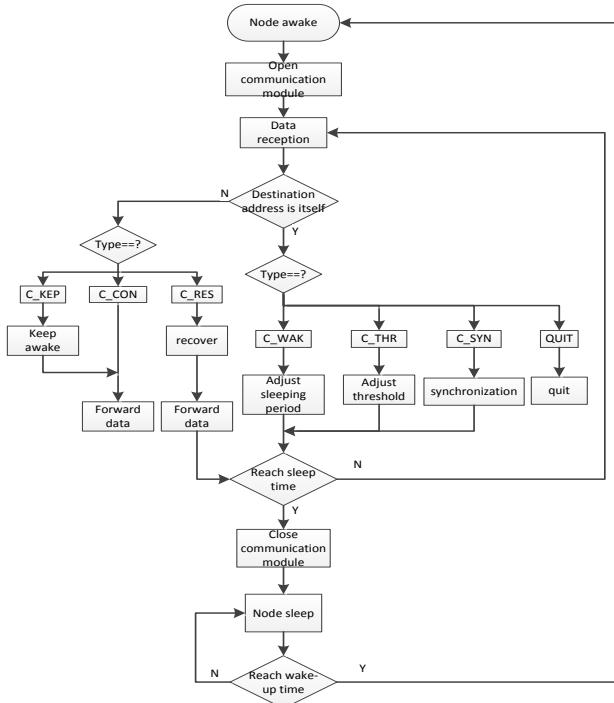
**Figure 7. Type B Application Forced Wake-Up Parameters Adjustment Flowchart**

The wake-up mode of type B application is divided into the regular and the forced wake-up.

For the regular wake-up, at beginning, the node wakes up and opens communication module simultaneously. Secondly, the node does the receiving task and decides whether

the destination address of the received datagram is itself. If not, the node transmits the datagram according to the routing table. If so, it does the work according to the type of the received datagram. If the type is DATA, it means that the datagram is a data datagram. If the type is C\_WAK, it means that the datagram is a control command which can adjust the wake-up cycle according to the datagram's parameter field. If the type is C\_THR, it means that the datagram is a control command which can adjust the threshold according to the datagram's parameter field. If the type is C\_KEP, it means that the datagram is a control command to keep awake. The node checks itself to judge whether it is already in keeping awake state. If not, the node turns into keeping awake state and sends the datagram with the C\_KEP in the type field and the neighbor's address leading to the sink node in the routing table in the destination address field. Meanwhile it sends a confirmed datagram with the C\_CON in the type field to the node which sends the wake-up command. If so, it judges whether the node is the sink node. If the node is the sink node, it sends a confirmed datagram with the C\_CON in the type field to the node which sends the wake-up command. If the type is C\_CON, which means that the way to the sink node has established successfully, the data can be sent. If the type is C\_RES, which means that the incident is completed, the previous state and sleep counter of the node should be restored, and the node goes back into the circulation. If the type is C\_SYN, which means that it is the time to maintain synchronization, the node should adjust the next-sleep time according to the datagram's parameter field. If the type is QUIT, which means quit, the node should stop working.

And then the node judges whether it is the time to sleep in this period. If so, the node closes the communication module, saves the uncompleted work in the storage device and goes on until the next wake-up. If not, the node goes on monitoring the region. The whole process of type B application regular wake-up is shown in Figure 6.

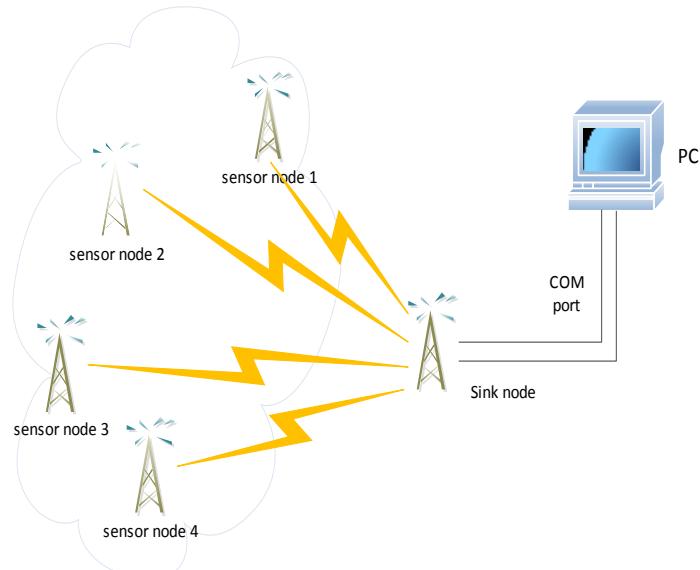


**Figure 6. Type B Application Regular Wake-Up Parameters Adjustment Flowchart**

For the forced wake-up, the node compares the collected data with the threshold. When the data reach the threshold, the node is forced to wake up. Then considering three scenarios, the first case is that the node can communicate with the sink node directly; the second case is that the node can't communicate with the sink node directly, but other nodes are in the awake state at the same time; the third case is that the node can't communicate with the sink node directly, and meanwhile other nodes are also in the sleep state. For the first case, the node simply needs to set itself into keeping awake state and the pathway to the sink node can be established successfully; for the second case, the node need to set itself into keeping awake state, and send the datagram with the C\_KEP in the type field and the neighbor's address leading to the sink node in the routing table in the destination address field, the node which receives this datagram replies a confirm datagram with the C\_CON in the type field, and continues to send the datagram with the C\_KEP in the type field and the neighbor's address leading to the sink node in the routing table in the destination address field. The process doesn't stop until reach the sink node. And then the pathway is established; for the third case, there are only a few changes with the second case. The node sends the datagram with the C\_KEP in the type field and the neighbor's address leading to the sink node in the routing table in the destination address field at the time of the next wake-up instead of right now. This can establish a pathway to the sink node and complete the work when it reaches the threshold it should do. After the work, the sink node sends the recovery command along the path, and the node which receives the recovery command calls off the keeping awake state, and goes back into the sleep and wake-up cycle. The whole process of type B application forced wake-up is shown in Figure 7.

## 5. Experiment

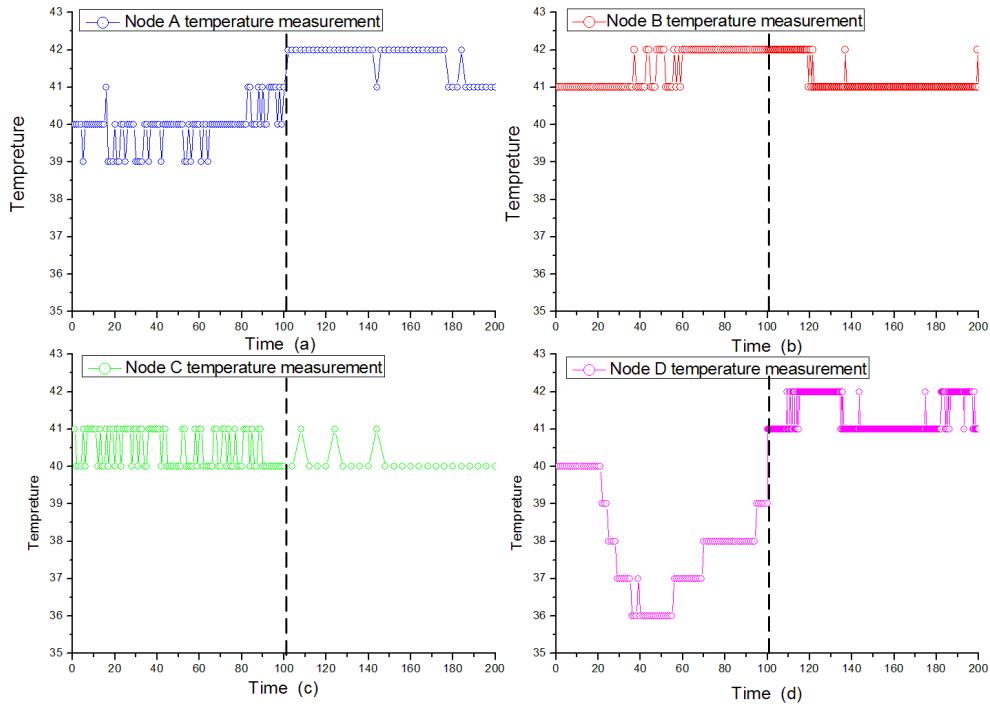
The experiments were arranged at the WSN laboratory of Jilin University. The indoor temperature was about 23°C. Only the feasibility of the sleep scheduling framework is wanted to be verified, generic architecture in the practical application of WSN can be simplified.



**Figure 8. Experimental Schematic**

The main steps are described as follow. Connect the sink node to PC through a COM port, and then build a star network with four sensor nodes. Using temperature probe in TI CC2430 chipset to collect the temperature data of the chip, combine with the received data displayed from Com Wizard to realize the gather of chip's temperature and abnormal alarm by modifying and transplanting ZStack-1.4.3-1.2.1 of Zigbee2006 of the TI.

*Experiment 1* is to verify the sleep scheduling framework provided by this paper has the ability of modifying the sample interval dynamically. At the time of network initialization, the all four nodes' sampling intervals were set to 1 minute. When network was operating normally, each sensor node sent its own temperature information to the sink node every 1 minute, and the temperature information of each node which was distinguished by the assigned short address when it jointed the network were displayed on the PC screen. After a period, the sink node sends control commands which change the sampling interval of sensor node. In order to highlight the changes of the sampling interval, the commands with different sampling interval were sent to the four nodes. Since then, each sensor node would send its own temperature information according to its own sampling interval. The experimental data were analyzed and processed to obtain the temperature information of the four nodes, as shown in Figure 9.

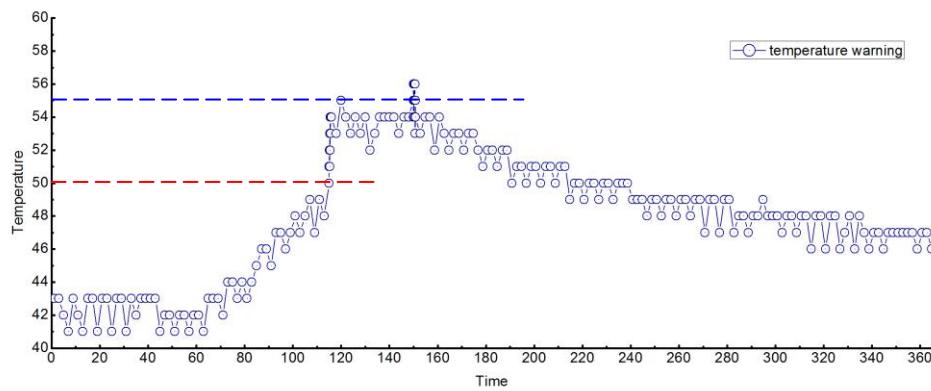


**Figure 9. Temperature Information of the Four Nodes**

The ordinates of the four nodes all represent the temperature, and the abscissas of the four nodes all indicate the change of time. From the diagram we can clearly see that, with the moment of 100 as the boundary line, the sampling interval of each node is

altered. The sink node received temperature information of the node A, B, C and D every 1 minute before the line, and then node A changes into receiving temperature information of the node every 2 minutes after the line; node B changes into every 30 seconds; node C changes into every 4 minutes and node D changes into every 15 seconds.

*Experiment 2* is to verify the sleep scheduling framework provided in this paper has the ability to set and alter the threshold of alarm. The alarm process will be triggered when the collected data reaches the threshold which is set at the time of network initialization or altered during the working time. At the time of network initialization, the threshold was set to 50°C. When the network was operating normally, each sensor node collected the temperature information of its monitoring the region as setting at the time of network initialization. After a period, a cup of hot water(about 75°C) was placed near one sensor node to make its temperature rise. When the temperature on the node reached 50°C, the node gave alerts, and the sample interval was set to 1 second automatically in order to monitor the emergency region continuously. And then the threshold was altered to 55°C, because the node's temperature is below the threshold, the node restored the previous state and the sample interval restored. When the temperature on the node reached 55°C, the node gave alerts, and the sample interval was set to 1 second automatically again. After a period, that cup of hot water was taken away from the node, and then the temperature of the node would reduce gradually. When the temperature was below 55°C, the node restored the state before alerts and monitored the region sequentially. By analyzing and processing the experimental data, the temperature information of the node can be obtain as shown in Figure 10.



**Figure 10. Temperature Information of the Incident Regional Node**

The ordinate represents the temperature of the node and the abscissa indicates the change of time. From the diagram we can clearly see that, at beginning, with the temperature of 50°C as the boundary line, below the red line the sink node received temperature information of the node every 2 minute, and then changed into receiving temperature information of the node every 1 second above the red line. At the moment

of 115, the threshold was altered to 55°C. So that the sink node received temperature information of the node every 2 minute again. Then the boundary line turns to the temperature of 55°C. Below the blue line the sink node received temperature information of the node every 2 minute, and then changed into receiving temperature information of the node every 1 second above the blue line. When the temperature is below 55°C, the sink node restored to receive temperature information of the node every 2 minute.

Experimental results indicate that: 1) the sleep scheduling framework realized in this paper is able to configure the sampling interval, alarm threshold conveniently; 2) the sleep scheduling framework successfully realizes the dynamical adjustment of sampling interval, alarm threshold.

## 6. Conclusion

This paper divides the practical application of WSN into two classes of A and B, illustrates the differences between them and their respective application scenarios, proposes an adjustable sleep scheduling framework, and explores the entire implementation process of the framework. It also gives a solution to the problem which is difficult to adjust the sleep scheduling in MAC layer directly through encapsulating the entire process and providing a unified service to upper application. Experiment results verify the feasibility of the study in this paper, which make the sleep scheduling configuration in the WSN applications agile and convenient, avoid the repetition labor, improve work efficiency, and solve the issue of the strong application correlation of sleep scheduling in a better way. However, the implementation of the sleep scheduling framework in this paper is in the experimental level, so we will consummate the framework's implementation in the application level for applying in the real applications in further research.

## Acknowledgement

This work was supported by Natural Science Foundation of China under Grant No. 60973136, 61073164 and Program of International Science and Technology Cooperation under Grant No. 2008DFA12140.

## References

- [1] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges", Proc. of the IEEE, vol. 91, no. 8, (2003), pp. 1247-1256.
- [2] 21 ideas for the 21st century," Business Week, (1999) August 30, pp. 78-167.
- [3] C. Park and P. Chou, "Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes", in Proc. SECON, vol. 1, (2006), pp. 168-177.
- [4] J. Ma, Dept. of Comput, Hong Kong Polytech, Univ., K. W. Lou ,Y. Wu , X.-Y. Li and G. Chen, "Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks", Conference, (2009) April, pp. 19-25.
- [5] L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks". [J] . ACM/Kluwer Mobile Networks and Applications, vol. 11, no. 5, (2006) October, pp. 723-740.
- [6] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks", in IEEE Infocom, (2002).
- [7] T. van Dam and K. Langendoen, "An adaptive energy-effcient mac protocol for wireless sensor networks", in SenSys, (2003).
- [8] P. Lin, C. Qiao and X. Wang, "Medium Access Control with a Dynamic Duty Cycle for Sensor Networks", IEEE WCNC, vol. 3, (2004) March, pp. 1534-39.
- [9] X. Wang, G. Xing and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks", International Workshop on Quality of Service (IWQoS), (2010), pp. 1-9.
- [10] H. Byun, "Adaptive Duty Cycle Control with Queue Management in Wireless Sensor Networks", IEEE Transactions on Mobile Computing, (2012).

- [11] L. Qi, J. Zhang, M. Xu, Z. T. Fu, W. Chen and X. S. Zhang, "Developing WSN based traceability system for recirculation aquaculture", Mathematical and Computer Modeling, vol. 53, (2010), pp. 2162-2172.
- [12] M. N. Sudha, M. L. Valarmathi and A. S. Babu, "Energy efficient data transmission in automatic irrigation system using wireless sensor networks", Computers and Electronics in Agriculture, vol. 78, (2011), pp. 215–221.
- [13] A. Baggio, "Wireless sensor networks in precision agriculture", proceedings of REALWSN'05, Stockholm, (2005) June 20-21.
- [14] K. Langendoen, A. Baggio and O. Visser, "Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture", 14th 14th Int. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Rhodes, Greece, (2006) April.
- [15] S. Yoo, J. Kim, T. Kim, S. Ahn, J. Sung and D. Kim, "A2S: Automated agriculture system based on WSN," in Proc. IEEE Int. Symp. Consum. Electron, (2007) June, pp. 1–5.
- [16] J. Burrell, T. Brooke and R. Beckwith, "Vineyard computing:Sensor networks in agricultural production", IEEE Pervasive Computing, vol.3, no.1, (2004), pp. 38–45.
- [17] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless sensor networks for habitat monitoring", In ACM International Workshop on Wire-less Sensor Networks and Applications (WSNA'02), Atlanta, GA, (2002) September.
- [18] L. Bernardo, R. Oliveira and R. Tiago R, "A fire monitoring application for scattered wireless sensor networks", the proceeding of WinSys, (2007), pp. 28-31.
- [19] S. Bertoldo, O. Rorato and C. Lucianaz C, "A Wireless Sensor Network Ad-Hoc Designed as Anti-Theft Alarm System for Photovoltaic Panels", Wireless Sensor Network, vol. 4, no. 4, (2012), pp. 107-112.
- [20] L. Yang, M. Ji, Z. Gao, W. Zhang and T. Guo, "Design of Home Automation System Based on ZigBee Wireless Sensor Network", International Conference on Information Science and Engineering, vol. 10, (2009), pp. 2610–2613.
- [21] Yang Li and Ji Maorong, "Design of Home Automation System based on ZigBee Wireless Sensor Network", The 1st International Conference on Information Science and Engineering, (2009).

## Authors



**Yanheng Liu**, he was born in 1958. Professor of JiLin University. His main research interests include network security, network management, mobile IP and QoS.



**Lei Zhang**, he received his B.Sc. in Computer Science and Technology (2011) from JiLin University. His main research interests include wireless sensor network and network management.



**Aimin Wang**, he was born in 1970. Associate professor of JiLin University. His main research interests include ad hoc, mobile communication, multimedia communication, and wireless sensor network.



**Dayang Sun**, he was born in 1979. Lecturer of JiLin University. His main research interests include network security and wireless sensor network.

