An Energy-Efficient MAC Design for IEEE 802.15.4-Based Wireless Sensor Networks

Yu-Kai Huang[1], Sze-Wei Huang[1], and Ai-Chun Pang[12]

[1] Graduate Institute of Networking and Multimedia
[2] Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan 106, ROC
{d94944009,r93944018,acpang}@csie.ntu.edu.tw

Abstract. This paper presents a new energy-efficient MAC design to improve the performance of IEEE 802.15.4-based wireless sensor networks. Our proposed mechanism adaptively determines the sleeping schedules of sensor nodes based on the network traffic load to achieve the balance of throughput and energy consumption. This mechanism consists of two phases: schedule exchange phase and schedule generation phase. In the schedule exchange phase, the schedule parameters are piggybacked in normal transmissions. In the schedule generation phase, sensor nodes adaptively determine the sleeping schedule from the schedule parameters. Eventually, the schedules of all sensor nodes converge to one schedule. The experimental results show that the proposed mechanism achieves sleeping schedule convergence and high energy efficiency.

Keywords: Power Saving, Energy Efficiency, Wireless Sensor Networks, IEEE 802.15.4, Low Rate Wireless Personal Area Networks (LR-WPANs)

1 Introduction

Wireless sensor networking is an emerging technology that has a wide range of potential applications including animal/plant habitation monitoring, target tracking, building monitoring, and robotic exploration. Such networks consist of large numbers of distributed nodes that organize themselves into multi-hop wireless systems. The sensor nodes are usually operated by batteries to simplify network deployment. With many nodes placed in their target environment, recharging batteries becomes more difficult, or even impossible. Therefore, energy efficiency has been a critical issue in wireless sensor networks.

The design of medium access control (MAC) plays an important role for energy efficiency of sensor nodes. In the MAC layer, most of the energy wastage comes from *idle listening*. Since sensor nodes do not figure out when it becomes the receiving side of a message from one of its neighbors, the sensor nodes have to turn on its radio receiver all the time and to keep listening even if the nodes are in the idle mode. The previous work has shown that *idle listening* consumes additional 50% to 100% of the energy [1]. To minimize the energy consumption caused by *idle listening*, an intelligent MAC algorithm shall be developed to make its best effort to turn off the radio when sensor nodes are idle.

The recent works such as SMAC [2] [3] and TMAC [4] have adopted a synchronized sleep/wakeup cycle to allow nodes to operate at low duty cycle for power saving. SMAC reduces idle listening by periodically putting nodes into sleep state. TMAC is an improvement of SMAC. In TMAC, if there is no activity in the vicinity of a node for a time T_A, the node will go to sleep for reducing idle listening. TMAC has the same performance as SMAC under constant traffic loads, but it saves more energy under a light-traffic condition. Also, TMAC uses the control packets RTS/CTS to exchange energy-consumption information. Both SMAC and TMAC can not adapt traffic variation well since its static design for fixed sleeping schedules of sensor nodes.

The release of IEEE 802.15.4, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)"[7] represents a milestone in wireless personal area networks. IEEE 802.15.4 is a new standard uniquely designed for low rate wireless personal area networks(LR-WPANs). It targets ultra-low complexity, cost, and power for low-data-rate wireless connectivity among inexpensive fixed, portable, and moving devices [8]. The standard also supports multi-hop packet delivery. Therefore, in comparison with 802.11, 802.15.4 is more suitable for wireless sensor networks.

Since IEEE 802.15.4 is a good alternative for wireless sensor networks, there are several issues about the sleeping schedule design. The system performance of a wireless sensor network is affected seriously by these issues.

– *Sleeping Schedule Adaption for Traffic Variation*
  Since the traffic load of a wireless sensor network is not necessarily the same all the time, a fix sleeping schedule does not cope with the traffic variation. A duty cycle for heavy traffic loads results in energy wastage when the traffic becomes light, while the duty cycle for light traffic loads causes low throughput under increasing traffic loads. Particularly, in wireless sensor networks, the occurrence of emergencies generates heavy traffic in a short time such that the system performance seriously degrades. Therefore, it is required to adaptively determine the duty cycle according to the traffic situation.
– *The Occurrence of Multiple Schedules*
  A sleeping schedule protocol establishes and maintains sleeping schedules for a wireless sensor network. When a node fails to hear an existing schedule, it shall create a new schedule for itself. In a large network, it is expected that a number of nodes create their own schedules. Therefore, multiple schedules occur in a large network [9]. Multiple schedules result in energy wastage and long latency for retransmission.
– *The Control Overhead*
  When multiple schedules occur, the existing protocols usually use additional packets to exchange schedule information. The schedule exchange overhead causes additional power consumption. Therefore, it is necessary to simplify schedule exchange mechanism.

In this paper, we propose a new energy-efficient MAC design for IEEE 802.15.4-based wireless sensor networks. To consider the above issues, our MAC design adaptively determines the sleeping schedule based on the traffic load. To reduce the control overhead, devices piggyback schedule parameters embedded in data and acknowledgment packets. Therefore, the additional packets are avoided. Afterward, a device uses its neighbors' schedule parameters to adjust its own schedule. This mechanism converges multiple schedules to a single schedule gradually. The performance evaluation for our MAC design is conducted through the well known NS-2 simulator. Simulation results indicate that in terms of energy consumption and power efficiency, the proposed mechanism outperforms the legacy 802.15.4 MAC protocol.

The remainder of this paper is organized as follows: Section 2 describes the MAC protocol for IEEE 802.15.4. In Section 3, we formally define the problem, and propose an MAC design to adaptively determine the sleeping schedule for a wireless sensor network. In Section 4, the capability of the proposed MAC design is investigated under a series of experiments. Section 5 is the conclusion.

2 IEEE 802.15.4 MAC

The new IEEE standard, 802.15.4 defines the physical layer (PHY) and medium access control (MAC) sublayer specifications for low-rate wireless personal area networks (LR-WPANs), which supports simple devices that consume minimal power and typically operate in the personal operating space of 10 m or less. Thus, many corporations manufacture new wireless sensor boards for IEEE 802.15.4[10][11][12]. The IEEE 802.15.4 defines two different modes for medium access: beacon-enabled mode and nonbeacon-enabled mode.

2.1 Nonbeacon-Enabled Mode

In nonbeacon-enabled mode of 802.15.4, unlike 802.11, the data transfer model is quite simple and does not need to use the additional control messages. When a device wishes to transfer data, it simply transmits its data frames using unslotted CSMA-CA mechanism. The receiver acknowledges the successful reception of the data by transmitting an optional acknowledgment frames.

The nonbeacon-enabled networks use an unslotted CSMA-CA channel access mechanism. Each time a device wishes to transmit data frames or MAC command, it shall wait for a random period. If channel is found to be idle, following the random backoff, the device will transmit its data. If channel is found to be busy, following the random backoff, the device shall wait for another random period before trying to access the channel again. Acknowledgment frames shall be sent without using a CSMA-CA mechanism.

In the nonbeacon-enabled networks, the coordinator do not use beacon frames to synchronize the attached devices, and any devices can communicate with any other device as long as they are in range of one another. Therefore, the nonbeacon-enabled networks are easy to enlarge the network scale and suitable for a large scale sensor network. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, and intelligent agriculture would benefit from nonbeacon-enabled networks [14]. In the beacon-enabled mode, there is power-saving mechanism that the superframe can have an active and an inactive period, and all nodes may enter a low-power mode during inactive period [15]. However, there is no power-saving mechanism in the nonbeacon-enabled networks. Thus, we proposed an energy-efficiency MAC design for the nonbeacon-enabled networks, such that the nonbeacon-enabled networks achieve more power savings.

3 The Energy-Efficient MAC Design

In this section, we propose the energy-efficient MAC design for wireless sensor networks. Our MAC design includes two phases: *Schedule Exchange* and *Schedule Generation*. The *Schedule Exchange* phase operates during the awaken period of a sensor node. The phase is composed of two components. One is used to exchange schedule parameters among sensor nodes, and the other one counts the communication frequency representing the traffic load between two neighboring sensor nodes. To reduce the control overhead for parameter exchanging, the schedule parameters are piggybacked in data and acknowledgment packets.

The *Schedule Generation* phase operating in the sleeping period generates a new schedule for a sensor node, and slightly adjusts the schedule based on the traffic load. A new schedule is adaptively determined based on the schedule parameters and communication frequency derived in the *Schedule Exchange* phase. If there is any extreme traffic situation such as bursty packet arrivals, the new schedule will be adjusted to avoid starvation.

3.1 System Parameters

Before describing the details of our design, we elaborate the system parameters used in the *Schedule Exchange* phase.

In DATA packets

•W: The waiting time of data packets. This parameter is used in schedule adjustment of the *Schedule Generation* phase, and is used to avoid the starvation. A source node transmits a data packet, and the destination node gets W embedded in the source node's packet. If the waiting time is larger than the predefined threshold, the destination node prolong the awaking period.

In ACK packets

•S: The time that the node will sleep. The destination node sends the ACK packet with this parameter to the source node, and the source node knows when the destination node will sleep.

•Lw: The duration that the node wakes up. The node records the neighbors' Lw in the neighbor list, and uses these Lw and the communication frequency to generate the node's new Lw.

•Ls: The duration that the node sleeps. The node records the neighbors' Ls in the neighbor list, and uses these Ls and the communication frequency to generate the node's new Ls.

Communication Frequency C_n: The number of communications between a node n and its neighboring node in the awaken period. Each node will count and record communication frequency of its neighboring nodes, and the communication frequency of each neighboring node becomes a weight for the generating of a new schedule.

3.2 Schedule Exchange Phase

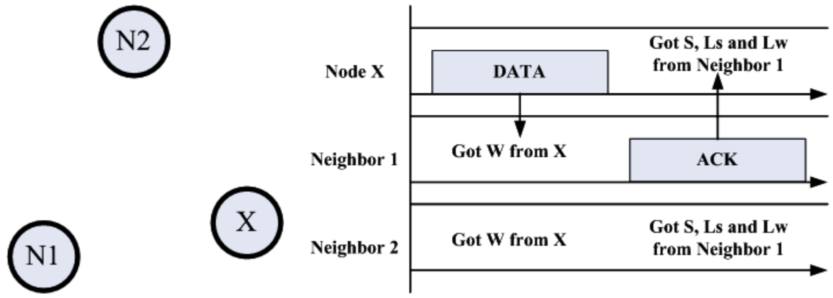
In this phase, schedule parameters are exchanged, and the communication frequency is recorded. Every node maintains a neighbor list storing schedule parameters and the communication frequency of every neighbor. Schedule parameters are piggybacked in the DATA and ACK packets. When nodes send packets, schedule parameters are also exchanged among these nodes.

When a node receives a data packet, it records the waiting time of the source node. Then the node responds an acknowledgement packet back to the source node with its schedule parameters. The node receiving the acknowledgement updates the schedule parameters in the neighbor list. If other neighbors of the two node are awake, they can get the schedule parameters by overhearing the data and acknowledgement packets.

We briefly describe the schedule exchange operation through the following example (see Figure 1). In this example, there are three nodes in the sensor network and Figure 1(a) shows the sensor network topology. In Figure 1(b), node X sends a data packet with waiting time W to neighbor 1, and neighbor 1 receives the data packet and gets node X's waiting time W. At the same time, neighbor 2 overhears the data packet from node X and gets the node X's waiting time W. After received the data packet, neighbor 1 sends an acknowledgement packet with schedule parameters(S, Ls and Lw) back to node X, and node X receives the acknowledgement packet and updates neighbor 1's schedule parameters in the neighbor list. In the same way, Neighbor 2 will get neighbor 1's schedule parameters by overhearing the acknowledgement packet from neighbor 1.

3.3 Schedule Generation Phase

In this phase, we use schedule parameters mentioned in the previous section to generate and adjust the new schedule. This phase is to adaptively determine nodes' schedules based on the traffic load and converge multiple schedules to a single schedule. A node adopts a neighbor's communication frequency as the neighbor's weight, and average the weighted schedules of all neighbors as the node's new schedule. When a node enters the sleeping state, it computes the new schedule by the schedule parameters based on the communication frequency recorded in the neighbor list during the awaken period. Nodes adjust schedules in two situations. One is to cope with the extreme traffic situation after schedule


N2
N1
X
Node X
DATA
Got S, Ls and Lw from Neighbor 1
Neighbor 1
Got W from X
ACK
Neighbor 2
Got W from X
Got S, Ls and Lw from Neighbor 1

(a) Sensor Network Topology

(b) Schedule Exchange Operation

Fig. 1. An Example of Schedule Exchange

generation phase, and the other is to prolong the awaken period to avoid the starvation according to the waiting time W piggybacked in data packets from one of neighbors.

Schedule Generation Operation To make multiple schedules converge to one schedule, nodes estimate the next time that neighbors sleep and wake up for schedule generation. The estimation method considers three cases, and we use schedule parameters(S, Ls and Lw) to derive two variables(P_n and X_n) for the three cases. P_n is a estimated time that the neighbor n will wake up in this or next schedule after the node wakes up. X_n is the duration of that the node should be awaken, and it is calculated by S ,Ls and Lw for the neighbor n.

- Case 1: $S_n > S + L$.
  In this case, S_n, the sleeping time of the neighbor n is larger than the time $S + L$ that the node X will wake up. Therefore, the node X directly set the time that the neighbor n will wake up in this schedule as P_n and the duration from that the node X will wake up to the sleeping time of the neighbor n as X_n(see Figure 2(a)).
$$P_n = S_n + Ls_n \tag{1}$$
$$X_n = S_n - (S + Ls) \tag{2}$$
- Case 2: $S_n < S + L$ and $S_n + Ls_n > S + L$.
  In this case, S_n, the sleeping time of the neighbor n is smaller than the time $S + L$ that the node X will wake up but the time $S_n + L_n$ that the neighbor n will wake up in this schedule is larger than the time $S + L$. Therefore, the node X estimates P_n by increasing the duration of this schedule of the neighbor n, $Lw_n + Ls_n$. X_n is the duration from that the node X will wake up to the estimated sleeping time of the neighbor n (see Figure 2(b)).
$$P_n = S_n + Ls_n + Lw_n + Ls_n \tag{3}$$

$$X_n = S_n + Ls_n + Lw_n - (S + Ls) \tag{4}$$

- Case 3: $S_n + L_n \leq S + L$.
  In this case, the time $S_n + L_n$ that the neighbor n will wake up in this schedule is equal to or smaller than the time $S + L$ that the node X will wake up. In other words, P_n will be the time $S_n + Ls_n$ that the neighbor n wakes up plus m times the duration of this schedule of the neighbor n until P_n is larger than $S + Ls$ and the value of m is equal to or larger than 0. X_n is the duration from that the node X will wake up to the last estimated sleeping time of of the neighbor n (see Figure 2(c)).

$$\begin{aligned} &P_n = S_n + Ls_n + m \times (Lw_n + Ls_n) + \\ &Lw_n + Ls_n, \text{where } m \geq 0 \end{aligned} \tag{5}$$

$$\begin{aligned} &X_n = S_n + Ls_n + m \times (Lw_n + Ls_n) + \\ &Lw_n - (S + Ls), \text{where } m \geq 0 \end{aligned} \tag{6}$$

To determine adaptively schedules based on the traffic, nodes use the communication frequency as the weight for the schedule generation. In the schedule generation phase, nodes generate the new schedule parameters by the weighted P_n and X_n that are multiplied by the communication frequency of the neighbor n (see equation 7,8 and 9).

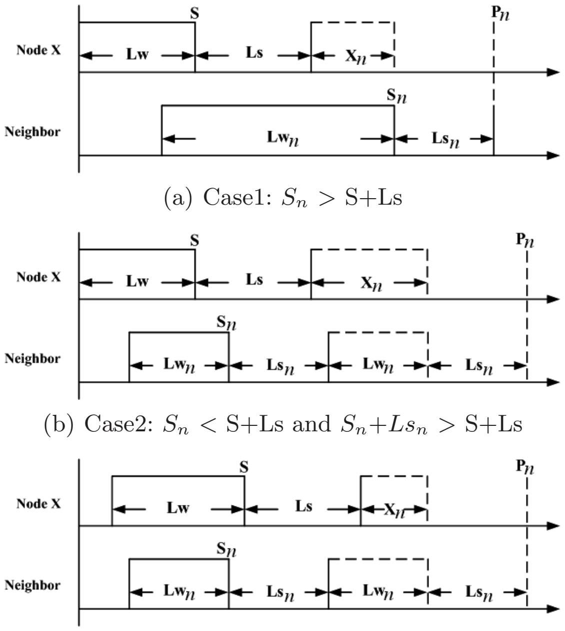
$$Lw_{new} = \frac{\sum_n (X_n \times C_n)}{\sum_n C_n}, \text{where } n \in \text{All Neighbors} \tag{7}$$

$$S_{new} = (S + L) + Lw_{new} \tag{8}$$

$$\begin{aligned} &Ls_{new} = \tfrac{\sum_n (P_n \times C_n)}{\sum_n C_n} - S_{new}, \\ &\text{where } n \in \text{All Neighbors} \end{aligned} \tag{9}$$

Schedule Adjustment Operation To cope with the extreme traffic situation, nodes adjust its schedule for better throughput and energy dissipation reduction. For example, when the traffic load becomes heavy suddenly, nodes should decease the sleeping period and increase the awaking period for the throughput. Moreover, if there is no traffic between a node and its neighbors, the node should adjust its schedule by increasing the sleeping period and decreasing the awaking period for reducing energy dissipation. In the schedule adjustment, nodes adjust the sleeping period and the awaking period by slow-start algorithm[16]. Nodes increase the period exponentially (period = period × 2) until a predefined slow-start threshold is reached. Once the threshold is reached, the period is increased linearly (period = period + 1).

Another schedule adjustment is that a node prolongs the awaking period by waiting time W to avoid the starvation of its neighbor. When the node receives



(a) Case1: S_n > S+Ls

(b) Case2: S_n < S+Ls and S_n+Ls_n > S+Ls

(c) Case3: S_n+Ls_n ≤ S+Ls

Fig. 2. P_n and X_n Estimation of Schedule Generation

data packets from one of neighbors, it will get the waiting time W of the neighbor. If the waiting time W from the neighbor is larger than a predefined threshold, the node will prolong the awaking period immediately. On the other hand, if the waiting time W is smaller than the threshold, the node will not do the schedule adjustment by the waiting time W.

4 Performance Evaluation

This section investigates the performance of our energy-efficient MAC design for IEEE 802.15.4-based wireless sensor networks.

4.1 Simulation Environment

The simulation experiments are conducted through NS-2 [17] simulator, and the simulation time lasts for 5000 seconds. In the experiments, a sensor node generates normal messages with constant inter-arrival times, and generates emergency messages with 1 time per 100 seconds. Two simulation topologies are considered in the simulation. One is a simple chain with 10 nodes and the other is a random topology with 25 nodes. In the chain topology, the source node (i.e., the first node of the chain) sends packets to the destination node (i.e., the last node of the chain). In the random topology, there are two source nodes, and the source nodes will send packets to the two destination nodes across the topology.

4.2 Input Parameters and Output Measures

We use IEEE 802.15.4 MAC implementation in NS-2 that follows the specification of IEEE 802.15.4 MAC layer. The input parameters are listed described as follows.

• **Normal traffic**. We change the traffic load by varying the inter-arrival period of messages. We assume that the message inter-arrival period varies from 1 to 10 seconds.

• **Emergency traffic**. We use the emergency traffic to cause the traffic unstable. We assume that an emergent message is generated per 0.2 seconds.

• **Energy Consumption**. To consider the energy consumption of a node, we measure the amount of time that the radio on each node has spent in different modes: sleep, idle, receiving or transmitting. The energy consumption in each mode is calculated by multiplying the time with required power listed in Table 1 to operate the radio in that mode.

Parameters	value
Initial Energy	1000Joules
Idle Power	712μWatts
Transmission Power	31.32mWatts
Receiving Power	35.28mWatts
Sleeping Power	144nWatts
Number of devices	$10, 25$

Table 1. System Parameters

The output measures we adopt in the simulation are described as below.

• **Average Energy Consumption** The average energy consumption is obtained by averaging the energy consumption of all nodes.

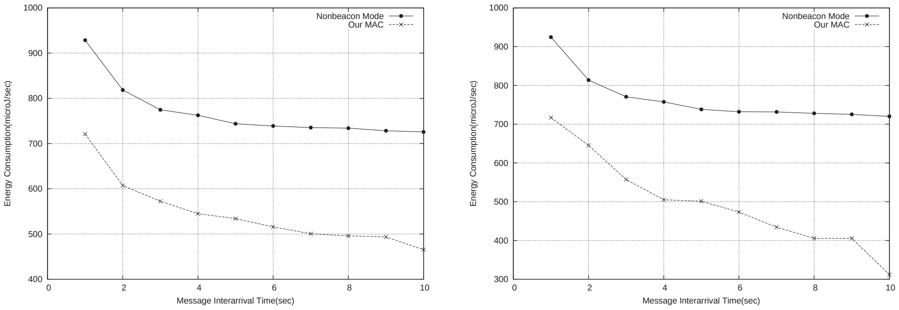
• **Power Efficiency** Power efficiency is defined as the throughput achieved per unit of energy consumed, where the throughput represents the number of successfully delivered packets.

$$PowerEfficiency = \frac{Throughput(Packets)}{EnergyConsumption(Joules)}$$

• **Number of Schedules and Schedule Convergence Latency** We observe the schedules of all nodes, and count the number of the schedules when we sample the network. The sampling interval is 10 seconds. As the number of the sampled schedules converges to 1, the convergence latency is also measured.

4.3 Numerical Results

Figure 3 shows the average energy consumption for the original nonbeacon-enabled network and our proposed MAC design under the chain and the random topologies. We first use the chain of 10 nodes for measurement. Figure 3 (a) indicates that the average energy consumption decreases as the message inter-arrival time increases. Moreover, the average energy consumption of the proposed MAC design is lower than the energy consumption in the idle state (Table 1). It indicates that we reduce the energy consumption of idle listening. In the random topology (Figure 3 (b)), the average energy consumption is still lower than that of the idle state. Since the latency of our MAC design is a little larger than the original one, it is tolerated in the wireless sensor network application.



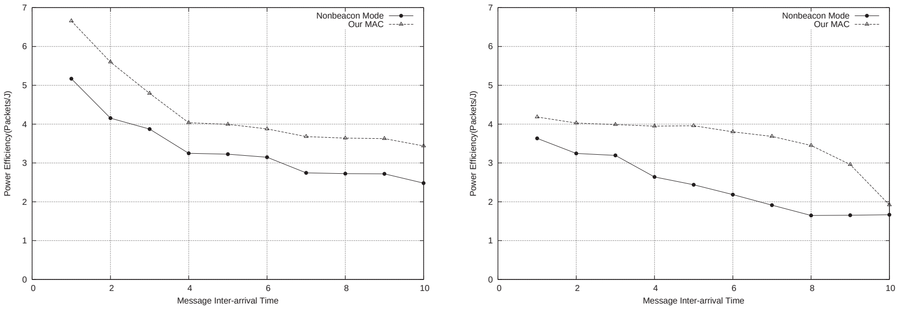
(a) Chain:10 nodes

(b) Random:25 nodes

Fig. 3. Average power consumption

The output measure of power efficiency is shown in Figure 4 for the two topologies. In the chain topology, our MAC design achieves higher power efficiency than original nonbeacon-enabled network (see Figure 4 (a)). On the other hand, in the random topology, the power efficiency of the proposed MAC design is a little better than the original one (see Figure 4 (b)). Since many nodes in the boundary are often in the sleeping state in the random topology when the message inter-arrival time increases, the latency will increase and the throughput will be lower. Therefore, the power efficiency of our MAC design will be close to that of the original one.

Figure 5 indicates the number of the schedules in the network. When the emergency traffic occurs, multiple schedules will be generated, and the convergence latency will increase. After a while, the number of schedules in the network will converge to a single schedule gradually, and the latency will be reduced. In Figure 5 (b), since the nodes in the random topology are more distributive, multiple schedules occurring will make the latency increase seriously. Therefore, our MAC design solves this problem to reduce the latency. Although the emergency

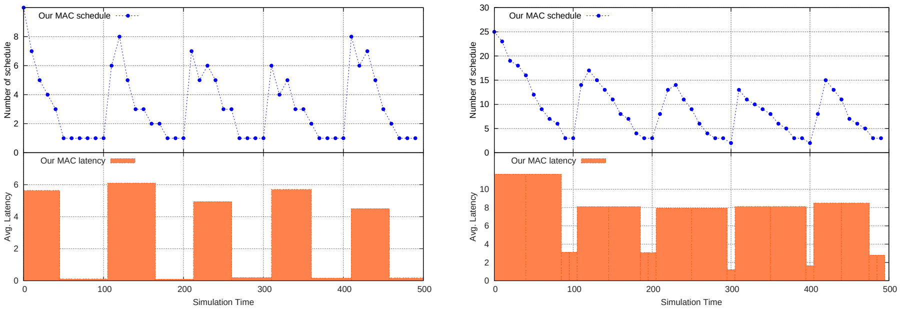


(a) Chain:10 nodes

(b) Random:25 nodes

Fig. 4. Power efficiency

event causes the traffic unstable, our proposed MAC design will make multiple schedules converge to one schedule to reduce the latency.



(a) chain:10 nodes

(b) Random:25 nodes

Fig. 5. Schedule Convergency

5 Conclusion

To improve the performance of the MAC protocol for IEEE 802.15.4-based wireless sensor networks in nonbeacon-enabled mode, this paper presents a new energy-efficient MAC design. Our MAC design adaptively determines the sleeping schedule of sensor nodes based on the network traffic load to achieve the balance of throughput and energy consumption. Our proposed MAC design consists of two phases: schedule exchange phase and schedule generation phase. In the schedule exchange phase, exchanging schedules is simple without extra overheads since the schedule parameters are piggybacked in DATA and ACK packets.

In schedule generation phase, nodes adaptively determine the sleeping schedule by using the received schedule parameters. The experimental results show that the proposed MAC design achieves sleeping schedule convergence as well as high energy efficiency.

References

1. M. Stemm and R. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," in *IEICE Transaction on Communication*, August 1997.
2. J. H. Wei Ye and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *INFOCOM*, June 2002.
3. J. H. Wei Ye and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," in *IEEE/ACM Transaction on Networking*, June 2004.
4. T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *SenSys*, November 2003.
5. S. R. Tao Zheng and V. Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *IPDPS*, 2005.
6. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Standard 802.11-1999 edition," 1999.
7. "802.15.4-2003 IEEE Standard for Information Technology-Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," 2003.
8. P. G. Ed Callaway and L. Hester, "Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks," in *IEEE Communications Magazine*, August 2002.
9. W. Y. Yuan Li and J. Heidemann, "Energy and Latency Control in Low Duty Cycle MAC Protocols," in *WCNC*, 2005.
10. "Crossbow Technology Inc.,MICAz wireless measurement system," in *http://www.zigbee.org*, June 2004.
11. "Moteiv Corporation, Telos (Rev B) Datasheet," in *http://www.moteiv.com*, December 2004.
12. "Intel iMote2," in *http://www.intel.com/research/exploratory/mote.htm*, February 2005.
13. N. F. Timmons and W. G. Scanlon, "Analysis of the Performance of IEEE 802.15.4 for Medical Sensor Body Area Networking," in *IEEE SECON 2004*, October 2004.
14. J. Gutierrez, "On the use of IEEE 802.15.4 to enable wireless sensor networks in building automation," in *Personal, Indoor and Mobile Radio Communications (PIMRC 2004)*, September 2004.
15. G. Lu, B. Krishnamachari, and C. S. Raghavendra., "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low-power wireless networks," in *Proceedings of the 23rd IEEE International Performance, Computing and Communications COnference (IPCCC 2004)*, April 2004.
16. J. F.Kurose and K. W. Ross, *Computer Networks.* Addison Wesley, 2003.
17. "The NS-2 Simulator," in *http://www.isi.edu/nsnam/ns/*.