논문 2010-6-19

무선 센서 네트워크에서 엿듣기 에너지 효율을 위한 기동 방식

Overhearing Energy Efficient Wakeup Schemes in Wireless Sensor Networks

이병용*, 문필주**

Byoung-Yong Lee*, Phil-Joo Moon**

요 약 무선 센서네트워크에서 센서 노드들에 대한 에너지 소비를 최소화하기 위한 연구가 최근에 활발히 진행되고 있다. 특히 각각의 센선 노드들은 제한적인 에너지로 인해서 관련이 없는 패킷들의 엿듣기 현상에 의한 에너지 소비를 줄이고자 한다. 본 논문에서는 이웃 센서 노드들이 서로 다른 wakeup time을 통해서, 엿듣기로 인한 에너지 소비를 줄이는 방식을 제안 한다. 주로 wakeup scheme에 대한 연구는 전송 지연을 줄이기 위한 연구에 초점을 두고 있다. 우리는 이러한 wakeup scheme을 통해서 엿듣기에 의한 에너지 소비를 줄이고자 한다. 본 논문의 실험 결과에서는 제안하는 wakeup schem이 센서 네트워크의 수명을 연장시켜주고 있음을 보여주고 있다.

Abstract Reducing the energy consumed by sensor node is a critical issue in wireless sensor networks. In particular, energy in each sensor node is too limited to waste on overhearing of packets that are not relevant. In this paper, we propose a wakeup scheme to reduce overhearing energy through the wakeup time difference between neighboring nodes. Other research papers on wakeup schemes usually focus on decreasing the latency. We propose a technique to reduce wasted energy for overhearing using the wakeup scheme. Simulation results indicate that our proposed wakeup scheme improves the sensor network lifetime.

Key Words : Wireless Sensor Network, Wakeup Scheme, Wakeup Time Schedule, Ubiquitous

Ⅰ. Introduction

Wireless sensor networks can be used in various physical world applications. But there are many limitations of sensors such as small amount of energy, limited CPU, and small size of memory. Among these challenges, we focus on reducing the energy consumption. Generally, a sensor node consumes its energy during processing, receiving, transmitting and overhearing. The energy for processing, receiving, and transmitting are necessary actions by the sensor node. However, the energy consumption for overhearing is wasted for a sensor network, since the packet is not used.

In this paper, we propose a wakeup scheme for reducing the overhearing energy consumption in wireless sensor networks. Wakeup scheduling techniques [2] usually focus on reducing latency. Also much research focuses on the MAC layer to reduce energy consumption that is used for collision,

*준회원, 텍사스 주립대학 컴퓨터학과
**정회원, 평택대학교 정보통신학과
접수일자 : 2010.10.18, 수정완료일자 : 2010.11.22
게재확정일자 2010.12.15

overhearing, control packet overhead, and idle listening [4]. We suggest a new way to reduce the overhearing through the wakeup time difference between neighboring nodes.

Ⅱ. Proposed wakeup scheme

1. Odd and Even Wakeup Scheduling

First we suggest the simple wakeup schedule named Odd and Even Wakeup Scheduling (OEWS), which can reduce the overhearing energy consumption as half of the sensor nodes at the same level wake up alternatively. The key idea is that sensor nodes having even id number and those having odd id number in the same level have different wakeup time schedules because of its different wakeup schedule. For example, in Fig.1 (a), we suppose that node 1 intends to send the data to node 8. At a specific even time, sensor node 1 in level 1 can send the data to nodes 2 and 4 having even number id. However node 3 having odd id number would not receive this data. But at a specific odd time, only node 3 will receive data sent from node 1. In the next step as shown in Fig.1 (b), after node 4 receives data from node 1, nodes in level 1 fall in sleep mode again and nodes in level 3 wake up. When node 4 intends to send the data to node 8, node 4 will send the data at the even wakeup timing. Node 7 follows the odd wakeup timing, and even if it does not receive data from node 4, node 7 will fall into sleep mode again.

Fig.2 shows the Odd and Even wakeup Scheduling. We use the synchronous wakeup schedule. First when level 1 transmits the data to level 2, the nodes in other levels are in sleep mode. Level 1 follows the schedule for a parent and level 2 follows the schedule for children. In here, level 1 and level 2 in Fig.2 correspond to node 1 and node 2, 3, 4 in Fig.1 respectively. Parent wakes up at every time slot but children nodes wake up alternatively. After the node in level 2 receives the data from a node in level 1, the node in level 2 changes the children wakeup time schedule to the parent wakeup time schedule. Then the nodes in level 1 enter sleep mode and the nodes in level 3 are in wakeup mode and follow the children wakeup time schedule.

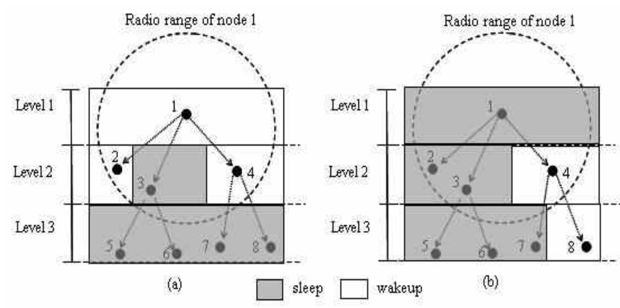

그림 1. Odd and Even 기동 스케줄 예제
Fig. 1. Odd and even wakeup scheduling Example

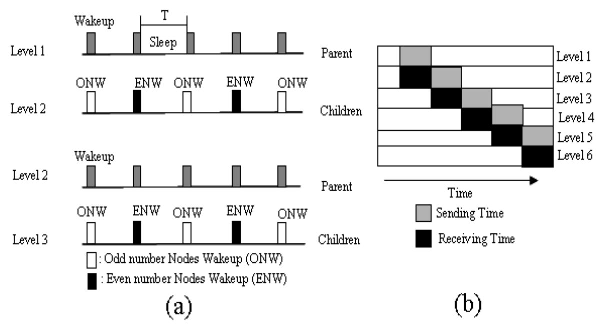

그림 2. Odd and Even 기동 타임 스케줄
Fig. 2. Odd and even wakeup time scheduling

Fig.2 (b) shows the entire wakeup scheduling based on [5]. When the base station makes the initial tree structure, it sets wakeup duration of each level in advance. Therefore each node knows what its level is and when is its wakeup time.

2. Individual Wakeup Scheduling

We propose another wakeup scheduling named Individual Wakeup Scheduling (IWS). Fig.4 shows the Individual Wakeup Scheduling. Each child node has a different wakeup time schedule than other children nodes in the same level. Therefore, at some specific time, only one child node wakes up and the other children nodes are in sleep mode. For example, in Fig.3 (a), when node 1 intends to send the data to node 4, node 1 waits until node 4 is in its wakeup time. The parent knows the wakeup schedule of each child node. When node 4 wakes up, other child nodes like node 2 and 3 are in sleep mode. Hence node 2 and node 3 would not receive the data from node 1 and do not overhear the packet. In Fig.3 (b), after node 4 receives the data from node 1, node 4 follows the parent wakeup schedule. In the parent level, even if other nodes such as node 2 and 3 are in level 2, only node 4 wakes up. In IWS, we can use the wakeup schedule for routing. In the case of the S-MAC protocol, control packets contain source and destination nodes. But IWS sends the data to the destination node through the wakeup time schedule. Therefore we can save the control packet size of source and destination. Another advantage is that IWS has zero overhearing energy consumption.

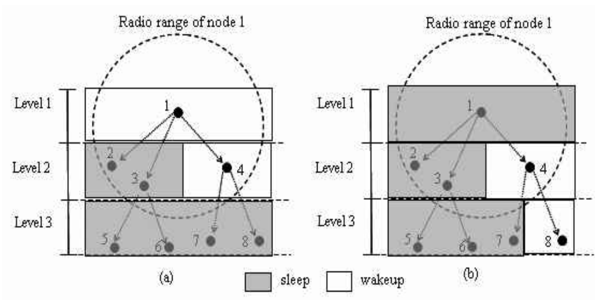

그림 3. 개별적 기동 스케줄 예제
Fig. 3. Individual wakeup scheduling Example

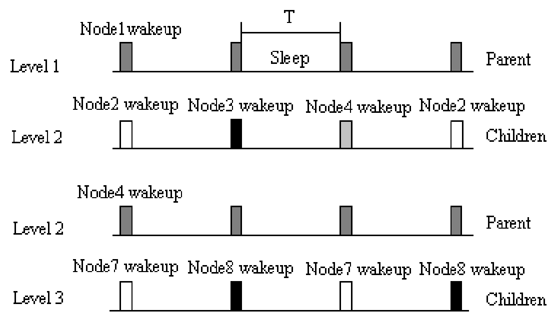

그림 4. 개별적 기동 타임 스케줄
Fig. 4. Individual wakeup time scheduling

In Fig.5, we present the algorithm for a parent node. In lines 1 – 2, parent node checks its wakeup schedule. If a parent still follows the children schedule, it changes to parent schedule. In lines 3 – 6, we store the children nodes of the current parent into array children[j]. We define G to be an undirected graph and V to be set of sensor nodes. In lines 7 – 11, if parent wakup time matches with a wakeup time of the target child node, parent node sends the data to the target child node. Otherwise the parent node waits until its wakeup time matches with the target child wakeup time. In lines 12-13, if parent level has time output, the parent node goes into sleep mode.

Parent Algorithm

```
Input : j=0, n_i = number of nodes,
1:  if Schedule = children_wakeup_schedule
2:      then change to parent_wakeup_schedule
3:  for n_i ∈ V[G]
4:      if n_i ∈ children of current parent node
5:          children[j] ← n_i
6:          j=j+1
7:  for target_child_id ∈ children[j]
8:      if target_child_id ≠ children[j]
9:          then wait()
10:     if target_child_id = children[j]
11:         then send (data)
12: if level_wakeup_time = 0
13:     then sleep (until next wakeup time)
```

그림 5. 부모노드 알고리즘
Fig. 5. Algorithm for the parent node

Children Algorithm

```
1:  if Schedule = parent_wakeup_schedule
2:      then change to children_wakeup_schedule
3:  find_parent ()
4:  if parent send the data
5:      then receive (data)
6:  if level_wakeup_time = 0
7:      then sleep (until next wakeup time)
```

그림 6. 자식노드 알고리즘
Fig. 6. Algorithm for the Children nodes

In Fig.6, we show the algorithm for children nodes. In lines 1 -2 they check theirs wakeup schedule. If children follow the parent schedule, it changes to children schedule. In line 3, each child tries to find the parent node. In lines 4 – 5, they wait for the data from the parent node. If specific child node receives the data from the parent node, it becomes a new parent.

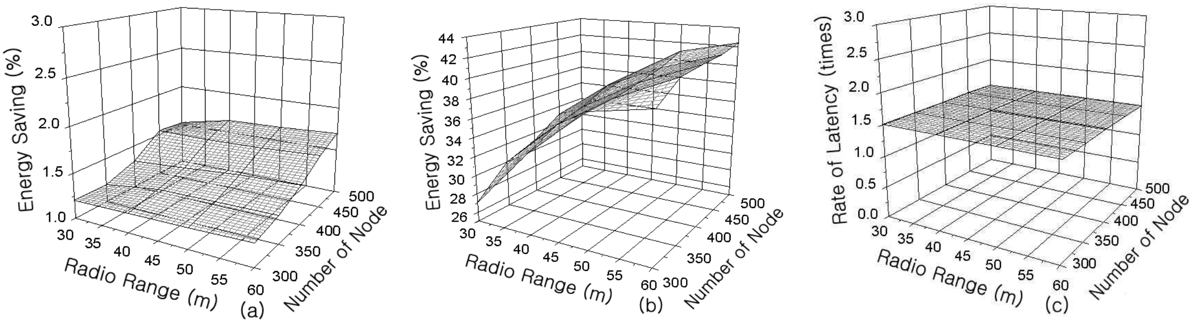

그림 7. OEWS 대 S-MAC (a) 총 에너지 소비량에서 에너지 절약율 (b) 엿듣기에서 에너지 절약율 (c)전송지연
Fig. 7. ODEW vs S-MAC (a) Energy saving in total energy consumption (b) Energy saving in overhearing (c) Latency

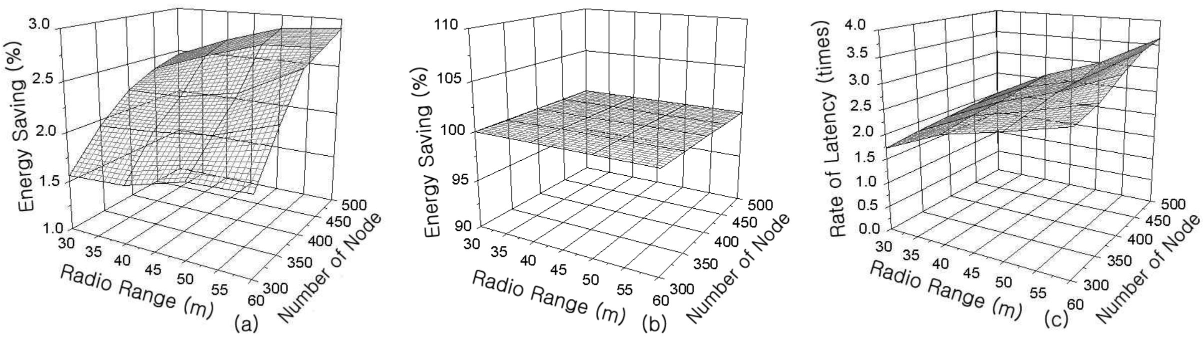

그림 8. IWS 대 S-MAC (a) 총 에너지 소비량에서 에너지 절약율 (b) 엿듣기에서 에너지 절약율 (c)전송지연
Fig. 8. IWS vs S-MAC (a) Energy saving in total energy consumption (b) Energy saving in ovehearing (c) Latency

III. Simulation

In this section, we present simulation results of OEWS and IWS. The purpose of this paper is to reduce the overhearing energy consumption. But there is a tradeoff between overhearing energy and data delivery latency. Therefore, we evaluated the efficiency of energy consumption and latency comparing our technique with S-MAC [4]. In S-MAC, one of the sources of wasted energy which they tried to reduce is the overhearing energy.

In the experiments, we randomly spread the homogeneous sensors in a 300 × 300m^2 sensor field area. All sensor nodes have the same fixed radio range and same energy. We use the Rings topology for initial routing tree structure[3] For measuring the energy consumption for transmitting, receiving, and overhearing data, we used the LEACH energy model[1].

1. Efficiency to reduce overhearing energy

In this experiment, we measured the rate of energy saving compared to the S-MAC. Even though S-MAC already reduces overhearing energy, the experiment result shows that OEWS and IWS reduce overhearing energy more than S-MAC. Fig.7 (a) and (b) show the energy saving results of OEWS comparing with S-MAC protocol. We increased the number of sensor nodes from 300 to 600 and the radio range from 30m to 60m. In Fig.7 (a), we compare energy saving rate in total energy consumption including transmitting, receiving and overhearing. This result shows that OEWS reduces more energy than S-MAC protocol. Fig.7 (b) shows the result when we only compare the overhearing energy with S-MAC protocol. We can save the overhearing energy up to 43% more than S-MAC.

Fig.8 (a) and (b) show the result of energy saving rate in IWS. IWS saves more energy than OEWS comparing with S-MAC protocol. Because we remove the overhearing energy consumption, in Fig.8 (b), the saving rate of overhearing energy is 100% comparing with S-MAC protocol's overhearing energy.

2. Effect on latency

In this section, we analyze the data latency between OEWS, IWS and S-MAC protocol. In Fig.2, when some events happen between the T period which is the duration of sleep, the node in level 1 waits until its next wakeup time. The probability of an event occurring between T periods is uniformly distributed. Therefore we represent the uniform distribution between A and B by $X \sim U[A,B]$ [2]. X is random delay time. A and B are the smallest delay time and the largest delay time respectively. Hence delay time of S-MAC is represented by the following:

$$X \sim U[\,(h-1)T\,,\, hT\,] \quad (1)$$

Therefore, average delay time is :

$$E(X) = (\,h - \frac{1}{2}\,) \quad (2)$$

In formulas (1) and (2), h is the number of hops. Also we can represent OEWS and IWS by the following respectively:

$$X \sim U[\,(h-1)T\,,\, 2hT\,] \quad (3)$$

$$X \sim U[\,(h-1)T\,,\, NhT\,] \quad (4)$$

Hence, average delay time of OEWS and IWS are:

$$E(X) = (\,\frac{3}{2}\,h - \frac{1}{2}\,)T \quad (5)$$

$$E(X) = (\,\frac{N+1}{2}\,h - \frac{1}{2}\,)T \quad (6)$$

In (4) and (6), N means the average number of sibling nodes. Fig.7 (c) shows the result of latency in OEWS comparing with S-MAC. We simulated with number of nodes from 300 to 600 and radio range from 30 to 60m. In this environment, latency of OEWS is 1.51 times the latency of S-MAC protocol. Fig.8 (c) shows the result of latency in IWS. In this case, as density of sensor field increases, latency is increased. With the same environment in Fig.8 (c), latency of IWS is between 1.72 and 3.64 times more than latency of S-MAC protocol.

Ⅳ. Conclusion

In this paper, we have proposed OEWS and IWS for reducing the overhearing energy consumption with different wakeup times. Even if advantage of OEWS and IWS is to reduce the overhearing energy consumption, there is a delay time because of a trade off between energy saving and delay time. Our simulation result also show OEWS and IWS have good performance. OEWS and IWS are more suitable for high density sensor network because overhearing energy cnsumption is high when nodes are having many neighborhood sensor nodes.

References

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks," Proc. 33rd Hawaii International Conf. System Sciences, Jan. 2000

[2] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," MobiHo'06, 2006, Florence, Italy.

[3] S. Nath, P. Gibbons, S. Seshan, and Z. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," In Proc. 2nd ACM SenSys, pages 250-262, 2004.

[4] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in INFOCOM 2002.

[5] S. Madden, M. Franklin, J. Hellerstein, W. Hong "TAG: a Tiny Aggregation Service for Ad-Hoc

Sensor Networks," In OSDI, 2002
[6] Gianpaolo Cugola, Matteo Migliavacca "A Context and Content-Based Routing Protocol for Mobile Sensor Networks," in Wireless Sensor Networks, 6th European Conference, EWSN 2009, Cork, Ireland, Februrary 11-13, 2009

저자 소개

이 병 용

• 1998년 2월 : 숭실대학교 대학원 전자계산학과 공학석사
• 2009년 8월 : 텍사스 주립대학교 대학원 컴퓨터공학박사
• 2006년 ~ 2009년 : 텍사스 주립대학교 Teaching Assistant
• 2009년 ~ 2010년 : LH U-Eco City Center 연구원
• 2010년 ~ 현재 : 가톨릭대학교 강사
<주관심분야 : 센서네트워크, 통신만보안, 데이터베이스>

문 필 주(정회원)

• 1988년 2월 : 숭실대학교 전자계산학과 공학사
• 1991년 8월 : 숭실대학교 대학원 컴퓨터과학과 공학석사
• 1988년 8월 : 숭실대학교 대학원 컴퓨터과학과 공학박사
• 1988년 ~ 2001년: 한국전자통신연구원 팀장(책임연구원)
• 2001년 ~ 현재 : 평택대학교 정보통신학과 교수
<주관심분야 : 가입자망기술, 인터네트워킹기술, 통신망 보안, 센서네트워크, 모바일 애플리케이션>