A Two Level TDMA Scheduling Protocol with Intra-cluster Coverage for Large Scale Wireless Sensor Network

Haigang Gong,[†] *and Ming Liu*[††]*,*

University of Electronic Science and Technology , College of Computer and Science, Sichuan, P. R. China

Summary

One of the key problem for Wireless Sensor Network (WSN) is the design of Medium Access Control(MAC) protocol, which medium access is the major consumer of sensor energy. TDMA-based MAC protocol is inherently collision free, and can rule out idle listening since nodes know when to transmit. Cluster-based TDMA protocol is more scalable than traditional TDMA protocol, but it introduces inter-cluster interference for which FDMA and CDMA are not good solutions due to their hardware complexity and high cost. In this paper, we present TLTS, a Two Lever TDMA Scheduling protocol, in which the first scheduling ensures that neighboring clusters collecting their member's data during different frames in order to avoid inter-cluster interference and the second scheduling schedules members of cluster sending their data to their respectively cluster heads during different slots to avoid intra-cluster interference. Simulation results show that TLTS performs better than HEED when node density turns higher. TLTS is more practical than HEED which uses CDMA code to avoid inter-cluster interference for deploying large scale WSN.

Key words:

wireless sensor networks; media access control ; TDMA; energy efficiency; Intra-cluster coverage

Introduction

Recent advances in wireless communications and microelectro-mechanical systems have motivated the development of extremely small, low-cost sensors that possess sensing, signal processing and wireless communication capabilities. Hundreds and thousands of these inexpensive sensors work together to build a wireless sensor network (WSN), which can be used to collect useful information (i.e. temperature, humidity) from a variety of environment. The collected data must be transmitted to remote base station (BS) for further processing. WSNs have been envisioned to have a wide range of applications in both military as well as civilian domains [1]-[3] such as battlefield surveillance, machine failure diagnosis, and chemical detection.

The main constraint of sensor nodes is their low finite battery energy, which limits the lifetime and the quality of the network. Since sensor nodes are often left unattended e.g., in hostile environments, which make it difficult or impossible to recharge or replace their batteries, the protocols running on sensor networks must consume energy efficiently in order to achieve a longer network lifetime. There have been a lot of energy efficient protocols proposed for WSNs [4]-[7], aiming to maximize the lifetime of the system under different circumstances.

Like other shared-medium networks, medium access control (MAC) is also one of the most important key techniques that ensure the successful operation of WSN. A MAC protocol decides when competing nodes may access the shared medium and tries to ensure that no two nodes are interfering with each other's transmissions. So, medium access is a major consumer of sensor energy and a lot of MAC protocols for WSNs have been studied in recent years [9]-[22]. MAC protocols could be categorized into two classes: schedule-based (TDMA, FDMA) and contention-based (802.11, S-MAC [4]). TDMA is an important schedule-based approach that controls the access to a single channel. TDMA provides collision-free transmission since a set of time slots are prearranged. An efficient TDMA schedule can save energy by allowing nodes to turn on the radio only during the scheduled transmission times of their neighbors, without wasting energy due to idle listening and collision, which are the two major sources of energy wastage [4]. For the inherently property of energy conserving, TDMA protocols have been recently attracted significant attention for many applications [9]-[15].

In this paper, we present TLTS, a cluster-based Two Level TDMA Scheduling protocol for large-scale wireless sensor network in which sensor nodes operate only on single frequency radio. TLTS aims to improve energy utilization efficiency by avoiding inter-cluster and intra-cluster transmission interference with two TDMA schedules. Simulation results show that TLTS has better performance.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 states the problem and presents system model. Section 4 describes our TLTS protocol in detail. Section 5 discusses the simulation results. Finally, Section 6 concludes the paper and presents future research directions.

Manuscript received February 25, 2006.
Manuscript revised February 28, 2006.

2. Related work

The major attractions of TDMA protocol are that it is inherently collision free and that idle listening can be ruled out since nodes know when to receive incoming data. *Stankovic* et al. discuss TDMA protocols in detail [24]. Earlier TDMA protocols are mostly centralized and access point knows global location information, which is poorly scalable. However, allocating time slots using distributed algorithm proves to be NP-hard [25].

The Self Stabilizing (SS-TDMA) protocol [10] uses a fixed schedule throughout the lifetime of the network. It operates on regular topologies like square and hexagonal grids. Nodes know their location and collision free schedule can be made according the location information. Author show that SS-TDMA can result in acceptable performance, but their constraints on the location of the nodes renders it impractical in many deployment scenarios.

TRAMA protocol [14] proposed by Rajendran et al. is to replicate the scheduling process over all nodes within the network. Nodes regularly broadcast information about traffic flows routed through them and the identities of their one-hop neighbors. This information is sufficient to determine a collision-free slot assignment by means of a distributed hash function that computes the winner of each slot based on the node identities and slot number. Although TRAMA achieves high channel utilization, it does so at the expense of considerable latency and high algorithmic complexity.

The approach taken by van Hoesel et al. in the LMAC protocol [15] is to assign nodes an unused slot number within a two-hop neighborhood to ensure collision-free transmission. Nodes organize time into slots, grouped into fixed-length frame. A slot consists of a traffic control section (12 bytes) and a fixed-length data section. When a node wants to send packet, it broadcasts a message in the control section containing the destination and length until its time slot comes around, and then immediately transmits the data. Nodes listening to the control header turn off their radio during the data part if they are not a receiver of the message. During each frame, message is just forwarded one hop toward the gateway until it reaches gateway. The drawback of LMAC is that nodes must always listen to the control sections of all slots in a frame even if slots are unused. In addition, LMAC is not suitable for large-scale network for that in large-scale network, the number of nodes' two-hop neighbors is too much which cause schedule collision.

Cluster-based TDMA protocols in [26]-[28] prove to be having good scalability. The common feature of these protocols is to partition the network into some clusters, in which cluster heads are responsible for scheduling their members. In LEACH [26], cluster heads broadcast a TDMA schedule packet within their cluster after clustering. Cluster members send the monitored data to their cluster heads that forward the data to the remote base station. HEED [27] is similar to LEACH in their TDMA scheduling. But cluster heads communicate with the base station in multi-hop way after they collect data from their members. Both LEACH and HEED operate on round and cluster heads are selected periodically. However, cluster-based TDMA protocols introduce inter-cluster transmission interference. Because clusters created by distributed clustering algorithm are often overlapped and several cluster heads may cover the same nodes. As shown in Fig. 1, shadowed node C is member of cluster head A and is also covered by cluster head B. While node D sends data to B, if node C transmits to A simultaneously, the reception of B will be interfered with the signal from C. And the packet sent by D is corrupted and energy dissipated for transmitting and receiving is consumed unnecessarily.

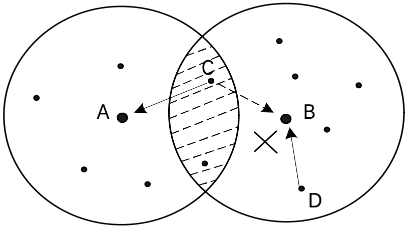

Fig. 1. Inter-cluster interference

From the discussion above, each TDMA protocol has its respective limitations. SS-TDMA operates on regular deployment and needs the nodes' location; TRAMA produces more protocol overhead for its high algorithmic complexity; LMAC requires the support of gateway node and has poorer scalability; Cluster-based TDMA protocols cause inter-cluster interference, thus transmission collision happens. Our TLTS protocol is a cluster-based TDMA protocol. The main idea of TLTS is that neighbouring cluster heads collect their member's data during different TDMA frame to avoid inter-cluster interference. As in Fig. 1, cluster head A gathers the data during the first TDMA frame and cluster head B collects the data during the second TDMA frame, avoiding the transmission collision that appears when A and B gather the data at the same time.

3. Problem statement and system model

3.1 Problem statement

Inter-cluster interference must be addressed for cluster-based TDMA protocol. The approach taken by Sonia et al. [26] is the combination of TDMA and FDMA. Cluster members send their data to cluster heads according to TDMA schedule and neighboring clusters use different frequency (FDMA) to transmit so as to avoid inter-cluster interference. LEACH and HEED then exploits Direct Sequence Spread Spectrum (DS-SS) code, a CDMA-like scheme, to avoid inter-cluster interference. Each cluster uses different weak-correlated spread spectrum code. Cluster heads can filter the code belongs to other clusters among multiple received signals and decode the signal sent from their members.

However, FDMA is not applicable in the context of sensor network since sensor nodes are often restricted to transmit only on one frequency once the frequency is set before deployment. For instance, MICA motes of UCB [30], the most popular sensor nodes equipped with one radio, operate on a fixed frequency during the lifetime of network. CDMA requires expensive operations for encoding or decoding a message and is not preferred for sensor network that lack the special hardware required for CDMA and that have limited computing power. Either FDMA or CDMA needs to extend the function of sensor's hardware, which complicates the hardware design and increases node's hardware costs. The increase of hardware cost is undoubtedly uneconomical especially for large-scale wireless sensor network. Therefore, the problem is how to avoid inter-cluster interference and improve energy utilization efficiency with a reasonable TDMA schedule in a large-scale wireless sensor network in which nodes operates only on a single frequency.

3.2 Network model

Assume that N nodes are dispersed in a square M×M field randomly, and the follow assumptions hold:

1) Node synchronization is available which is necessary for TDMA based protocols.
2) The only base station sits at a fixed location outside the field.
3) All nodes have same capabilities.

Nodes are left unattended after deployment and nodes are stationary.

3.3 Wireless radio model

We use the same radio model as HEED [27] for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics. To transmit a *k*-bit message a distance *d*, the radio expends energy as (1).

$$E_{Tx} = \begin{cases} k * E_{elec} + k * e_{fs} d^2, & d < d_0 \\ k * E_{elec} + k * e_{amp} d^4, & d \geq d_0 \end{cases} \quad (1)$$

To receive this message, the radio expends energy as (2):

$$E_{Rx} = k * E_{elec} \quad (2)$$

E_{elec}, the electronics energy, depends on factors such as the digital coding, modulation, and filtering of the signal before it is sent to the transmit amplifier. And the amplifier energy, $e_{fs}d^2$ or $e_{amp}d^4$, depends on the distance to the receiver.

4. TLTS protocol design

TLTS is a cluster-based TDMA scheduling protocol that is orthogonal to cluster-based protocols such as LEACH and HEED. We use HEED protocol to group clusters and gather data. As shown in Fig. 2, a round in HEED includes two phases: cluster formation and data gathering. After clustering, cluster heads broadcast a simple TDMA schedule to tell their members when to send their data, then following K TDMA frame. Cluster heads gather member's data in a TDMA frame that called a data cycle that means all member nodes send their sensing data to their cluster heads, thus a round includes K data cycle.

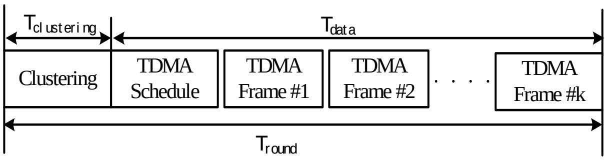

Fig.2. Frame structure of HEED

TLTS substitutes TDMA schedule scheme of HEED after cluster formation as described in Fig. 3. Data gathering consists of a two level TDMA scheduling (TLTS) and K super-frames. A super-frame corresponding to a data cycle in HEED and consists of N_f frames, which each frame comprises m time slots. The goal of TLTS is to assign different frame to neighbouring clusters so as to avoid inter-cluster interference and to allocate slots to cluster members to avoid collision within clusters. TLTS consists of two phases: inter-cluster TDMA schedule and intra-cluster TDMA schedule.

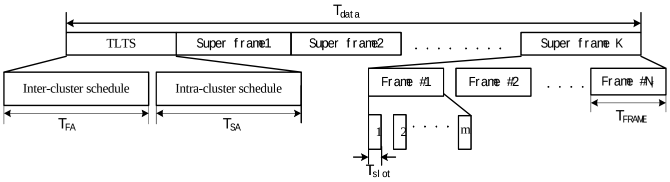

Fig. 3. Frame structure of TLTS

4.1 Inter-cluster TDMA schedule

Inter-cluster TDMA schedule assigns different frames to neighboring clusters so that neighboring cluster heads could gather their members' data during different time which avoids the inter-cluster interference. Frame assignment is analogous to coloring problem in graph theory [31] and frequency allocation in cellular communication [26]. Define frame assignment problem as follows:

Given a connected graph $G=(V,E)$,V is set of cluster heads , $E=\{(u,v)\,|\,u,v\in V\}$, edge *(u , v)* means *u* is adjacent to *v*. If $\forall(u,v)\in E$, then $u.frame\#\neq v.frame\#$ 。

FA (Frame Assignment) algorithm is described in Fig. 5. Each cluster head maintains CH_{nb}, a set of its one-hop neighbouring cluster heads. n_i is the number of cluster heads that haven't been assigned frame and t should be long enough to receive messages from neighbours. In step *i*, a cluster head (CH) broadcasts a RESERVATION message with probability 1/ n_i+1 and becomes competing CH (line 5~10). RESERVATION message includes the CH's ID and an unused frame number (frame #). After time t, if it doesn't receive any other RESERVATION message, meaning no neighbours subscribe frame number in this step, it will broadcast a CONFIRM message to confirm its reservation (line 17~20). After receiving this CONFIRM message, its neighbours should label the reserved frame number used and delete the CH from its CH_{nb} (line 23). If it receives other RESERVATION message, meaning its neighbours subscribe frame number at the same time, collision happens if they reserve the same frame number. The competing CH that has max ID would be the winner of that frame number and the losers will book the frame number in the next step until all cluster heads have got their frame number.

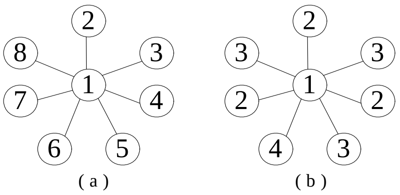

Fig. 4. Two frame assignment schemes

N_f influences the performance of TLTS. A large N_f introduces long delay and a small N_f leads to frame assignment collision that neighbouring cluster heads get same frame number. N_f =*D*+1 meets the requirement to assign different frame number to neighbouring clusters [31], where *D* is the max number of neighbouring cluster heads. However, N_f can be far fewer than *D*+1 for that cluster heads that are not adjacent can use the same frame number. Fig. 4 shows two frame assignment schemes (*D*=7). Fig. 4a assigns 8 frames to 8 neighbouring clusters. In fact, 4 frames is enough as shown in Fig. 4b.

```
1. GotFrame = FALSE
2. REPEAT:
3.     n = |$CH_{nb}$|; contending = FALSE;
4.     p = 1 / (n+1)
5.     IF (random (0,1) <= p)
6.         my_frame# = SELECT frame# different
                                    from my neighbors
7.         RESERVATION.ID = my_ID;
8.         RESERVATION. frame# = my_frame#
9.         BROADCAST (RESERVATION)
10.        contending = TRUE

11.    WAIT t

12.    IF ( contending )
13.        IF (received RESERVATION)
14.            IF (my_ID > RESERVATION. ID AND
                   my_frame# = RESERVATION. frame#)
15.                contending = FALSE
16.            ELSE
17.                CONFIRM. ID = my_ID ;
18.                CONFIRM.frame# = my_frame#
19.                BROADCAST( CONFIRM )
20.                GotFrame = TRUE

21.    WAIT t

22.    IF( received CONFIRM)
23.        $CH_{nb}$ = $CH_{nb}$ - CONFIRM.ID

23. UNTIL (GotFrame)
```

Fig.5. Frame assignment algorithm

4.2 Intra-cluster schedule

Intra-cluster TDMA schedule is similar to the TDMA schedule in HEED but with a more field of frame number in the schedule packet. Cluster heads just broadcasts TDMA schedule packet containing slot assignment information as in Fig. 6. The field of frame number tells cluster members during which frame their cluster heads collect the data and the field of slot tells them during which slot of that frame they should be active and send data to cluster heads.

Frame number	Member 1's ID	Slot 1	Member 2's ID	Slot 2	. . .	Member m's ID	Slot m

Fig. 6. Format of intra-cluster schedule packet

The number of cluster members decides the length of schedule packet. Assume N nodes are randomly deployed in a M×M field A. Then the average number of cluster members is $N\cdot\pi R^2/M^2$, where R is cluster radius. The higher node density, the more cluster members and the longer the length of schedule packet that leads to more energy consumption for transmitting and receiving the packet. More over, the more cluster members, the longer frame time, which leads to longer delay. In fact, sensors are usually deployed densely (high up to 20nodes /m^3 [2]). In such a high-density network with energy-constrained sensors, it is neither necessary nor desirable to have all nodes work at the same time. In [33], authors think it is hard to guarantee full coverage for a given randomly deployment area even if all sensors are on-duty and small sensing holes are not likely to influence the effectiveness of sensor networks and are acceptable for most application scenarios. So we introduce the idea of intra-cluster coverage discussed in our previous work [8]. Intra-cluster coverage means that only a subset of cluster members should be active if these nodes cover the most of sensing field. Based on our previous work [8], cluster heads randomly choose m nodes according to (3).

$$P_{cover} = \sum_{i=1}^{m} C_m^i \left(\frac{r}{R}\right)^{2i} \left(1-\frac{r^2}{R^2}\right)^{m-i} \tag{3}$$

where P$_{cover}$ is the expected coverage ratio of sensing field determined by applications, and r is sensing radius. After cluster heads choosing active node, they broadcast the schedule packet. Cluster members go to be active if they find their *ID* in the packet. Otherwise, they become to be asleep. For example, distributing 2000 nodes in a 200× 200m^2 field, r = 12m, R = 30m, then the average number of cluster members is 120 or so. With intra-cluster coverage, if P$_{cover}$ = 99% which means 99% of sensing field is expected to be monitored, 27 members should be active in each cluster. If P$_{cover}$ =95%, only 16 nodes should be active.

Therefore, using intra-cluster coverage has two advantages. The first is to preserve energy consumption in each round by turning redundant nodes' radio off so that network lifetime is prolonged. The second is to reduce TDMA schedule overhead when node density is high enough for that the length of schedule packet is invariable. In contrast, the length of schedule packet in HEED increases while node density goes higher.

4.3 FA algorithm analysis

Lemma 1: FA algorithm terminates in O (*D*) steps, where *D* is the max number of neighboring cluster heads.

Proof. Assume FA algorithm terminates in L steps. At the beginning, a cluster head has n_1 neighboring cluster heads to compete for frame number. After L-1 steps, n$_1$ cluster heads should reserve their frame number successfully. Then the last cluster head gets its frame number in the last step.

In step i, n$_i$+1 cluster heads compete for frame number. The probability to subscribe the frame number is $p_i = 1/(n_i+1)$, where n_1 is the number of neighbors that have not got frame number in the previous *i*-1 steps. Then the probability of only one cluster head reserving frame number successfully is

$$P_i = p_i(1-p_i)^{n_i} \tag{4}.$$

The number of cluster heads reserving frame number successfully is $N_i \ge (n_i+1)\cdot P_i$ 。 After L-1 steps, the number of cluster heads reserving frame number successfully is

$$n_1 = \sum_{i=1}^{L-1} N_i \ge \sum_{i=1}^{L-1}(n_i+1)\cdot P_i = \sum_{i=1}^{L-1}\left(1-\frac{1}{d_i+1}\right)^{n_i} \tag{5}$$

Note that $(1-\frac{1}{n_i+1})^{n_i} \ge e^{-\frac{n_i}{n_i+1}} \ge e^{-1}$, then

$$n_1 \ge (L-1)\cdot e^{-1} \tag{6}$$

Thus we have

$$L \le n_1 \cdot e + 1 \le D \cdot e + 1 \tag{7}$$

In the worst case, algorithm will terminate in $D \cdot e + 1$, i.e. O (*D*)。

Table I. Simulation parameters

	Parameters	**Value**
common parameters	Monitoring Area(M×M)	200×200 m^2
	Number of nodes (*N*)	200~2000
	Sensing radius (*r*)	12 m
	Cluster radius (*R*)	30 m
	Distance threshold (d_0)	75 m
	Data packet	100 bytes
	E_{elec}	50nJ/bit
	ε_{fs}	10nJ/bit/m^2
	ε_{amp}	0.0013pJ/bit/m^4
	K	30
	Slot assignment time(T_{SA})	5ms
	Clustering time($T_{clustering}$)	300ms
TLTS	P_{cover}	95%
	m	16
	t	1ms
	Slot time (T_{slot})	8ms
	Frame time(T_{FRAME})	128ms
	Frame assignment time(T_{FA})	50ms
HEED	Slot time (T_{slot})	8ms
	Frame time (T_{FRAME})	Varies with the number of cluster members

5. Performance evaluation

5.1 Simulation parameters

To test the performance of TLTS, we ran simulations using ns-2. The simulation parameters are listed in Table I. Data transmission rate is set to 115.2 kps, the same as that in TR1000 [32] transceiver equipment on MICA motes [30] when using ASK modulation. Then it spends 7ms or so transmitting 100 bytes data and T_{slot} is set to 8ms. The P_{cover} is set to 95% and 16 active nodes (m=16) are enough to ensure 95% intra-cluster coverage and frame time is m $\times T_{slot}$=128ms. To HEED, frame time is variable with the number of cluster members.

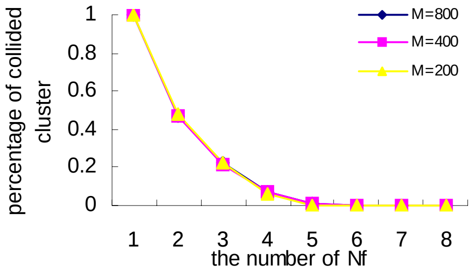

Fig. 7. The number of collided clusters vs. N_f

5.2 Setting of N_f

Fig. 7 plots the relationship between the number of clusters that have been assigned same frame number and N_f. The less the N_f, the more opportunity that same frame number is assigned to neighbouring clusters, which increases the number of clusters that have same frame number and leads to inter-cluster interference severely.

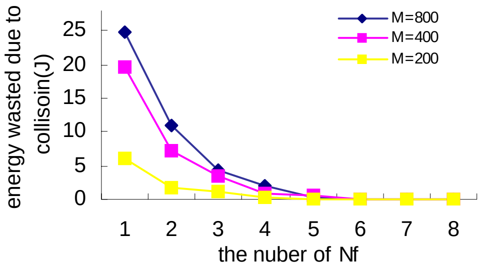

Fig. 8. Energy wastage due to collision vs. N_f

Fig. 8 shows the relationship between the energy wastage due to frame number collision and N_f under different monitoring area. The larger the area is, the more the number of clusters, which causes more inter-cluster collision and energy wastage. With the increasing of N_f, the probability of assigning different frame number to neighbouring clusters increases too. As seen from the figures, almost all neighbouring clusters can get different frame number when N_f is greater than 5. Consequently, we set N_f to 6 in the following simulations and the following results are sampled when simulation runs 1000s.

5.3 Simulation results

Fig. 9 shows the energy consumption for FA algorithm under different node density deployment. In a 200×200 monitoring area, the energy overhead of FA algorithm is just about 0.012J. It is 0.03% of total energy consumption and could be negligible.

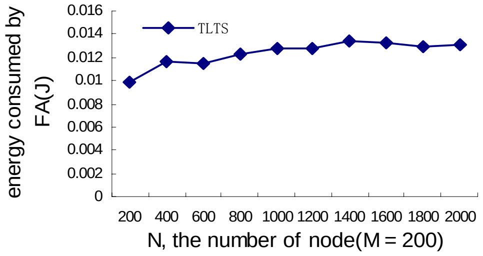

Fig. 9. Energy dissipated for FA

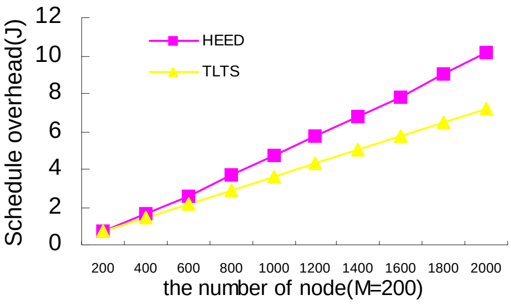

Fig. 10. TDMA schedule overhead

Fig. 10 describes TDMA schedule overhead of two protocols under different node density deployment in 200 ×200 monitoring area. TDMA schedule overhead is the energy dissipated for broadcasting TDMA schedule packet and receiving this packet by cluster members. For TLTS, it also includes the energy dissipated for frame assignment algorithm. The schedule overhead of HEED is much more than that of TLTS with the increasing of node density. For example, the schedule overhead of HEED is about twice than that of TLTS when node number is 2000. The reason is that the length of TDMA schedule packet of HEED increases with the increasing node numbers, which consumes more energy for broadcasting and receiving the packet. However, the length of schedule packet of TLTS doesn't vary with the number of nodes but with the parameter *m*, which is predefined before network setup. Therefore, the schedule over head of TLTS is less than that of HEED.

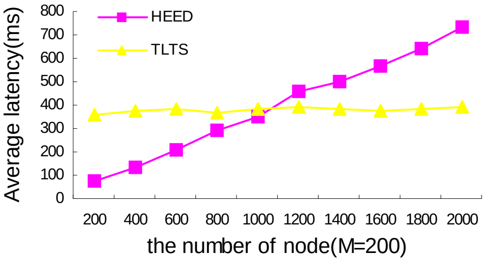

Fig. 11. Average packet latency

Fig. 11 shows the average packet latency under different node density deployment. The average packet latency means the average delay of cluster heads receive the data packet from their cluster members. Theoretically, the average packet delay of TLTS is half of a super-frame time that is N_f frame time and the average packet delay of HEED is half of a frame time. Similarly, the average delay of HEED increases linearly with the increasing node density and that of TLTS fluctuates a little e. This is because frame time of TLTS is 16 time slots all the time and frame time of HEED enlarges when the number of cluster members increases. When node density is low, the frame time of HEED is greater than the super-frame time of TLTS so that the average delay of HEED is less than that of TLTS. When node density goes higher than 0.03nodes/m^2, the average latency of HEED turns greater than that of TLTS because the frame time of HEED is larger than the super-frame time of TLTS under the high density deployment.

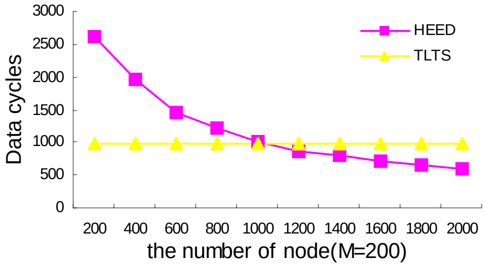

Fig. 12. Data cycles in 1000s

Fig. 12 shows the times of data cycles in 1000 simulation seconds. Due to the smaller frame time, data cycles of HEED are more than that of TLTS when there are small node numbers. When node density increases, the frame time of HEED enlarges and its data cycles are less than TLTS' s. Data cycles of TLTS is about 60% much more than that of HEED when node number is 2000.

6. Conclusion

In this paper, we present TLTS, a cluster-based two level TDMA scheduling protocol for large-scale wireless sensor network in which sensor nodes operates only on single frequency radio. TLTS improves energy utilization efficiency by avoiding inter-cluster and intra-cluster transmission interference with two TDMA schedules. In addition, TLTS uses intra-cluster coverage to reduce the packet delay. Simulation results show that TLTS has better performance than HEED when node density is higher than 0.03nodes/m^2. And the hardware cost of sensor running TLTS may be much less than that of sensors running HEED that needs special hardware. So, TLTS is more suitable for large-scale wireless sensor network than HEED.

Reference

[1] D. Estrin, R. Govindan, J. Heidemann and S. Kumar. Next Century C Challenges: Scalable Coordination in Sensor Networks. In *Proceeding of ACM MobiCom'99*, Washington, August 1999.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E.Cayirci. Wireless Sensor Networks: A survey. *Computer Networks*, March 2002.

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the World with Wireless Sensor Networks. *In International Conference on Acoustics, Speech, and Signal Processing (ICASSP2001)*, Salt Lake City, Utah, May 2001.

[4] W. Ye, J. Heidenmann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE INFOCOM*, New York, NY, June 2002.

[5] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for Self-Organization of a Wireless Sensor Network. *IEEE Personal Comm. Mag.*, vol. 7, no. 5, Oct. 2000

[6] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks. *ACM Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[7] V. Kawadia and P. R. Kumar. Power Control and Clustering in Ad Hoc Networks. In *Proceedings of IEEE INFOCOM*, April 2003

[8] LIU Ming, et al. .A Distributed Energy-Efficient data Gathering and aggregation Protocol for Wireless sensor networks.. to appear in *Journal of Software.*

[9] K. Arisha, et al. Energy-aware TDMA-based MAC for sensor networks. In IEEE Workshop on Integrated Management of Power Aware Communications, Computing and NeTworking(IMPACCT 2002), New York City, NY, May 2002

[10] S. Kulkarni, et al. TDMA service for sensor networks. In *24th Int. Conf. on Distributed Computing Systems*

(ICDCS04), ADSN workshop, pp. 604-609, Tokyo, Japan, March 2004.

[11] J. Li, et al. A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks. In *3rd Int.Symp. on Information Processing in Sensor Netwokrds(IPSN04)*, pp 55-60, Berkley, CA, April 2004.

[12] G. Pei and C. Chien. Lower power TDMA in large wireless sensor networks. In *Military Communications Conference (MILCOM 2001)*, volume 1, pp. 347-351, Vienna, VA, Oct. 2001

[13] L. van Hoesel, et al. Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs. In *IEEE VTC 2004 Spring*, Milan, Italy, May 2004.

[14] V. Rajendran, et al. Energy-efficient, collision-free medium access control for wireless sensor networks. In *1st ACM Conf. on Embedded Netwoked Sensor Systems (SenSys 2003)*, pp. 181-192, Los Angeles, CA, Nov. 2003

[15] L. van Hoesel, et al. A lightweight medium access protocol (LMAC) for wireless sensor networks. In *1st Int. Workshop on Networked Sensing Systems(INSS 2004)*, Tokyo, Japan, June 2004

[16] T. van Dam, et al. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *1st ACM Conf. on Embedded Networked Sensor Systems (Sensys 2003)*, pp. 171-180, Los Angeles, CA, Nov. 2003

[17] A. El-Hoiydi, et al. WiseMAC, an ultra low power MAC protocol for the WiseNet wireless sensor network. In *1st ACM Conf. on Embedded Networked Sensor Systems (Sensys 2003)*, Los Angeles, CA, Nov. 2003

[18] C. Guo, L. Zhong, and J. Rabaey. Low power distributed MAC for ad hoc sensor networks. In *IEEE GlobeCom*, San Antonio, AZ, November 2001.

[19] R. Jurdak, C. Lopes, and P. Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *IEEE Communications Surveys and Tutorials*,6(1), 2004.

[20] F. Liu, K. Xing, X. Cheng, and S. Rotenstrech. Energy-efficient MAC layer protocols in ad hoc networks. In M. Cardei, I. Cardei, and D.-Z. Du, editors, *Resource Management in Wireless Networking*. Kluwer Academic Publishers, 2004.

[21] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In *Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, NM, April 2004.

[22] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *ACM/IEEE Transactions on Networking*, *12* (3), pp. 493-506, June, 2004

[23] K. Sohrabi, et al. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communication Magazine*, vol. 7, no. 5, pp. 16-27, Oct. 2000

[24] J. A. Stankovic, et al. Real-time communication and coordination in embedded sensor networks. In *Proc. of IEEE*, vol. 91, no. 7 ,, Jul. 2003

[25] E. Arikan. Some complexity results about packet radio networks. *IEEE Trans. on Inform. Theory*, vol. IT-30, pp. 681–685,1984.

[26] S. Waharte, et al. Performance comparison of distributed frequency assignment algorithms for wireless sensor networks. in Proceedings of the IFIP International Conference on Network Control and Engineering (Net-Con'2004), Palma de Mallorca, Spain, November 2004

[27] O. Yonis, et al. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, volume 3, issue 4, Oct-Dec, 2004

[28] W. R. Heinzelman, et al. An Application -Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, Oct. 2002

[29] I. Rhee, et al. Distributed scalable TDMA Scheduling Algorithm. *NCSU Technical Report 2004*, Department of Computer Science, North Carolina State University.

[30] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. Pister. System architecture directions for network sensors. In Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), November 2000.

[31] D. A. Grable, et al. Fast distributed algorithms for brooks-vizing colourings. In *ACM-SIAM Symposium on Discrete Algorithms*, 1998, pp. 473-480

[32] RF Monolithics. http:// www. rfm. com/, *ASH Transceiver TR1000 Data Sheet*

[33] Y. Gao, K. Wu, and F. Li, “Analysis on the redundancy of wireless sensor networks,” Proceedings of WSNA 03, September 2003, San Diego, CA.

Haigang Gong received the M.S. degree in University of Electronic Science and Technology of China in 2003 and the Ph. D. degree in Nanjing University in 2006, respectively. His research interests include wireless sensor network, mobile computing, p2p computing, grid computing.

Ming Liu received the M.S. degree in University of Electronic Science and Technology of China in 2003. Currently he is the Ph. D. candindate in Nanjing University in 2006, respectively. His research interests include wireless sensor network, mobile computing, p2p computing, grid computing.