PAPER *Special Section on Emerging Technologies for Practical Ubiquitous and Sensor Networks*

Cross-Layer Design for Low-Power Wireless Sensor Node Using Wave Clock

Takashi TAKEUCHI[†a)], ***Student Member***, **Yu OTAKE**[†], **Masumi ICHIEN**[††], ***Nonmembers***, **Akihiro GION**[††], ***Member***, **Hiroshi KAWAGUCHI**[†], ***Nonmember***, **Chikara OHTA**[†], ***and*** **Masahiko YOSHIMOTO**[†], ***Members***

SUMMARY We propose Isochronous-MAC (I-MAC) using the Long-Wave Standard Time Code (so called "wave clock"), and introduce cross-layer design for a low-power wireless sensor node with I-MAC. I-MAC has a periodic wakeup time synchronized with the actual time, and thus we take the wave clock. However, a frequency of a crystal oscillator varies along with temperature, which incurs a time difference among nodes. We present a time correction algorithm to address this problem, and shorten the time difference. Thereby, the preamble length in I-MAC can be minimized, which saves communication power. For further power reduction, a low-power crystal oscillator is also proposed, as a physical-layer design. We implemented I-MAC on an off-the-shelf sensor node to estimate the power saving, and verified that the proposed cross-layer design reduces 81% of the total power, compared to Low Power Listening.

key words: *cross-layer design, crystal oscillator, long-wave standard time code, low power listening, MAC*

1. Introduction

A wireless sensor network is comprised of many wireless sensor nodes, each of which is driven by a small battery. The sensor nodes obtain environmental information and send it to a base station with a multi-hoping scheme. For various applications, the wireless sensor network is useful. However, changing batteries on thousands of sensor nodes would be a considerable burden. Thus, it is virtually impossible. Sensor nodes must be low power to maximize a total available time in a whole network system.

Of the power consumed by all sensor nodes, the greatest fraction is the power used for wireless communications. An effective way to reduce the energy is shorten the idle listening, in which the receiver is activated, even when no packet is received. To reduce the power of this idle listening, a type of MAC named Cycled Receiver MAC, which includes S-MAC [1], Low Power Listening (LPL) [2] and WiseMAC [3], has been developed. Using Cycled Receiver MAC, each node enters a receiving mode only during a specific wakeup duration time that occurs in every wakeup period. Reducing the duty cycle ratio, the power used for idle listening is also decreasing. In general, with a cycled receiver MAC, the longer the wakeup period, the longer the delay time for connection establishment. Therefore, under the condition of the same duty cycle ratio, the shorter wakeup period, the more advantageous it can be in terms of the delayed time.

The wakeup duration time of S-MAC is normally set as 115 ms. On the other hand, for LPL or WiseMAC, it is less than 50 μs, which is sufficient duration to monitor the channel usage. Consequently, if the same duty cycle ratio is given, the wakeup period of S-MAC is larger; it is less advantageous than the other two methods because of its longer delay time.

With LPL, the length of a preamble is set to the wakeup period. Consequently, the longer wakeup period causes the more power consumption of preamble transmission; that requirement conflicts with our goal to reduce the power consumption that is attributable to idle listening.

Another method, WiseMAC, is an updated version of LPL. Using WiseMAC, a receiving node makes its corresponding transmitter learn its future wakeup schedule by including the next wakeup timing in the ACK data. Therefore the transmitting node can shorten the preamble length for the next transmission to the receiver based on the knowledge of the schedule. However, a clock drift which is caused by temperature variation in a node increases the inaccuracy of this learning mechanism gradually with time. Therefore, this learning process of the wakeup schedule would not work as efficiently as expected unless some frequent communications have been established. To our knowledge, only the effect of the downlink from an access point to a sensor node seems to have been examined with WiseMAC. It has not been well evaluated in multi-hop communications.

In this paper, we propose a low-power MAC and cross-layer design with it. Our proposed Isochronous-MAC (I-MAC) [4] is based on LPL that has a periodic wakeup time. I-MAC also has a periodic wakeup time, but it is synchronized on each node with the actual time, using the Long-Wave Standard Time Code (so called "wave clock") [7]. Since a sender can predicts a next wakeup time of an intended receiver with high accuracy, we can minimize the duration of a preamble on the sender. As well as on the receiver side, the receiving time for the preamble can be reduced thanks to the short duration of the preamble. The next section briefly mentions I-MAC. Refer to [4] for more detail on I-MAC.

In I-MAC, the time on sensor nodes are matched to the actual time using the Long-Wave Standard Time Code, for instance, at intervals of an hour. One session for synchro-

Manuscript received February 28, 2008.
Manuscript revised June 3, 2008.

[†]The authors are with Department of Computer Science and Systems Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
[††]The authors are with Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.
a) E-mail: take@cs28.cs.kobe-u.ac.jp

DOI: 10.1093/ietcom/e91–b.11.3480

Copyright © 2008 The Institute of Electronics, Information and Communication Engineers

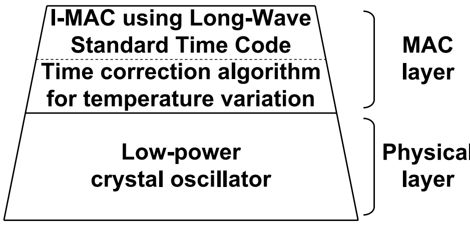

Fig. 1 Cross-layer design for I-MAC.

nization takes a few minutes. Between the sessions, a timer using a crystal oscillator internally keeps the time on a sensor node. However, in reality, the oscillation frequency of the crystal oscillator varies along with temperature, which causes a time difference among nodes. We will describe a time correction algorithm for the temperature variation in Sect. 3. The time correction algorithm minimizes the preamble length in I-MAC.

To verify the effectiveness of I-MAC, we modeled the power characteristics of I-MAC, and implemented I-MAC on an off-the-shelf sensor node as a prototype. Section 4 derives the power model from the measured value of the prototype.

In Sect. 5, we will introduce a low-power crystal oscillator for I-MAC, as a physical-layer design. Since the internal timer has to count the periodic wakeup time using the crystal oscillator, we cannot stop the timer in operation. Therefore, the power of the crystal oscillator is reflected on an idle power. The design of the low-power crystal oscillator is important.

Figure 1 illustrates the scope of this paper. I-MAC has the low-power features using both the short preamble length in the MAC layer and the low-power crystal oscillator in the physical-layer. In other words, the implementation of I-MAC is achieved by the cross-layer design. The short preamble is realized by the time correction algorithm.

Section 6 evaluates the total power in our proposed cross-layer design using I-MAC, the time correction algorithm, and low-power crystal oscillator.

Finally, we conclude this paper in Sect. 7.

2. Isochronous-MAC (I-MAC)

Figure 2(a) shows a manner in which a packet is sent and received in LPL. In LPL, a preamble length has to be larger than T to make the receiver to get the preamble because of the independent wakeup timing. When a wakeup period T is short, communication delay becomes small because sensor nodes are frequently wakeup. However, when the T is too short, the power becomes larger because of its frequent communications. On the other hand, when the T is too long, the power also becomes larger. Because the preamble length becomes long ant it adversely affects the power consumption of radio communications. Thus LPL has excellent characteristics in terms of delay and communication power only if a wakeup period T is proper [4].

In I-MAC in Fig. 2(b), the wakeup times on a sender

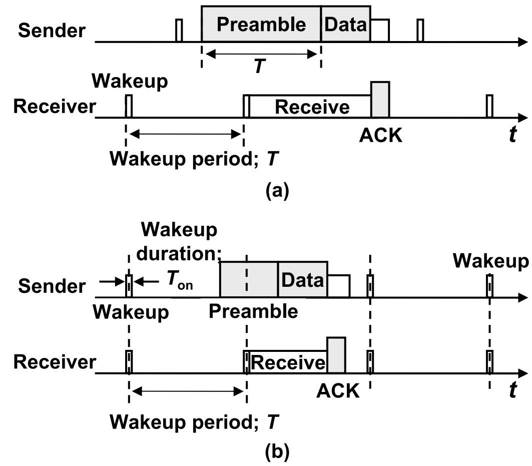

Fig. 2 Timing diagrams of (a) LPL and (b) I-MAC.

and receiver can be synchronized since the actual time is set using the Long-Wave Standard Time Code. Currently, the Long-Wave Standard Time Code is available in Japan, Europe, and the United States. The session of the time synchronization takes place by receiving a time code, broadcasted from a base radio station on a 40/60 Hz amplitude modulation (AM) wave. This way of time synchronization by the AM waves is available even inside of a building (e.g. a wall clock). In addition, the Long-Wave Standard Time Code receivers consume very low power of sub-μW. It is less restricted in terms of the power consumption and usage environment for wireless sensor networks than the Global Positioning System (GPS). GPS receivers has good sensibility, but they consume more than tens of mW [5], [6]. To make matter worse, they operate very high frequency and need external RF filters. It means the cost becomes expensive. Thus the Long-Wave Standard Time Code receivers are low power and low cost compared to GPS receivers, and it is more suitable for wireless sensor networks.

With I-MAC, since the synchronized wakeup time on each node can be easily known in advance, the preamble length can be shortened.

The time code includes the information of time from the second to the year level; it is sent at the rate of 1 bit/s using AM. One piece of time code information is sent in one minute. To ensure current time synchronization, a few pieces of time code information are received normally. As a result, one-time synchronization takes a few minutes. The absolute time error between the base radio station and the node is $\pm 1.5\,\mu$s [8], which means that we could potentially shorten the preamble length to $3.0\,\mu$s if the frequency of the crystal oscillator was not varied with temperature. In practical, the preamble length in I-MAC depends on the absolute time error and the frequency variation of the timer.

On the other hand, in LPL, the preamble length is as long as T. Thus, I-MAC can minimize the preamble length and eliminate needless communication when T is long. This is important to save power because a power consumed by radio frequency (RF) circuits is dominant on a wireless sensor node. Even if I-MAC is sensitive to the frequency variation caused by temperature, it has a great potential to reduce the power in most situations compared with LPL. The power comparison between LPL and I-MAC will be made in Sect. 4.

By using I-MAC, all nodes in the entire network can wake up concurrently in the same period. When each wakes up, it senses the channel during a wakeup duration T_{on}, which is the same action as that taken by LPL.

To send data from a sender, it first sends a preamble to neighboring nodes. After that, it sends data. For the purpose of the collision avoidance, we assume that each sender senses the channel status before sending the preamble. The sender puts off sending the preamble by picking a random contention slot, if the channel is busy.

A receiver also senses the channel during T_{on}. If the channel is idle, the receiver reverts to an idle state. In a case that the receiver correctly receives a preamble and data, it returns an acknowledge signal (ACK), and then reverts to the idle state.

The time at which each node wakes up might be shifted slightly because of the frequency variation. The preamble length must be determined by this fact. Now, we assume $\pm d$ as the maximum time difference from the actual time (see Fig. 3). Thereby, the relative time difference between any two nodes is $\pm 2d$ in the worst case. This tells that the preamble length must have the width of $4d + T_{\mathrm{on}}$. d can be represented by the following equation;

$$d = \frac{D}{C} + F, \qquad (1)$$

where D is the maximum time error per day, C is the number of time synchronization per day, and F is the absolute time error (F is theoretically $3.0\,\mu$s (= $\pm 1.5\,\mu$s), as mentioned before). In the equation, we assume that the time error increases linearly with time.

By using a general crystal oscillator, we can attain the maximum time error caused by the temperature variation below 350 ms per day. If all nodes are at a same temperature, the internal times on the nodes might have a different time from the actual time, but they are all aligned. Hence, there are no time differences among the nodes at all, in this case.

The issue is that, for instance, there are some nodes at a high temperature, and the other are at a low temperature. The time difference between the high-temperature nodes and low-temperature nodes becomes larger along with time. To reduce the time difference, we propose a time correction algorithm in the next section.

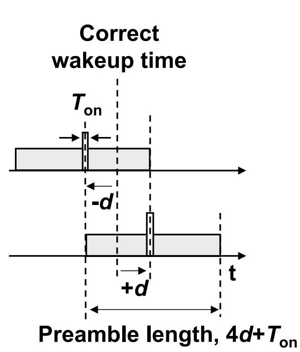

Fig. 3 Time difference between two nodes in I-MAC.

3. Time Correction Algorithm for Temperature Variation

A time correction technique is utilized in the Flooding Time Synchronization Protocol [9]. As well, a similar technique may be used in I-MAC. However, I-MAC requires shorter time error due to the precise internal time. To suppress the time error, we could make frequent time synchronization, although it requires more power in an AM wave circuit.

In this section, we propose a new time correction algorithm for I-MAC. This algorithm is a software approach, and thus consumes much less power than the frequent time synchronization. The algorithm exploits a temperature prediction and a temperature-frequency characteristic of a crystal oscillator. This is based on the fact that the time error is caused by temperature, as pointed in the previous section.

Every measuring cycle T_{measure}, each node measures temperature in the proposed algorithm (note that a thermometer can be easily implemented on a silicon chip). Then, we define the temperature measured in the previous cycle as t_{previous}, and the temperature measured in the present cycle as t_{present}. The predicted temperature in future t_{predict} can be obtained from t_{previous} and t_{present}, by the first-order approximation. Every correcting cycle T_{correct}, the time correction is carried out $n-1$ times during the period of T_{measure}, where $T_{\mathrm{correct}} = T_{\mathrm{measure}}/n$. t_{predict} in the m-th correction ($m = 1, 2, \ldots, n-1$) is represented as the following equations;

$$t_{\mathrm{predict}} = t_{\mathrm{present}} + \left(\frac{t_{\mathrm{present}} - t_{\mathrm{previous}}}{T_{\mathrm{measure}}}\right) T_{\mathrm{correct}} m. \qquad (2)$$

By using t_{predict}, each node calculates the predicted frequency of the crystal oscillator f_{predict} with a temperature-frequency characteristic, and corrects the internal time until the next time synchronization.

As an example, we calculate a time difference between nodes in sunny and shaded areas. Figure 4 is a ground temperature model used in this simulation. This is a case of

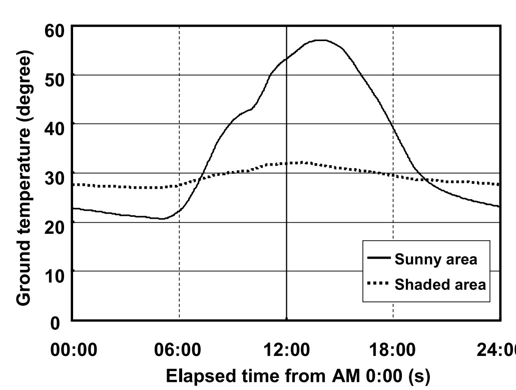

Fig. 4 Temperature model of Tokyo in August.

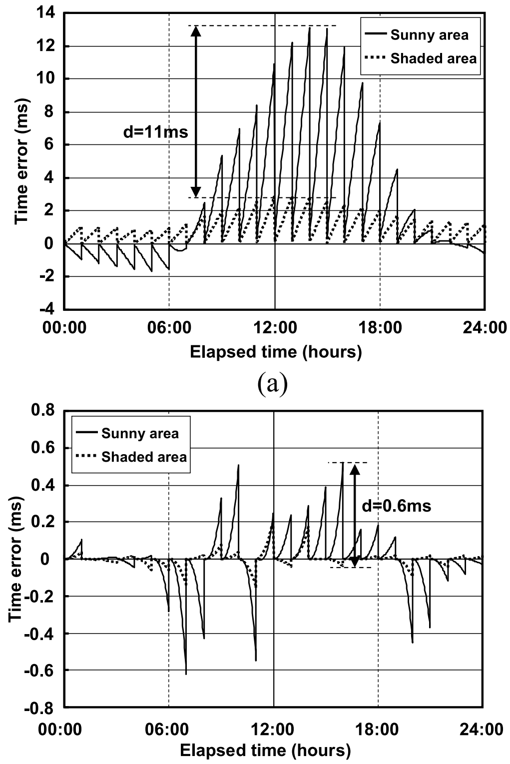

Fig. 5 Time error simulations in cases (a) without and (b) with the time correction algorithm.

Tokyo, in which the temperature difference between sunny and shaded areas becomes large in summer.

Next, we assume that T_{measure} is one hour and T_{correct} is two minutes. Figure 5(a) shows the simulation result of the time errors in the sunny and shaded areas in a case without any time correction. A temperature compensated crystal oscillator (TCXO) that has the lowest temperature coefficient in commercial products is utilized. The maximum time difference d between the nodes in the sunny and shaded areas becomes 11 ms, even with a TCXO. In contrast, the time correction algorithm for the temperature variation suppresses it to 0.6 ms. Figure 5(b) illustrates how the algorithm works.

4. Implementation of I-MAC and Its Power Model

4.1 Prototype of I-MAC

We implemented I-MAC on an off-the-shelf sensor node (S-NODE, Ymatic Ltd. [10]) to demonstrate the effectiveness of I-MAC. Figure 6(a) is the photograph of the prototype. We are now designing a one-chip solution for I-MAC. In combination with small antennas, the prototype sensor node will be shrunk to a watch size.

Figure 6(b) is the block diagram of the prototype. We appended a Long-Wave Standard Time Code unit (LWSTC-Unit) to the S-NODE in order to communicate with the Long-Wave Standard Time Code. In the LWSTC-Unit, we

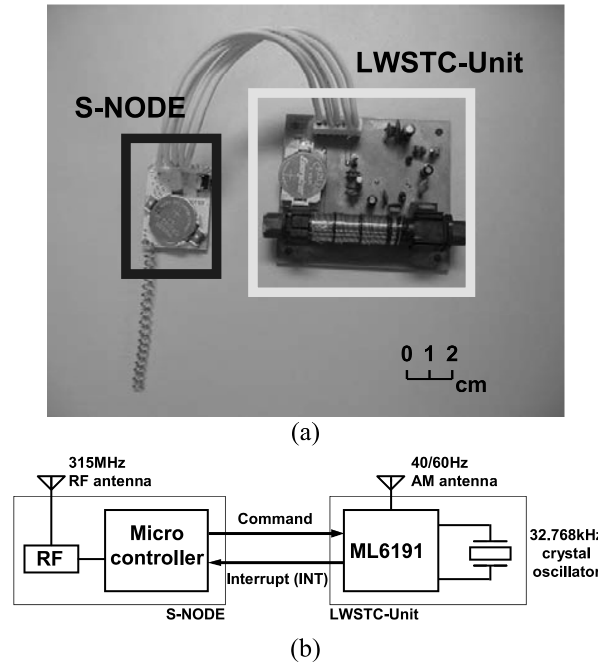

Fig. 6 (a) Photograph and (b) block diagram of I-MAC prototype.

utilized ML6191 produced by Oki Electric Industry Co., Ltd. [11]. This LSI includes a real-time clock (internal timer) operated by a crystal oscillator of 32.768 kHz, and consumes an operating power of 0.0858 mW.

The micro controller sends a command when the session of the Long-Wave Standard Time Code is necessary. After matching with the actual time, the real-time clock on ML6191 internally keeps the time until the next time synchronization. Every single second, the LWSTC-Unit outputs an interrupt (INT) signal. Since the INT signal is synchronized with the actual time with high accuracy, we can expect that all nodes simultaneously receive the INT signals, which is the basis of I-MAC.

4.2 Modeling of Energy Consumption in I-MAC

Next, we will use a model to analytically identify parameters that would be associated most closely with energy consumption. To do so, the model is simplified; packet collisions are ignored.

A power consumption P_{total} to be obtained for LPL or I-MAC is definable with an active time T_{total}, and a total energy consumption E_{total}, as expressed in the following equation;

$$P_{\text{total}} = \frac{E_{\text{total}}}{T_{\text{total}}}. \tag{3}$$

We proceed with the modeling process separately for the energy consumptions at the communication time (sending and receiving times) and idle time.

First, we describe the energy consumption at the communication time. Now, we define M as the average number of data transmissions made during T_{total}, and N as the average number of nodes within the transmission range from any one node. Because every node residing in the transmission area is expected to send data M times, it is concluded that any one node is expected to receive data of MN on average, during which time neither packet collision nor retransmission is presumed to be made. In the data of MN, packet receipts of M are assumed to be made by the own node, and the other $M(N-1)$ are made by the other nodes.

Here, we respectively define each pair of E_{send} and T_{send}, $E_{\mathrm{recv-own}}$ and $T_{\mathrm{recv-own}}$, and $E_{\mathrm{recv-other}}$ and $T_{\mathrm{recv-other}}$, as three of data transmissions; for one piece of data sending, for one piece of data receiving by the own node, and for one piece of data receiving by another node. Using these variables, the communication energy E_{com} is consumed by the sending and receiving data, and the communication time T_{com} is required to accomplish that. They are represented as the following equations;

$$E_{\mathrm{com}} = ME_{\mathrm{send}} + ME_{\mathrm{recv-own}} + M(N-1)E_{\mathrm{recv-other}}, \quad (4)$$

$$T_{\mathrm{com}} = MT_{\mathrm{send}} + MT_{\mathrm{recv-own}} + M(N-1)T_{\mathrm{recv-other}}. \quad (5)$$

The energies consumed to send and receive one piece of data (E_{send}, $E_{\mathrm{recv-own}}$, and $E_{\mathrm{recv-other}}$) and the necessary times for that (T_{send}, $T_{\mathrm{recv-own}}$, and $T_{\mathrm{recv-other}}$) are obtained next. Using S_{ack} as an ACK size, S_{data} as a data length, and R as a channel rate, we define T_{ack} as a time to send or receive an ACK signal, and T_{data} as a time to send and receive one piece of data. We represent them in the form of

$$T_{\mathrm{ack}} = \frac{S_{\mathrm{ack}}}{R}, \quad (6)$$

$$T_{\mathrm{data}} = \frac{S_{\mathrm{data}}}{R}. \quad (7)$$

Defining T_{preamble} as a preamble transmission time, followed by P_{tx}, P_{rx}, and P_{sleep}, which respectively represent power consumptions at the times of transmission, receiving, and sleeping, E_{send} and T_{send} are given as

$$E_{\mathrm{send}} = P_{\mathrm{tx}}(T_{\mathrm{preamble}} + T_{\mathrm{data}}) + P_{\mathrm{rx}}T_{\mathrm{ack}}, \quad (8)$$

$$T_{\mathrm{send}} = T_{\mathrm{preamble}} + T_{\mathrm{data}} + T_{\mathrm{ack}}. \quad (9)$$

The average period spent for each node to start receiving data after it detects a preamble is $T_{\mathrm{preamble}}/2$ (see Fig. 2). Hence, $E_{\mathrm{recv-own}}$ and $T_{\mathrm{recv-own}}$ can be expressed as

$$E_{\mathrm{recv-own}} = P_{\mathrm{rx}}\left(\frac{T_{\mathrm{preamble}}}{2} + T_{\mathrm{data}}\right) + P_{\mathrm{tx}}T_{\mathrm{ack}}, \quad (10)$$

$$T_{\mathrm{recv-own}} = \frac{T_{\mathrm{preamble}}}{2} + T_{\mathrm{data}} + T_{\mathrm{ack}}. \quad (11)$$

When one piece of data is addressed to another node, the other nodes receive the data, but then enter into an idle state without sending an ACK signal. With this fact in mind, $E_{\mathrm{recv-other}}$ and $T_{\mathrm{recv-other}}$ are given as

$$E_{\mathrm{recv-other}} = P_{\mathrm{rx}}\left(\frac{T_{\mathrm{preamble}}}{2} + T_{\mathrm{data}}\right), \quad (12)$$

$$T_{\mathrm{recv-other}} = \frac{T_{\mathrm{preamble}}}{2} + T_{\mathrm{data}}. \quad (13)$$

Next, the energy consumption at an idle time is discussed. Since T is given as the wakeup period (see Fig. 2), the energy E_{T} consumed during T is provided as follows during the idle time;

$$E_{\mathrm{T}} = P_{\mathrm{rx}}T_{\mathrm{on}} + P_{\mathrm{sleep}}(T - T_{\mathrm{on}}), \quad (14)$$

where T_{on} is the wakeup duration mentioned in Sect. 2. Hence, the energy E_{idle} consumed at the idle time can be represented by the following equation;

$$E_{\mathrm{idle}} = \left(\frac{T_{\mathrm{total}} - T_{\mathrm{com}}}{T}\right)E_{\mathrm{T}}. \quad (15)$$

Among the variables defined above, the only difference between LPL and I-MAC is the preamble transmission time T_{preamble}. The T_{preamble} for LPL is T. In contrast, according to (1) and the discussion in Sect. 2, the T_{preamble} for I-MAC is represented by the following equation;

$$T_{\mathrm{preamble}} = \frac{4D}{C} + 4F + T_{\mathrm{on}}. \quad (16)$$

The length of T_{preamble} causes a change in E_{com} and E_{idle}. Hence, we respectively define $E_{\mathrm{com-LPL}}$ and $E_{\mathrm{com-IMAC}}$ for the E_{com}s of LPL and I-MAC, and respectively denote $E_{\mathrm{idle-LPL}}$ and $E_{\mathrm{idle-IMAC}}$ for their E_{idle}s.

Note that, in I-MAC, extra power is needed by the session of the Long-Wave Standard Time Code, which we define as P_{LWSTC}. T_{LWSTC} is a period for one session (90 s on average in good reception condition). Thus, during T_{total}, the energy E_{LWSTC} consumed by the sessions of the Long-Wave Standard Time Code is as follows;

$$E_{\mathrm{LWSTC}} = \frac{T_{\mathrm{total}}}{86400}CP_{\mathrm{LWSTC}}T_{\mathrm{LWSTC}} + P_{\mathrm{clock}}T_{\mathrm{total}}, \quad (17)$$

where P_{clock} is a power of an internal timer and a crystal oscillator. Recall that I-MAC must internally keep the time.

In conclusion, the total energy consumptions, $E_{\mathrm{total-LPL}}$ of LPL and $E_{\mathrm{total-IMAC}}$ of I-MAC, are respectively given as the following equations;

$$E_{\mathrm{total-LPL}} = E_{\mathrm{com-LPL}} + E_{\mathrm{idle-LPL}}, \quad (18)$$

$$E_{\mathrm{total-IMAC}} = E_{\mathrm{com-IMAC}} + E_{\mathrm{idle-IMAC}} + E_{\mathrm{LWSTC}}. \quad (19)$$

4.3 Power Comparison between LPL and I-MAC

By applying the measured values obtained from the prototype to the models discussed in the previous subsection, we can compare the power consumptions between LPL and our proposed I-MAC.

First, we mention the parameters defined in the power models. R is set to 9.8 kbps. On the prototype, T_{on} is set to $2/R$ since we utilize Manchester encoding for on-off-keying. Both S_{ack} and S_{data} are 20 Bytes. T_{LWSTC} is assumed to be 120 s. C, D and F are set to 24 times per day, 350 ms per day and 3.0 μs, respectively. Thereby, d becomes 14.6 ms in I-MAC.

Table 1 shows the power parameters measured on the

Table 1 Power parameter set measured on prototype.

	Parameters	Power (mW)
S-NODE	P_{sleep}	0.0066
	P_{tx}	23.58
	P_{rx}	21.72
LWSTC-Unit	P_{LWSTC}	0.0858
	P_{clock}	0.0084

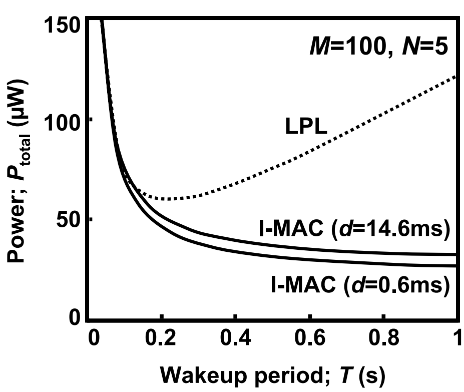

Fig. 7 Relations between power and wakeup period in LPL and I-MAC.

S-NODE and LWSTC-Unit.

Figure 7 shows the relations between the power and the wakeup period for both LPL and I-MAC, calculated with the models and parameters. The figure depicts the case in which the number of data transmissions M is 100 and the number of neighboring nodes N is 5.

In LPL, there exists an optimum wakeup period that results in the least power. In the range of the short wakeup period, the power by the frequent wakeup becomes dominant, while in the range of the long wakeup period, the power by the long preamble increases proportionally to the wakeup period. One must select an adequate wakeup period for a setting of M and N. If that choice were mistakenly made, the impact given to the power consumption would be great. Relationship between M, N and power is mentioned in more detail in [4]. It can be said that the power in LPL is parameter sensitive.

On the other hand, I-MAC has no optimum wakeup period and the power converges as the wakeup period increases. In the case of I-MAC, a longer wakeup period will reduce the power more. Therefore, the wakeup period may be set as large as the allowable range of delay.

5. Low-Power Crystal Oscillator

To achieve further power reduction in I-MAC, we propose a low-power crystal oscillator. The proposed crystal oscillator reduces the power of the internal timer that always operates. It is important to reduce the timer power in order to make a network lifetime longer.

Figure 8 is the proposed Pierce type crystal oscillator circuit that can operate at a supply voltage of 0.5 V. This value is lower than a half of the supply voltage for conventional crystal oscillator.

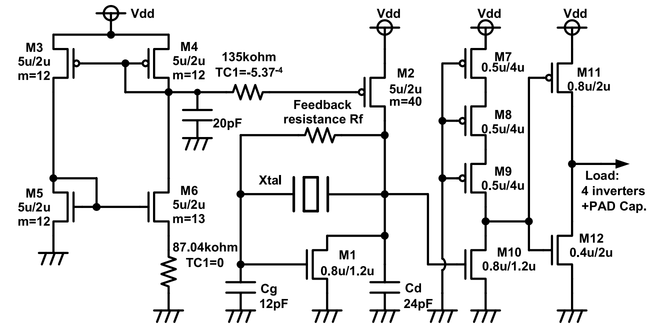

Fig. 8 Low-power crystal oscillator circuit.

Table 2 Power comparison of crystal oscillators.

Oscillators	Power (μW)
Epson Toyocom TG3530	5.1
This work	0.356

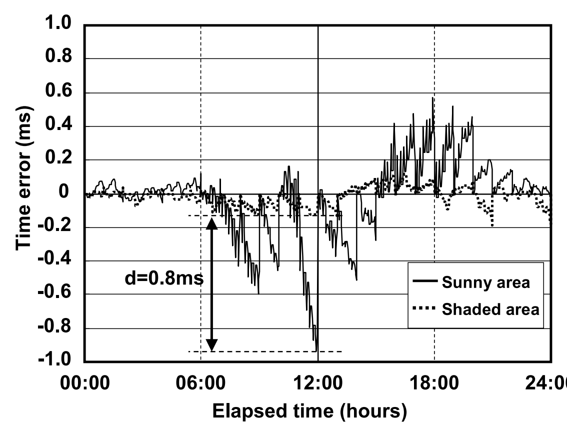

Fig. 9 Time errors in the proposed crystal oscillator.

A Pierce oscillator has simplicity and good frequency stability in nature [12]. The main amplifier is comprised of M1 and M2 transistors. M3 to M6 are a bias circuit. M7 to M12 make up an output buffer. To properly work at the low supply voltage, we utilize an SOI process and low-threshold-voltage transistors. The gate lengths of all transistors are large to suppress the threshold-voltage variation and leakage current. M2 can be biased in deep sub-threshold region while keeping oscillation margin over five times, and its bias current is very small. The low-power feature of the proposed crystal oscillator is achieved by the low supply voltage, the sub-threshold bias method, and the low leakage characteristics of the 0.15-μm SOI process.

Table 2 shows the simulated power of our designed crystal oscillator, compared with a low-power TCXO (Epson Toyocom TG3530 [13]). The power saving of 93% is achieved.

The maximum time difference d caused by the temperature variation is suppressed to 0.8 ms in the proposed crystal oscillator, thanks to the proposed time correction algorithm described in Sect. 3. Figure 9 illustrates the situation when the ground temperature model in Fig. 4 is used.

6. Evaluation of Power Reduction

To compare powers, we implement four scenarios to our prototype. The first one is the conventional LPL. The second is I-MAC using the TCXO without the time correction algorithm. The third is I-MAC using the TCXO with time correction algorithm. The last is our proposed cross-layer design using I-MAC, the time correction algorithm, and low-power crystal oscillator. Table 3 summarizes the preamble lengths and powers of the internal timers used in four scenarios.

Figure 10 shows the average power comparison per day of four scenarios, when the number of transmissions M is 100 and the number of neighboring nodes N is 5. The parameters in the experiment are also listed in the figure. Other parameters are the same as those described in Sect. 4.3. The wakeup period T is set to one second.

I-MAC can reduce the communication power substantially. With the time correction algorithm, the communication power is further reduced to a half because the preamble length $T_{preamble}$ can be reduced on each node. Moreover, by using the proposed crystal oscillator, the power of the internal timer is reduced to 44%. In total, our proposed cross-layer design saves a power of 81%, compared to the conventional LPL.

Here, in the Cycled Receiver MAC, the communication power is varied with M and N (see Eq. (4)). N is the same among all sensor nodes if they are identically-distributed. However, M is a parameter which is affected by a hop count from a base-station. Since sensor nodes gather data from terminals of the network to the base-station, if a node is located besides the base-station, its node's traffic is larger than a node which is located near the terminal of the network tree. This means that the node near the base-station has a large M, and thus the node consumes more communication power.

Figure 11 shows the relation between M and average power consumption of the four cases. I-MAC reduces the communication power significantly when M is large, because I-MAC decreases the duration of idle listening of all sensor nodes. On the other hand, when M becomes 6.5 below, I-MAC gets worse than LPL because of its overhead. There are additional power overheads caused by the time synchronization and internal timer in I-MAC. For this reason, the power reduction ratio is expected to be better near the base-station in I-MAC. Figure 10 is an example of a case that M = 100 and N = 5.

7. Conclusion

In this paper, we presented the cross-layer design using Isochronous-MAC (I-MAC), the time correction algorithm, and low-power crystal oscillator. I-MAC utilizes so-called "wave clock" (Long-Wave Standard Time Code) and can reduce communication power by synchronizing time among nodes. The time correction algorithm shortens the time difference caused by temperature variation, which further reduces the communication power to a half. As a physical layer, we proposed the crystal oscillator to save the power of the internal timer that always operates. The combination of our proposed I-MAC, the time correction algorithm, and the low-power crystal oscillator is our proposed cross-layer design, which achieves 81% reduction of the total power over the conventional Low Power Listening (LPL).

Table 3 Parameters in four scenarios.

	Preamble length, $T_{preamble}$	Power of internal timer, P_{clock}
1. LPL	T	-
2. I-MAC using TCXO w/o time correction	44 ms + T_{on}	8.4 μW
3. I-MAC using TCXO with time correction	2.4 ms + T_{on}	8.4 μW
4. I-MAC using proposed oscillator with time correction	3.2 ms + T_{on}	3.7 μW

T=1000ms: Wakeup period
T_{on}=0.2ms: Wakeup duration

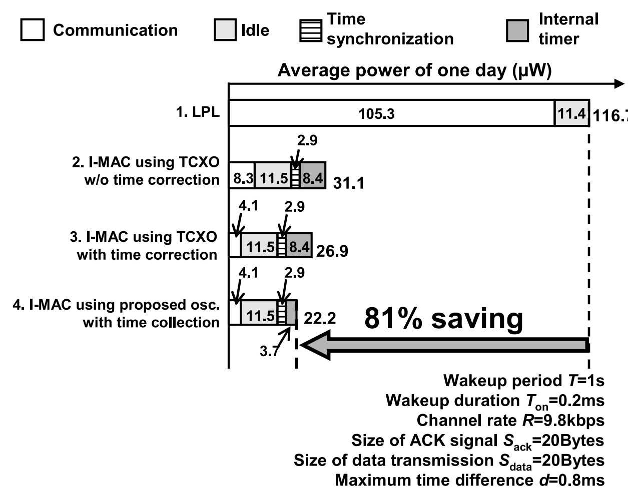

Fig. 10 Power evaluation.

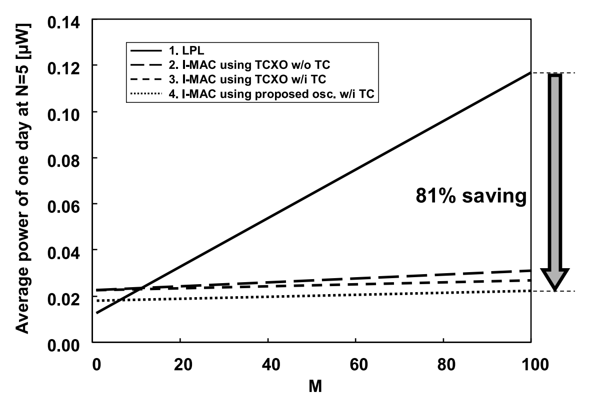

Fig. 11 Evaluation between average number of data transmissions per day and average power.

Acknowledgments

This work has been supported by SCOPE (the Ministry of Internal Affairs and Communications, Japan). The authors would like to appreciate Mr. Shunsuke Baba with Oki Electric Industry Co., Ltd. for helpful advice.

References

[1] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," IEEE Trans. Netw., vol.12, no.3, pp.493–506, April 2004.

[2] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," IEEE Micro, vol.22, pp.12–24, Nov./Dec. 2002.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," Proc. 5th symposium on Operating System Design and Implementation (OSDI 2002), Dec. 2002.

[4] M. Ichien, T. Takeuchi, S. Mikami, H. Kawaguchi, C. Ohta, and M. Yoshimoto, "Isochronous MAC using long-wave standard time code for wireless sensor networks," Proc. International Conference on Communications and Electronics, pp.172–177, Oct. 2006.

[5] T. Kadoyama, N. Suzuki, N. Sasho, H. Iizuka, I. Nagase, H. Usukubo, and M. Katakura, "A complete single-chip GPS receiver with 1.6-V 24-mW radio in 0.18-μm CMOS," IEEE J. Solid-State Circuits, vol.39, no.4, pp.562–568, April 2005.

[6] G. Gramegna, P.G. Mattos, M. Losi, S. Das, M. Franciotta, N.G. Bellantone, M. Vaiana, V. Mandará, and M. Paparo, "A 56-mW 23-mm^2 single-chip 180-nm CMOS GPS receiver with 27.2-mW 4.1-mm^2 radio," IEEE J. Solid-State Circuits, vol.41, no.3, pp.540–551, March 2006.

[7] "http://jjy.nict.go.jp/jjy/trans/index-e.html," Standard-time and Frequency-signal Emission (JJY), Japan Standard Time Project.

[8] T. Morikawa, "http://jjy.nict.go.jp/QandA/reference/Proceeding/sympo-pro1.pdf," Proc. Low Frequency Standard Time and Frequency Signal Symposium.

[9] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," Proc. ACM Conference on Embedded Networked Sensor Systems, pp.39–49, Nov. 2004.

[10] http://www.ymatic.co.jp, Ymatic Ltd.

[11] http://www.okisemi.com/datadocs/doc-eng/ml6191.pdf, Oki Electric Industry Co., Ltd.

[12] S. Farahvash, C. Quek, and M. Mak, "A temperature-compensated digitally-controlled crystal pierce oscillator for wireless applications," Proc. International Solid-State Circuits Conference, pp.352–353, Feb. 2008.

[13] http://www.epsontoyocom.co.jp/product/OSC/set08/tg3530sa/index.html, TC XO-TG3530SA, Epson Toyocom Co., Ltd.

Takashi Takeuchi received a B.E. degree in Computer and Systems Engineering from Kobe University in 2005. He is M.E. degree from Kobe University in 2007. He is currently enrolled in a Doctoral course in Kobe University. His interests include low-power analog circuit designs.

Yu Otake received a B.E. degree in Computer and Systems Engineering from Kobe University in 2007. He is currently a Master's student at Kobe University. His interests include controls for sensor networks media access control.

Masumi Ichien received the B.E. and M.E. degrees in electronic engineering from Kobe University, Hyogo, Japan, in 2005 and 2007, respectively. His interests include controls for sensor networks media access control.

Akihiro Gion received a B.E. degree in Computer and Systems Engineering from Kobe University in 2006. He is currently a Master's student at Kobe University. His interests include controls for sensor networks media access control.

Hiroshi Kawaguchi received B.E. and M.E. degrees in Electronic Engineering from Chiba University, Chiba, Japan, respectively, in 1991 and 1993. He received a Ph.D. degree in Engineering from the University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corp., Kobe, Japan, in 1993, where he developed arcade entertain ment systems. He moved to the Institute of Industrial Science, the University of Tokyo, as a Technical Associate in 1996, and was appointed a Research Associate in 2003. In 2005, he moved to the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan, as a Research Associate. Since 2007, he has been an Associate Professor at the Department of Computer Science and Systems Engineering, Kobe University. He is also a Collaborative Researcher with the Institute of Industrial Science, the University of Tokyo. His current research interests include low-power VLSI design, hardware design for wireless sensor networks, and recognition processors. Dr. Kawaguchi received the IEEE ISSCC 2004 Takuo Sugano Outstanding Paper Award and the IEEE Kansai Section 2006 Gold Award. He has served as a Program Committee Member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as a Guest Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He is a member of IEEE and ACM.

Chikara Ohta received the B.E., M.E. and Ph.D. (Eng.) degrees in communication engineering from Osaka University, Osaka, Japan, in 1990, 1992 and 1995, respectively. From April 1995, he was with the Department of Computer Science, Faculty of Engineering, Gumma University, Gumma, Japan, as an Assistant Professor. In October 1996, he joined the Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima, Tokushima, Japan, as a Lecturer, and there he has been an Associate Professor since March 2001. Since November 2002, he has been an Associate Professor of the Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University, Japan. From March 2003 to February 2004, he was visiting scholar in the University of Massachusetts at Amherst, USA. His research interests include performance evaluation of communication networks. He is a member of IEEE.

Masahiko Yoshimoto received a B.S. degree in Electronic Engineering from the Nagoya Institute of Technology, Nagoya, Japan, in 1975, and an M.S. degree in Electronic Engineering from Nagoya University, Nagoya, Japan, in 1977. He received a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Power Products Inc., Itami, Japan, in April 1977. During 1978–1983 he was engaged in the design of NMOS and CMOS static RAM, including a 64K full CMOS RAM with the world's first divided-word-line structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activities specifically emphasize research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low-power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference during 1991–1993. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received R&D100 awards from R&D Magazine in 1990 and 1996, respectively, for development of the DISP and development of a real-time MPEG2 video encoder chipset.