2007 International Conference on Sensor Technologies and Applications

Cross-Layer Design for Low-Power Wireless Sensor Node using Long-Wave Standard Time Code

Yu Otake, Masumi Ichien, Takashi Takeuchi, Akihiro Gion, Shinji Mikami,
Hidehiro Fujiwara, Hiroshi Kawaguchi, Chikara Ohta, and Masahiko Yoshimoto

Kobe University, Japan

porotake@cs28.cs.kobe-u.ac.jp

***Abstract*—We propose Isochronous-MAC (I-MAC) using the Long-Wave Standard Time Code, 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 requires a precise timer. However, a frequency of a crystal oscillator varies along with temperature, from node to node. We utilize a time correction algorithm to shorten the time difference among nodes. Thereby, the preamble length in I-MAC can be minimized, which saves a 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 I-MAC reduces 81% of the total power, compared to Low Power Listening.**

***Index Terms*—Cross-layer design, crystal oscillator, Long-Wave Standard Time Code, Low Power Listening, MAC**

I. 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, and it is virtually impossible. Sensor nodes must be low power to maximize a total available time in a whole network system.

In this paper, we propose a low-power MAC and cross-layer design with it. Our proposed Isochronous-MAC (I-MAC) [1] is based on Low Power Listening (LPL) [2] 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. 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 a 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 [1] for more detail on I-MAC.

In I-MAC, the time on a sensor node is matched to the actual time using the Long-Wave Standard Time Code, say, at intervals of an hour. One session 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 Section III. 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 IV mentions the power estimation using the measured value of the prototype.

In Section V, 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. Thus, the power of the crystal oscillator is reflected on an idle power. The design of the low-power crystal oscillator is important.

Section VI mentions the experimental result and the future work.

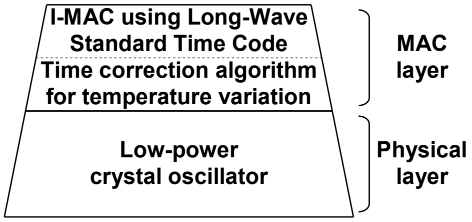

I-MAC using Long-Wave Standard Time Code
Time correction algorithm for temperature variation
MAC layer
Low-power crystal oscillator
Physical layer

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

Fig. 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. We propose the cross-layer design for I-MAC in the following sections.

II. Isochronous-MAC (I-MAC)

Fig. 2 (a) shows the manner in which a packet is sent and received in LPL. LPL has excellent characteristics in terms of delay and communication power if a wakeup period T is short. However, the length of a preamble must be equal to T or longer because of the independent wakeup timing, which incurs a large communication power when T is long.

In I-MAC in Fig. 2 (b), the wakeup times on a sender and receiver can be synchronized since the actual time is set using

0-7695-2988-7/07 $25.00 © 2007 IEEE
DOI 10.1109/SENSORCOMM.2007.48

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). It is less restricted in terms of usage environment than the Global Positioning System.

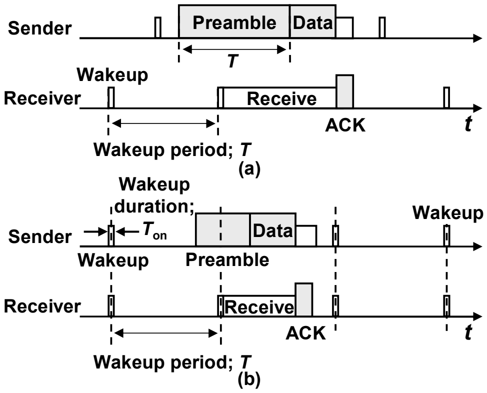


Fig. 2. Timing diagrams of (a) LPL and (b) I-MAC.

With I-MAC, since the synchronized wakeup time on each node can be easily known in advance, the preamble length can be shortened. The absolute time error in the Long-Wave Standard Time Code is only 1.5 μs, which means that we could potentially shorten the preamble length to 1.5 μs 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 while in LPL, it is proportional to the wakeup period 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 Section IV.

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 in light of this fact. Now, we assume ±d as the maximum time difference from the actual time, illustrated in Fig. 3. Thereby, the relative time difference between any two nodes is ±2d in the worst case. This fact tells that the preamble length must have the width of $4d+T_{\mathrm{on}}$. d can be represented by the following equation;

$$d = \frac{D}{C} + F\ , \tag{1}$$

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 1.5 μs, as mentioned before). In the equation, we assume that the time error increases linearly with time.

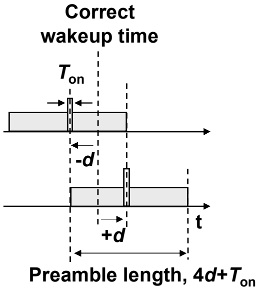


Fig. 3. Time difference between two nodes in I-MAC.

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, if they are all at a same temperature.

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 and low-temperature nodes becomes larger, along with time. To reduce the time difference, we propose a time correction algorithm in the next section.

III. Time Correction Algorithm for Temperature Variation

A time correction technique is utilized in the Flooding Time Synchronization Protocol [3]. 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 $T_{\text{correct}}=T_{\text{measure}}/n$. t_{predict} in the m-th correction (m=1, 2, …, $n-1$) is represented as the following equations;

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

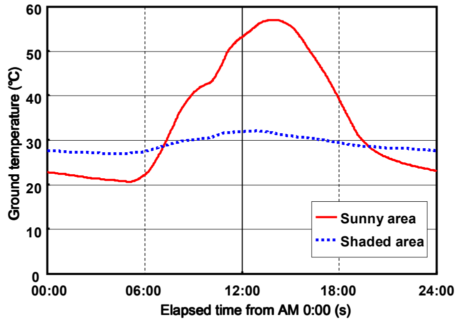
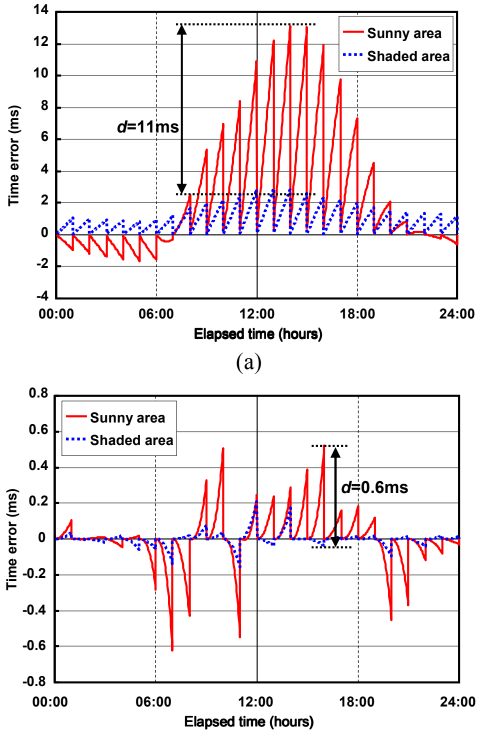


Fig. 4. Temperature model of Tokyo in August.



(a)

(b)

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

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. Fig. 4 is a temperature model used in this simulation. This is a case of Tokyo, in which the temperature difference between sunny and shaded areas becomes large in summer.

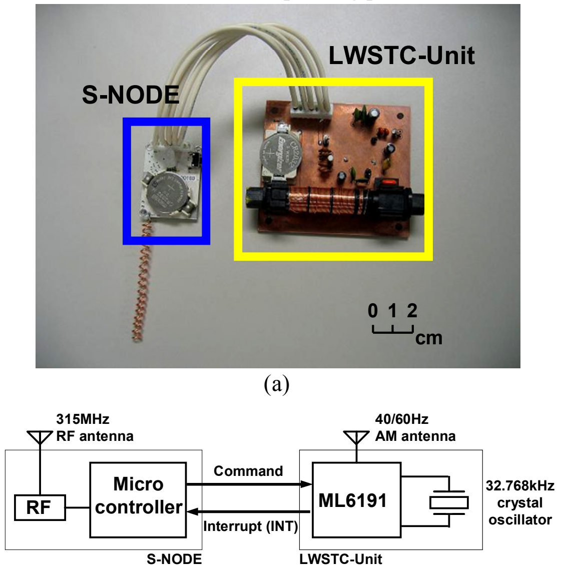
Next, we assume that T_{measure} is one hour and T_{correct} is two minute. Fig. 5 (a) shows the time errors in the sunny and shaded areas in a case without any time correction. The maximum time difference d between the nodes in the sunny and shaded areas becomes 11 ms. On the other hand, the time correction algorithm for the temperature variation suppresses it to 0.6 ms. Fig. 5 (b) illustrates how the algorithm works.

IV. Power Model and Implementation of I-MAC

A. Prototype of I-MAC

We implemented I-MAC on an off-the-shelf sensor node (S-NODE, Ymatic Ltd. [4]) to demonstrate the effectiveness of I-MAC. Fig. 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.

Fig. 6 (b) is the block diagram of the prototype. We appended the LWSTC-Unit to the S-NODE in order to communicate with the Long-Wave Standard Time Code. In the in LWSTC-Unit, we utilized ML6191 produced by Oki Electric Industry Co., Ltd [5]. 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.



(a)

(b)

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

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.

B. 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}, $E_{\text{recv-own}}$ and $T_{\text{recv-own}}$, and $E_{\text{recv-other}}$ and $T_{\text{recv-other}}$, 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;

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

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

The energies consumed to send and receive one piece of data (E_{send}, $E_{\text{recv-own}}$, and $E_{\text{recv-other}}$) and the necessary times for that (T_{send}, $T_{\text{recv-own}}$, and $T_{\text{recv-other}}$) 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

$$T_{\text{ack}} = \frac{S_{\text{ack}}}{R} , \tag{6}$$

$$T_{\text{data}} = \frac{S_{\text{data}}}{R} . \tag{7}$$

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

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

$$T_{\text{send}} = T_{\text{preamble}} + T_{\text{data}} + T_{\text{ack}} . \tag{9}$$

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

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

$$T_{\text{recv-own}} = \frac{T_{\text{preamble}}}{2} + T_{\text{data}} + T_{\text{ack}} . \tag{11}$$

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, $E_{\text{recv-other}}$ and $T_{\text{recv-other}}$ are given as

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

$$T_{\text{recv-other}} = \frac{T_{\text{preamble}}}{2} + T_{\text{data}} . \tag{13}$$

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;

$$E_T = P_{rx}T_{on} + P_{sleep}(T - T_{on}) , \tag{14}$$

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

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

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 Section II, the T_{preamble} for I-MAC is represented by the following equation;

$$T_{\text{preamble}} = \frac{4D}{C} + 4F + T_{\text{on}} . \tag{16}$$

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

$$E_{\text{LWSTC}} = \frac{T_{\text{total}}}{86400}CP_{\text{LWSTC}}T_{\text{LWSTC}} + P_{\text{clock}}T_{\text{total}} , \tag{17}$$

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, $E_{\text{total-LPL}}$ of LPL and $E_{\text{total-IMAC}}$ of I-MAC, are respectively given as the following equations;

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

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

C. 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. T_{on} is set to $2/R$ since, on the prototype, we utilized 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 1.5 µs, respectively. Thereby, d becomes 14.6 ms in I-MAC.

Table I shows the power parameters measured on the S-NODE and Sync-Unit.

Table I. 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 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.

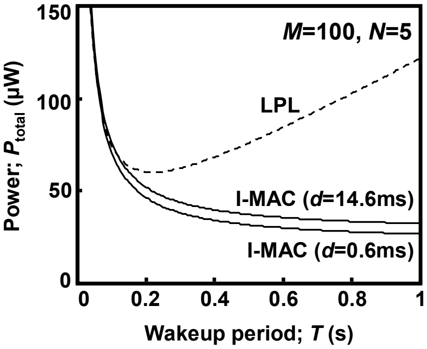


Fig. 7. Relations between power and wakeup period in LPL and I-MAC.

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. 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.

V. 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.

Fig. 8 is the proposed crystal oscillator circuit that can operate at a supply voltage of 0.5V. The low-power feature is achieved with the low supply voltage and a 0.15-µm SOI process. The main amplifier is comprised of from M1 and M2. M3 to M6 are a bias circuit. To properly work at the low supply voltage, we utilize low-threshold-voltage transistors. The gate lengths of M3 to M6 are large to suppress the short-channel effect and threshold-voltage variation. M7 to M12 make up an output buffer.

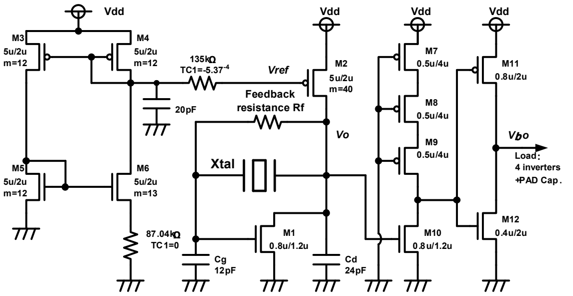


Fig. 8. Low-power crystal oscillator circuit.

Table II shows the simulated power of our designed crystal oscillator, compared with Epson Toyocom TG3530 [6] that has the lowest temperature coefficient in commercial products. The power saving of 93% is achieved.

Table II. Power comparison of crystal oscillators

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

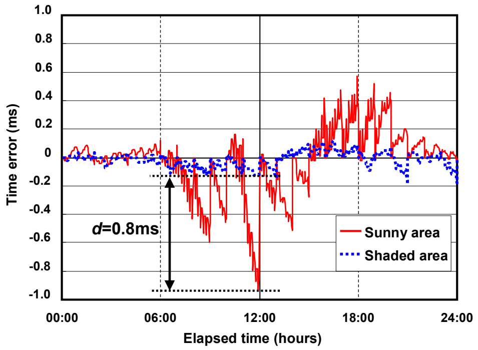
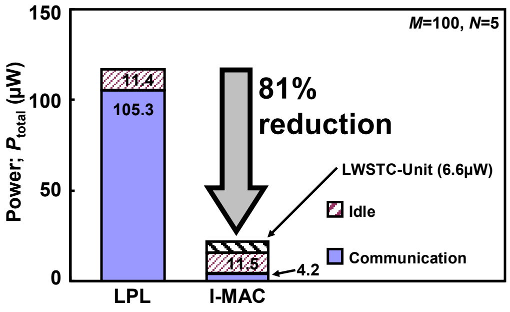


Fig. 9. Time errors in the proposed crystal oscillator.

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

VI. Experimental Result and Future Work

Fig. 10 shows the experimental power comparison between LPL and our proposed I-MAC, when the number of transmissions *M* is 100 and the number of neighboring nodes *N* is 5. The parameters in the experiment are also summarized in the figure. The wakeup period *T* is set to one second. Other parameters are the same to those described in Subsection IV.C. As the power parameters, Table I is reused, except for P_{clock}. We can reduce P_{clock} to 3.7 µW thanks to the proposed low-power crystal oscillator.



Wakeup period T=1s
Wakeup duration T_{on}=0.2ms
Channel rate R=9.8kbps
Size of ACK signal S_{ack}=20Bytes
Size of data transmission S_{data}=20Bytes
Maximum time difference d=0.8ms

Fig. 10. Experimental result.

The combination of our proposed I-MAC, the time correction algorithm, and the low-power crystal oscillator, reduces the total power by 81% compared with LPL. Although there is an additional power overhead caused by the time synchronization, I-MAC is superior to LPL. In particular, the communication power is dramatically reduced. As previously mentioned, we are developing the one-chip solution for I-MAC, which is expected to achieve further power reduction. Fig. 11 is the layout. The test chip is presently under fabrication.

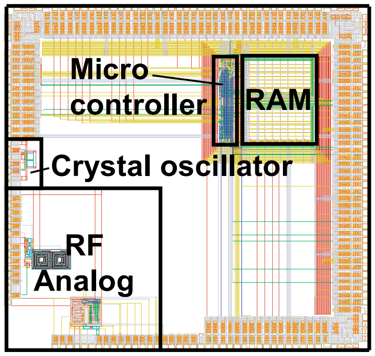


Fig. 11. Layout of I-MAC test chip.

Acknowledgement

This work has been supported in part by the Ministry of Internal Affairs and Communications, Japan, SCOPE, and by the Ministry of Education, Science, Sports and Culture, Japan, Grant-in-Aid for Scientific Research (C), No. 1850052.

References

[1] 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. of International Conference on Communications and Electronics*, pp. 172-177, Oct. 2006.

[2] J. Hill and D. Culler, “Mica: a wireless platform for deeply embedded networks,” *IEEE Micro*, pp. 12-24, Nov./Dec. 2002.

[3] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, “The Flooding Time Synchronization Protocol,” *Proc. of ACM Conference on Embedded Networked Sensor Systems*, pp. 39-49, Nov. 2004.

[4] “http://www.ymatic.co.jp,” Ymatic Ltd.

[5] “http://www.okisemi.com,” Oki Electric Industry Co., Ltd.

[6] “http://www.epsontoyocom.co.jp,” Epson Toyocom Co., Ltd.