

W-MAC: Supporting Ultra Low Duty Cycle in Wireless Sensor Networks[§]

Wooguil Pak*, Kyong-Tak Cho*, Jeongjoon Lee**, and Saewoong Bahk*

*School of Electrical Engineering & Computer Science, Seoul National University, INMC, Seoul, Korea

**Central R&D Lab., LS Industrial Systems, Co., LTD., Seoul, Korea

{wgpak, ktcho}@netlab.snu.ac.kr, jjlee@lisis.biz, sbahk@netlab.snu.ac.kr

Abstract—The duty cycle of a wireless sensor node is a key factor that determines the life time of a wireless sensor network. In general, sensor medium access control protocols reduce the duty cycle to achieve longer lifetime. However, we found out that their performance improvements substantially decrease under extremely low duty cycles (<0.1%). Our proposed W-MAC protocol targets achieving low energy consumption under extremely low duty cycle as well as low duty cycle (>0.1%). It is with estimation of the relative clock speed between the sender and the receiver, and uses it to minimize energy consumption. It saves energy significantly well compared to the existing schemes in an order of ten times, and the performance gap increases with lowering the duty cycle.

Keywords—Ultra low duty cycle, wireless sensor networks, MAC protocol, energy consumption

I. INTRODUCTION

Monitoring like AMR (Automatic Metering Reading) or AMI (Advanced Metering Infrastructure) is an important application of a wireless sensor network. While it requires infrequent communications at a low data rate and insensitive delay, it usually demands a long life time of the network.

In general, wireless sensor MAC protocols are classified according to the requirement of synchronization. Synchronous MACs such as S-MAC [5] and SCP-MAC [3] have low energy consumption for sending packets but are complicated due to the need of synchronization. This overhead makes synchronous MAC algorithms inappropriate for monitoring applications. Asynchronous MACs such as X-MAC [1] and WiseMAC [2] are very simple but they spend much energy in finding the neighbor's wakeup time. Due to this reason, under ultra low duty cycle, asynchronous MAC algorithms are not attractive either for monitoring applications. Moreover, a phenomenon called clock drift, which is known as the most critical factor deciding overall performance when the duty cycle and traffic load are very low, can further degrade the performances of synchronous and asynchronous MAC protocols.

In this paper, we propose a new MAC algorithm called W-MAC (Wide duty cycle MAC) that works effectively over a

wide range of duty cycles. It aims at lowering the energy consumption under low to extremely low duty cycle. It reduces the energy consumption for communication by minimizing the negative effect of the clock drift. Unlike SCP-MAC and WiseMAC, W-MAC is designed to operate also in a heterogeneous network where each node may have a different maximum clock drift rate and has no need of knowing its value.

The rest of the paper is organized as follows. Section 2 explains the clock drift effect briefly and analyzes the clock drift measurement experiments. We explain our proposed W-MAC algorithm in Section 3. We provide simulation results in Section 4, and conclude in Section 5.

II. EFFECTS AND CHARACTERISTICS OF CLOCK DRIFT

The term 'clock drift' means that the clock drifts apart from the other clock when the oscillators or crystals used by clocks don't run at an exact same speed. We will derive the effect of duty cycle and maximum clock drift rate on energy consumption of current sensor MAC protocols. Then, we will present the measurement results for clock drift.

A. Maximum clock drift rate

Most of the wireless sensor nodes like TelosB use crystals or oscillators which maximum clock drift rate [2][3] is 30ppm (Parts Per Million). This implies that two sensor nodes have at most 30 μ sec error every 1sec. Unlike synchronous MAC protocols [3][5] which use synchronization messages for error correction, the clock drift in asynchronous MAC protocols causes inaccurate estimation of the receiver's wakeup time which increases communication overhead [2].

B. Effect of duty cycle and maximum clock drift rate

We explained the term, maximum clock drift rate. Now, we derive the effect of the duty cycle and maximum clock drift rate on energy consumption for the existing sensor MAC protocols.

Some MAC protocols like SCP-MAC and WiseMAC exploit clock drift. When the maximum clock drift rate is θ , they expect the receiver to wake up between $W-2\theta W$, $W+2\theta W$ where W is the awake interval [2][3]. To investigate the effect of clock drift on energy consumption, we make the following assumptions.

- Node A sends a packet to node B every P (sec).
- Duty cycle is denoted by Δ/W . It is assumed that $P \gg W$, $\theta \ll WP^{-1}$.

[§] This work was partly supported by LS industrial Systems and the IT R&D program of MKE/IITA [2008-F-007-01, Intelligent Wireless Communication Systems in 3 Dimensional Environment].

- Nodes A and B are synchronized at $t=0$.
- Energy consumption for RX and TX is p per unit time.
- Node A sends strobed preambles [1] and node A can detect node B's wakeup when it receives the ack from B.
- Delays for Ack and packet transmission are ignored.

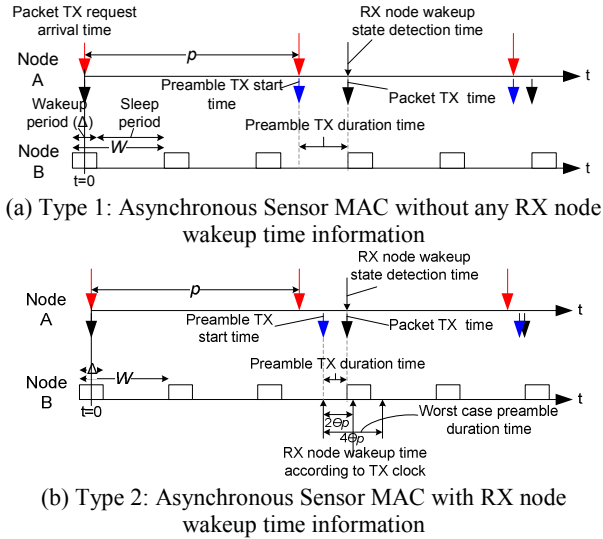


Fig. 1. Asynchronous Sensor MAC type comparison

Under the above assumptions, we compare two types of asynchronous MAC: Type1 and 2. Type1 such as X-MAC does not require any clock information of the receiver, whereas Type2 such as WiseMAC does. Since our analysis of Type2 can be applied to synchronous MAC protocols like SCP-MAC, we omit their analysis. Now we can write the average energy consumptions for Type1 and Type2 as

$$\overline{E}_1 = \overline{E}_{1,tx} + \overline{E}_{1,rx} = \frac{pW}{2P} + \frac{p\Delta}{W} = p\left(\frac{W}{2P} + \frac{\Delta}{W}\right), \quad (1)$$

$$\overline{E}_2 = \overline{E}_{2,tx} + \overline{E}_{2,rx} = \min(2\theta p, \frac{pW}{2P}) + \frac{p\Delta}{W} = p\left(2\theta + \frac{\Delta}{W}\right). \quad (2)$$

From the equations, we obtain $\overline{E}_1 \geq \overline{E}_2$. Therefore we consider the energy consumption of Type2 as a lower bound. For typical parameters of $\Delta=0.52$ msec [1], $\theta=30$ ppm, we obtain the ratio of average energy consumption of TX to that of RX as

$$\frac{\overline{E}_{2,tx}}{\overline{E}_{2,rx}} = \frac{2\theta W}{\Delta} = \frac{1}{8.6} \cdot W. \quad (3)$$

Hence the energy consumption of transmission overwhelms that of reception when $W > 8.6$ sec or $\Delta/W < 0.006\%$. For example, 63% of total energy consumption comes from transmission when W is 20 sec and 72% when W is 40 sec. From these examples, under an extremely low duty cycle, we can say that decreasing the transmission energy is important.

C. Relative clock drift measurement and characteristics analysis

We measured the relative clock drift of sensor nodes that reflects the delay caused in MAC and PHY layers. From the measurements, we obtained some important characteristics of

sensor clocks that can be used for efficient MAC algorithm design. Our experiment is the first trial to measure the relative clock drift including the delay caused in MAC and PHY layers.

We built two testbeds, one using 10 TelosB nodes and the other using 10 MicaZ nodes¹, all located very near from each other. Fig. 2 shows some part of the measurement for duration of 2729 min in outdoor environments. The X axis presents time in minute, and the Y axis presents clock difference (clock/min) or relative clock speed with respect to the base node (node 0).

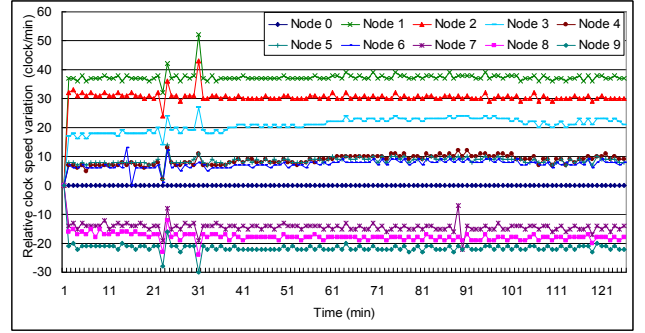
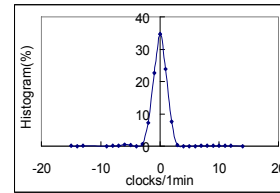


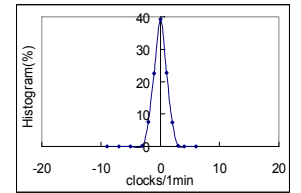
Fig. 2. Relative clock speed (TelosB, Outdoor, 1min interval).

From Fig. 2, we observed the following characteristics.

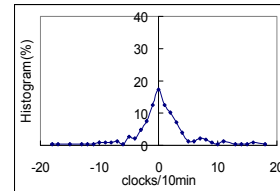
- Relative clock speed is between $(-2\theta, 2\theta)$.
- Each node has a small variation of relative clock speed ($< 2\theta$).
- Nodes in the same environment have similar relative clock speed variation (node 3, 4, 5 and node 1, 2, 6, 7, 8).
- All abrupt changes of the clock speed only occur for a very short duration. This means that the phenomenon is due to the delays from the MAC and PHY layers.



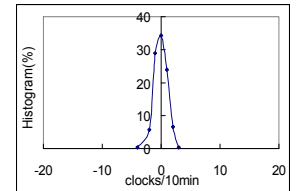
(a) TelosB (1min interval)



(b) MicaZ (1min interval)



(c) TelosB (10min interval)



(d) MicaZ (10min interval)

Fig. 3. Relative clock speed variation.

Fig. 3 illustrates the variation of relative clock speed of TelosB and MicaZ. The X axis presents acceleration and the Y axis the number of samples. Their distributions are very similar to Gaussian. But some local peaks from the results using TelosB, which occur when the acceleration is greater than 5, are

¹ We omit the measurements of MicaZ since the results are very similar

caused by delays at MAC and PHY layers. Note that the relative clock speed variations are smaller than 20 and very stable for all cases. Since there are no local peaks for MicaZ, it seems that the local peaks have dependency on the radio hardware chips.

III. W-MAC PROTOCOL

W-MAC is an asynchronous MAC protocol based on X-MAC. It makes use of the relative clock speed characteristics in order to minimize energy consumption. The unique components of our protocol are: relative clock speed estimation algorithm, delay margin estimation algorithm and relative clock drift margin estimation algorithm. These algorithms help W-MAC to estimate the effect of the relative clock drift rate and delay elements adaptively, and predict the wakeup time of the receiver node very efficiently.

We first explain the overall operation of W-MAC. Then, we present the three estimation algorithms.

A. Basic operation

W-MAC attempts to estimate the wakeup time of the receiver node if the MAC layer receives a packet transmission request from the upper layer. For the estimation, it needs a transmission history of at least two packets for each receiver node.

1) Initial packet transmission

When W-MAC tries to send a packet to a receiver for the first time, it sends a packet with the same way as X-MAC as shown in Fig. 4. W-MAC (or X-MAC) sends strobed preambles repeatedly until it receives a preamble ack packet. The receiver should reply with an ack packet if it receives the preamble during the wakeup period. If the transmitter receives the preamble ack, it sends a data packet to the receiver. The transmitter keeps sending preambles during the awake interval W at most and $0.5W$ on average [1].

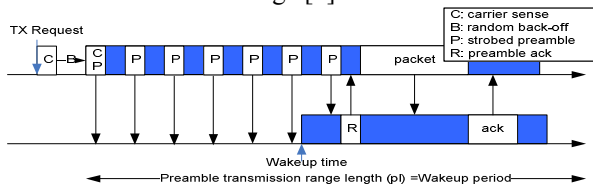


Fig. 4. W-MAC initial packet transmission.

2) Second packet transmission

When W-MAC is requested to transmit a packet to the same receiver for the second time, it sends the packet with a maximum clock drift rate, similarly to WiseMAC or SCP-MAC, owing to no information about the relative clock speed between the transmitter and the receiver. The transmitter should transmit preambles repeatedly for a time of $4\theta L$ ($L = t_{current} - t_{prev}$, t_{prev} : initial transmission time, $t_{current}$: current time) at most, and $2\theta L$ on average as in Fig. 5 [2][3].

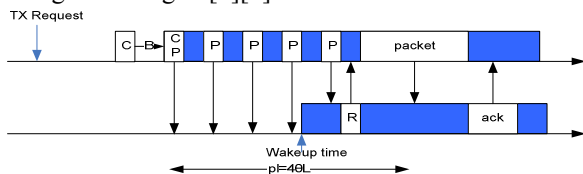


Fig. 5. W-MAC second packet transmission.

If the maximum clock drift rates are not known or they vary due to the heterogeneous network, W-MAC may use a sufficiently large value such as 300ppm.

3) Packet transmission in normal

After W-MAC has sent more than two packets to the same receiver, it can now send data packets with a very short preamble due to the precise estimation of the receiver's wakeup time as shown in Fig. 6. In the following subsections, we present the details of how W-MAC determines the precise time of when it starts and stops transmitting preambles.

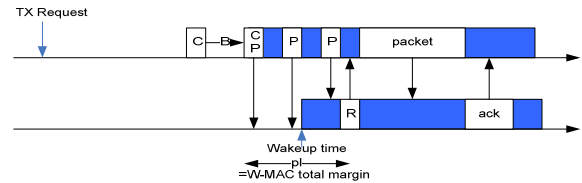


Fig. 6. W-MAC packet transmission in normal.

B. Relative clock speed estimation algorithm

We need to know the relative clock speed of the receiver to estimate its wakeup time. The following procedures are proposed for the estimation.

Firstly, W-MAC saves the recent preamble transmission time (t_{pr}) when it receives an acknowledgement packet following a preamble. If the currently saved preamble transmission time is $t_{pr,n}$ and the new preamble transmission time is $t_{pr,n+1}$, the difference, $t_{pr,n+1} - t_{pr,n}$ should be $K \cdot W$, where K is a positive integer, based on the receiver clock not on the transmitter clock.

We can calculate the relative clock speed of the receiver node based on the transmitter node by using this difference. From now on, we will calculate the relative clock speed ratio, r_n , of the receiver node's clock speed to the transmitter node's clock speed, instead of the relative clock speed itself for convenience. Then we have,

$$r_{n+1} = \frac{1}{t_{pr,n+1} - t_{pr,n}} \left\langle \frac{t_{pr,n+1} - t_{pr,n}}{W} \right\rangle \cdot W, \quad (4)$$

where $\langle \cdot \rangle$ is a round-off operator.

By using Exponentially Weighted Moving Average (EWMA), we can express the estimated ratio e_{n+1} of the relative clock speed ratio for the new transmitting packet using r_n as

$$e_{n+1} = \alpha \cdot r_n + (1 - \alpha) \cdot e_n, \text{ where } e_0 = r_0, 0 < \alpha < 1. \quad (5)$$

C. Delay margin estimation algorithm

We already mentioned that a sensor node can experience large clock speed variation caused by the delay at MAC and PHY layers, not by real clock speed variation. This means that we need to put some margin to incorporate the error, caused by these delay elements, when we decide the period for preamble transmission. The delay comes from packet queues in the MAC and PHY layer, transmission delay, interrupt handling, encoding, decoding and byte alignment. It has an upper bound value, independent of $t_{pr,n+1} - t_{pr,n}$. W-MAC calculates the error ε_n (the difference of the estimated wakeup time and the actual wakeup time) whenever it saves t_{pr} . Afterwards it renews the margin to $\beta \cdot \varepsilon_n$ if $\beta \cdot \varepsilon_n$ is larger than the current delay margin $m_{d,n}$ ($\beta > 1$). With this method, we can estimate the upper bound

of the delay margin. The delay margin estimation algorithm can be expressed as

$$m_{d,n+1} = \max(m_{d,n}, \beta \cdot \epsilon_n), \quad \beta > 1. \quad (6)$$

D. Relative clock drift margin estimation algorithm

The relative clock drift margin is needed to consider the error caused by relative clock drift. Unlike the delay margin, it is dependent on $t_{pr,n+1} - t_{pr,n}$ because the relative clock drift is caused by relative clock speed variation. To estimate the relative clock drift margin, we should estimate the relative clock drift rate in a similar way to the delay margin derivation. We can derive the relative clock drift margin by multiplying the clock drift rate by a time length $\tau_{pr,n+1} - t_{pr,n}$, where $\tau_{pr,n+1}$ is the estimation of the receiver's wakeup time for the next packet transmission (estimated $t_{pr,n+1}$) and θ_n is the margin for clock drift rate, which should be multiplied by a time value (clock/sec) to be considered as clock units.

$$\theta_{n+1} = \max(\theta_n, \gamma \cdot \frac{\epsilon_n}{t_{pr,n} - t_{pr,n-1}}), \quad \gamma > 1, \quad (7)$$

$$m_{c,n+1} = \theta_{n+1} \cdot (\tau_{pr,n+1} - t_{pr,n}), \quad \tau_{pr,n+1} = \left\lceil \frac{t_{tx_req,n+1} - t_{pr,n}}{W} \right\rceil \cdot W, \quad (8)$$

where $\lceil \cdot \rceil$ is a ceiling operator and $t_{tx_req,n+1}$ means the time when W-MAC receives the $(n+1)$ th packet transmission request from the upper layer.

E. Integrated Margin

Fig. 7 shows the result when we apply the delay margin and the relative clock drift margin independently to estimate the wakeup time of the receiver node. We assume that we know the relative clock counter of the receiver node at T_0 .

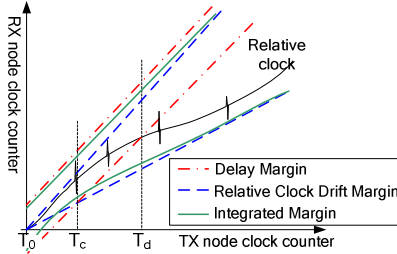


Fig. 7. Integrated margin.

It is profitable to use the delay margin when the elapsed time is small such as $(T_0 \sim T_d)$ and to use the relative clock drift margin when the elapsed time is big $(> T_c)$. Therefore it is convenient and effective to combine the two margins and we name it as an integrated margin. Integrated margin, m_{n+1} can be expressed as follows.

$$m_{n+1}(\tau_{pr,n+1}, t_{pr,n}) = \delta_{n+1} \cdot m_{d,n+1} + (1 - \delta_{n+1}) \cdot m_{c,n+1}, \quad (9)$$

$$\delta_{n+1} = \frac{M}{M + (\tau_{pr,n+1} - t_{pr,n})}, \quad (10)$$

where M is a constant and it is configurable.

The integrated margin is affected mainly by the delay margin if $\tau_{pr,n+1} - t_{pr,n}$ is smaller than M , and by the relative clock drift margin if $\tau_{pr,n+1} - t_{pr,n}$ is bigger than M . In Fig. 7 we can see that the integrated margin works effectively for all time range.

For the integrated margin, the modified delay margin, $m_{d,n+1}^*$ and relative clock drift rate margin, θ_{n+1}^* can be obtained from the below equations.

$$m_{d,n+1}^* = \max(m_{d,n}, \beta \cdot \epsilon_{d,n}), \quad \epsilon_{d,n} = \epsilon_n \cdot \frac{M}{M + (t_{pr,n} - t_{pr,n-1})}, \quad (11)$$

$$\theta_{n+1}^* = \max(\theta_n, \gamma \cdot \frac{\epsilon_{c,n}}{t_{pr,n} - t_{pr,n-1}}), \quad \epsilon_{c,n} = \epsilon_n \cdot \frac{t_{pr,n} - t_{pr,n-1}}{M + (t_{pr,n} - t_{pr,n-1})}. \quad (12)$$

IV. PERFORMANCE EVALUATION

We simulate W-MAC using ns2. We embed relative clock speed measurements into ns2 to simulate clock drifts. We use a 5x5 grid topology and the distance between the nodes is set to 50 m. We also set the carrier sensing range to 130 m, the transmission rate to 256Kbps. And we assume that transmission, reception and listening consume the same energy per unit time.

We simulate five different traffic scenarios. In each scenario, we choose five sources at random among the nodes of which distance is longer or equal to 2 hops from the sink node. The sink node is located at the center of the topology.

We compare W-MAC with Anycast [4], the original X-MAC and a modified X-MAC denoted as X-MAC+40L which has an added clock estimation functionality used for WiseMAC and SCP-MAC. For W-MAC configurations, all parameters are set as $\alpha=0.1$, $\beta=2$, $\gamma=2$, $M=1$ hour, $m_{d,0}=10$ clock and $\theta_0=6 \times 10^{-8}$ clock/sec.

Additional to the simulation, we implemented W-MAC, X-MAC and X-MAC+40L with TelosB and TinyOS, and compared it with other competitive protocols.

Fig. 8 and 9 shows the simulation results. Each number at the center of the graphs presents the performance improvement of W-MAC over the best among the other competitive ones. And Fig. 10 shows the experimental testbed results.

A. Total energy consumption

Fig. 8(a) and 8(b) show that W-MAC outperforms the others in total energy consumption. Clearly, the improvement is enlarged with the increase of packet TX interval P . On the contrary, X-MAC+40L improves less as P increases. Although the total energy consumptions of X-MAC+40L and W-MAC are determined by the clock drift rate, the performance gap comes from the fact that W-MAC can minimize the effect of clock drift by using the proposed estimation algorithms.

W-MAC can improve the performance over 90% against X-MAC+40L with the increase of W , when we keep the ratio of W to P the same in Fig. 8(c). W-MAC can communicate using very low energy thanks to the estimation algorithms.

B. Transmission energy consumption

Transmission energy consumption is a very important factor because it determines the life time of the source node. We can see that the transmission energy of W-MAC is very small. It uses only 15% of the transmission energy compared to X-MAC+40L which is the best among the competitive ones.

C. Packet reception ratio

Packet reception ratio is given as the ratio of the number of packets received by the sink node to that sent by the source nodes.

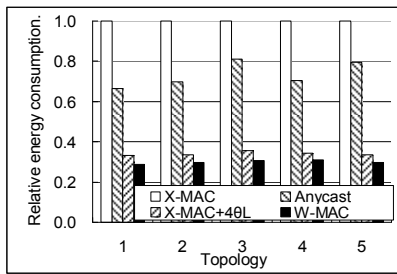
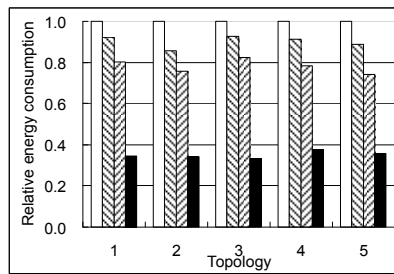
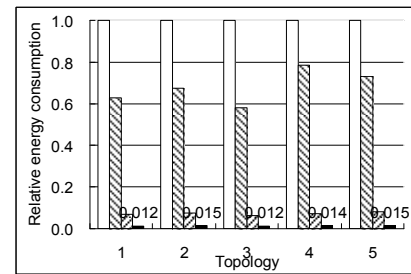
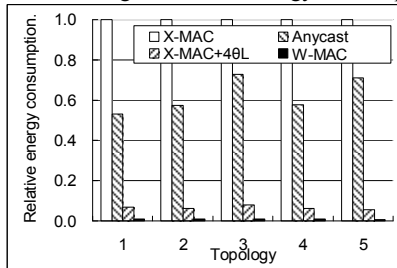
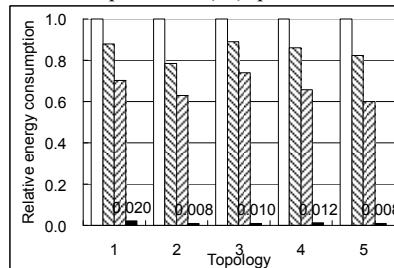
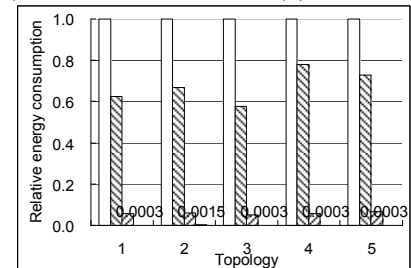
(a) $W=10\text{sec}, P=3\text{min}, S=30\text{min}$ (b) $W=10\text{sec}, P=30\text{min}, S=300\text{min}$ (c) $W=300\text{sec}, P=90\text{min}, S=900\text{min}$ Figure 8. Total energy consumption with wakeup interval (W), packet tx interval (P), and total simulation time (S).(a) $W=10\text{sec}, P=3\text{min}, S=30\text{min}$ (b) $W=10\text{sec}, P=30\text{min}, S=300\text{min}$ (c) $W=300\text{sec}, P=90\text{min}, S=900\text{min}$

Figure 9. Transmission energy consumption.

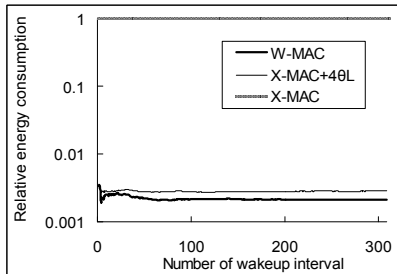
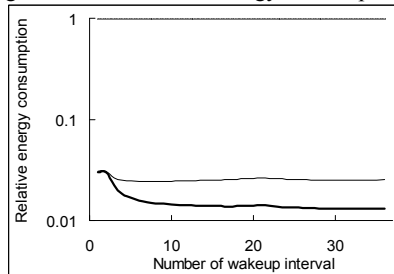
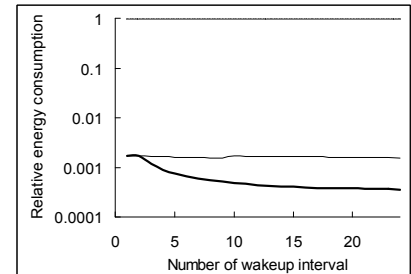
(a) $W=10\text{sec}, P=3\text{min}$ (b) $W=10\text{sec}, P=30\text{min}$ (c) $W=300\text{sec}, P=90\text{min}$

Figure 10. Total energy consumption in the testbed.

Only W-MAC and X-MAC+40L show 100% of packet reception ratio for all cases. An interesting result is that Anycast shows an unexpectedly low performance compared to W-MAC and X-MAC+40L. This is due to the long awake interval, for these experiments, making Anycast to have a high probability of occupying the channel for a long time. This can cause many collisions or random back-offs leading to packet drops and time-out respectively.

TABLE I
Packet reception ratio with awake interval (W), packet tx interval (P), and total simulation time (S).

$W(s), P(m), S(m)$	X-MAC	X-MAC+40L	W-MAC	Anycast
10,3,30	92.4%	100%	100%	98.4%
10,30,300	100%	100%	100%	100%
300,90,900	99.6%	100%	100%	99.6%

D. Experimental Evaluation

In Fig. 10, W-MAC consumes 62% and 0.1% of transmission energy compared to X-MAC+40L and X-MAC respectively when $W=10\text{sec}, P=3\text{min}$. If P is increased to 30min and W is fixed to 10sec, X-MAC+40L shows a bad performance due to increased number of preamble packets. But W-MAC consumes only 16.6% of X-MAC+40L by maintaining a small number of used preamble packets. When $W=300\text{sec}, P=90\text{min}$, W-MAC uses only 15.8% power consumption of X-MAC+40L.

V. CONCLUSION

In this paper, we proposed a new sensor MAC algorithm named W-MAC to minimize the energy consumption. It is designed to support extremely low duty cycle as well as low duty cycle. Owing to our estimation algorithms for relative clock speed and integrated margin, W-MAC can maximize the life time of a wireless sensor network. The performance improvements of W-MAC are confirmed by extensive simulations and evaluations. In future work, we will extend W-MAC to support various applications like WSN with high data rate or delay-sensitive traffic as well as monitoring.

REFERENCES

- [1] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," ACM SenSys, November 2006.
- [2] A. El-Hoiydi, and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks," ACM SenSys, November 2003.
- [3] W. Ye, F. Silva, and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," ACM SenSys, November 2006.
- [4] S. S. Kulkarni, A. Iyer, and C. Rosenberg, "An Address-Light, Integrated MAC and Routing Protocol for Wireless Sensor Networks," IEEE/ACM Transactions on Networking, August 2006.
- [5] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Transactions on Networking, June 2004.