Dynamic Threshold Selection for Frame Length-Based Wake-Up Control

Suhua Tang, *Member, IEEE*, Hiroyuki Yomo, *Member, IEEE*, and Sadao Obana

***Abstract*—In frame-length based wake-up control, a wake-up receiver smoothens the signal envelope by low-pass-filtering (LPF) before frame length detection. However, the LPF also has a side effect of distorting the rising and falling edges of the envelope. This letter analyzes the effect of LPF on frame length detection, and suggests measuring potential frame lengths with multiple thresholds and from them selecting the correct one. The analysis shows that using thresholds with exponential intervals helps to achieve a good performance with only a small number of thresholds. The effectiveness of the proposed scheme is also confirmed by experimental results.**

***Index Terms*—Wake-up control, frame length modulation, low pass filtering.**

I. Introduction

MOST wireless communication devices (e.g., Wi-Fi modules, ZigBee modules) work with batteries. These devices are not always used, but still consume much power in their idle state. Sleep in the idle state helps to reduce power consumption, and remote wake-up control is an important method to ensuring real-time response. This involves the design of wake-up signals and wake-up receivers (WuRx) [1]. However, it usually requires to install special hardware to transmit wake-up signals [2]–[4]. It is desirable to transmit wake-up messages via off-the-shelf hardware so as to reduce the cost of deployment, and successful examples can be found for wireless LANs (WLANs). The frame length property (physical transmission time) of WLAN signals is leveraged to convey wake-up ID and detected either by a ZigBee device [5] or a low-cost WuRx [6], [7]. Frame length modulation is also suggested for the wake-up control of sensor nodes [8].

Fig. 1(a) shows a framework of remote wake-up control for wireless nodes, where each node is equipped with a low-power WuRx. A wake-up ID is transmitted by varying the frame length (Fig. 1(b)) and detected by the WuRx (Fig. 1(c)). The wake-up signal passes an envelope detector (DET) and a low pass filter. Then, the MCU periodically samples the smoothed envelope, and uses counters to record the number of consecutive samples whose values are above predefined thresholds. The counter values are regarded as frame lengths. The MCU activates the node, when the detected frame lengths match the assigned ones.

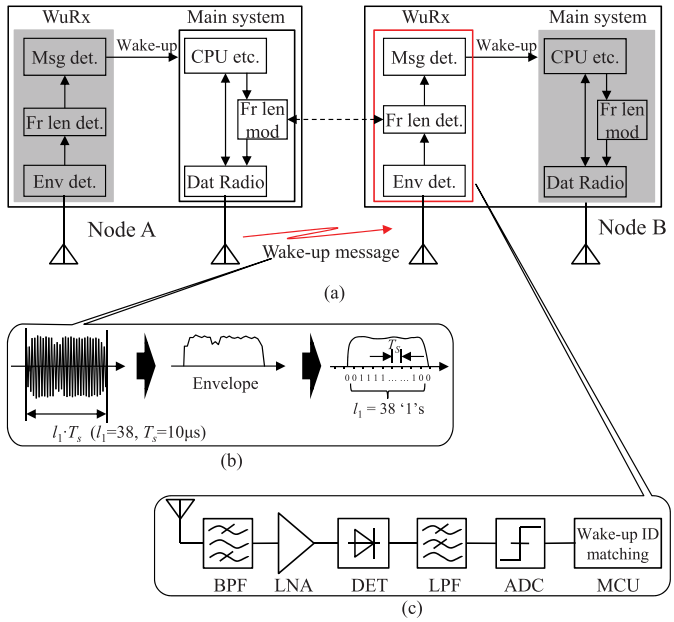

Fig. 1. Framework of wake-up control by exploiting frame length of wireless signals to convey wake-up messages. (a) Framework for wake-up control. (b) Envelope of wireless signals. (c) Wake-up receiver based on envelope detection.

However, the LPF has a side effect of distorting the rising and falling edges of the envelope, which affects the accuracy of frame length detection.

In this letter, we analyze the effect of LPF on frame length detection, and show that the performance of frame length detection depends on the selection of a suitable threshold. Although it is possible to save all the samples of an envelope and find the optimal threshold after the signal ends, it both takes much room for storage and delays system response. As an alternative, we suggest measuring potential frame length at multiple thresholds simultaneously and from them selecting the correct one. To reduce the number of thresholds, thresholds are separated with exponentially increasing intervals. The overall performance of frame length detection is evaluated by experimental data obtained on a prototype WuRx.

II. Impact of LPF and Threshold

For the simplicity, the LPF is constructed by a series resistor (R)-capacitor (C) circuit, as shown in Fig. 2(a). This LPF has a time constant $\tau = RC$ and a cutoff frequency (COF) $f_{COF} = 1/(2\pi\tau)$, which is its 3 dB bandwidth. On one hand, a lower COF leads to a better smoothing effect on the envelope. On the other hand, the frame length is defined by the rising and falling edges of an envelope. The two edges, being sharp (Fig. 2(b)) at the transmission time, are smoothed as well (Fig. 2(c)). Then, with a given threshold, the measured frame length will deviate from the true value, leading to false negative events.

Manuscript received May 5, 2015; accepted August 25, 2015. Date of publication September 1, 2015; date of current version December 15, 2015. This work was supported in part by JSPS KAKENHI Grant Number 26820149. The associate editor coordinating the review of this paper and approving it for publication was R. Madan.

S. Tang and S. Obana are with the Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan (e-mail: shtang@uec.ac.jp; obana@cs.uec.ac.jp).

H. Yomo is with the Faculty of Engineering Science, Kansai University, Osaka 564-8680, Japan (e-mail: yomo@kansai-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LWC.2015.2475268

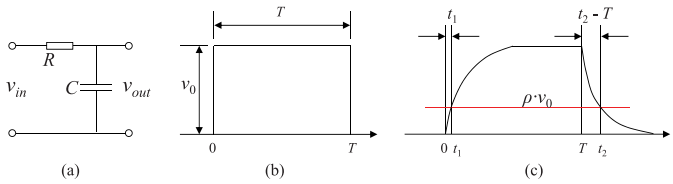

Fig. 2. Low pass filter with RC circuit and its impact on signal envelope. (a) Low pass filter. (b) Original envelope. (c) Envelope after low pass filtering.

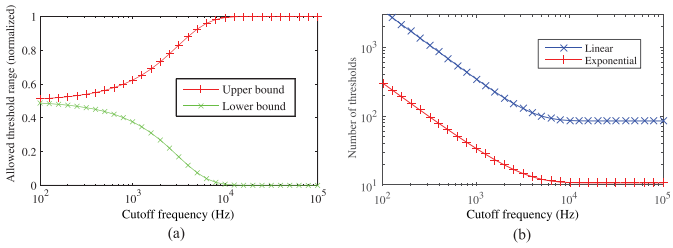

Fig. 3. Effect of cutoff frequency: (a) Allowed range of normalized threshold ρ, (b) number of required thresholds. Conditions: $v_{min} = 20$, $v_{max} = 1700$, $\varepsilon = 80$ μs.

We first consider an ideal envelope shown in Fig. 2(b), which is represented as a pulse

$$x(t) = v_0 \cdot [u(t) - u(t-T)], \tag{1}$$

with an amplitude v_0 and a width T, where $u(t)$ is the unit step function. The envelope, after passing the LPF, becomes

$$y(t) = v_0(1-e^{-t/\tau}) \cdot u(t) - v_0\left(1 - e^{-(t-T)/\tau}\right) \cdot u(t-T). \tag{2}$$

The smoothed envelope is compared against a threshold $\rho \cdot v_0$ $(0 < \rho < 1)$, to measure the frame length. It crosses this threshold at time t_1 and t_2, respectively, which can be computed by

$$v_0(1 - e^{-t_1/\tau}) = \rho \cdot v_0 \rightarrow t_1 = -\tau \cdot \log(1-\rho), \tag{3}$$

$$v_0 \cdot e^{-(t_2-T)/\tau} = \rho \cdot v_0 \rightarrow t_2 = T - \tau \cdot \log \rho. \tag{4}$$

The measured frame length is $t_2 - t_1$, and its deviation from the true value T is

$$d(\rho, \tau) = |t_2 - t_1 - T| = \tau \cdot \left| \log \frac{\rho}{1-\rho} \right|. \tag{5}$$

$d(\rho, \tau)$ decreases as ρ increases from 0 to 0.5, and increases as ρ increases from 0.5 to 1.0. It reaches the minimum 0 when ρ is equal to 0.5, i.e., $0.5v_0$ is the optimal threshold. At the same ρ, $d(\rho, \tau)$ increases with τ (COF decreases). Therefore, there is a tradeoff between noise reduction (low COF is desired) and low distortion in frame length (high COF is desired).

In a real system, there is a maximal allowed error (ε), and $d(\rho, \tau) \leq \varepsilon$ is required for a successful detection of a frame length. Then, according to Eq. (5), ρ should satisfy

$$\frac{1}{1 + \exp(\varepsilon/\tau)} \leq \rho \leq \frac{\exp(\varepsilon/\tau)}{1 + \exp(\varepsilon/\tau)}. \tag{6}$$

The allowed range of ρ, under different COFs, is shown in Fig. 3(a). As COF decreases, the allowed range of ρ also decreases. When a fixed (absolute) threshold is used, ρ almost changes from 0 to 1 when signal strength changes from the highest to the lowest. In such cases, a relatively large COF, e.g., 10 KHz, needs to be selected in order to control the distortion

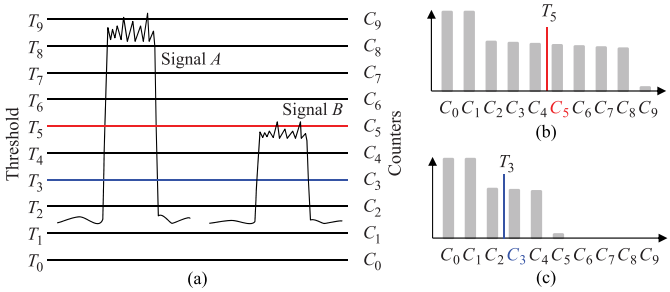

Fig. 4. Detection of frame length by exploiting multiple thresholds. (a) Thresholds and counters. (b) Counters for signal A. (c) Counters for signal B.

in frame length. Alternatively, to improve system performance at low signal strength, a small COF is needed to remove the noise, but the optimal threshold varies with signal strength. A straightforward method to finding the optimal threshold is to store the whole envelope, and then perform frame length detection via post-processing. But this poses a big burden on the storage of the WuRx, and the post-processing of frame length detection also degrades the response of the wake-up control.

III. Multiple Thresholds and Counters

To solve the above problem, we suggest using multiple thresholds $(T_i,\ i = 0, 1, \cdots, N)$ and multiple counters (C_i) to measure possible frame lengths, and finding the optimal threshold and frame length by post-processing. Each counter C_i records the number of consecutive samples whose values are above the threshold T_i. These counters, after normalization, correspond to a complementary cumulative distribution function of envelope samples, e.g., $C_n = Pr(Env \geq T_n)$ and $C_n - C_{n+1} = Pr(T_n \leq Env < T_{n+1})$. Fig. 4 shows the envelopes of signal A and B, with different strengths.

A WuRx periodically switches between two modes, as follows: (i) Measuring noise level mode. The noise level is assumed to be short-time stationary. WuRx clears the counters at the beginning of a measurement period, and counts through the whole period. In Wi-Fi and sensor networks, packets are separated by spaces, during which the WuRx only senses noise. When there is traffic from other nodes, the WuRx will detect envelopes with different levels. At the end of the period, the dominant power levels (with a high percentage) will be found, and the lowest dominant power level (e.g., T_1 in Fig. 4) is regarded as the noise level. (ii) Measuring frame length mode. When a signal comes, counters are increased if the envelope sample is above the corresponding thresholds. When the envelope decreases below a threshold, the corresponding counter is stopped. In the post-processing stage, the difference between adjacent counters is computed and the maximal value corresponds to the range (e.g., $[T_8, T_9)$ for Signal A in Fig. 4) containing v_0. Then, the optimal threshold (e.g., T_5 for Signal A in Fig. 4) is found by the procedure in Section III-A or Section III-B and the corresponding counter value (C_5) is regarded as the frame length.

The number of required thresholds depends on the signal range, and the policy of threshold setting. We first investigate the *linear* threshold policy, and then the *exponential* threshold policy, and compare the number of required thresholds. In the analysis, it is assumed that there is no noise.

A. Linear Thresholds

T_i is equal to $i \cdot \delta$ (δ is a parameter), and the max signal corresponds to $v_{max} = N \cdot \delta$. Then, v_0 is quantized to $n = int(v_0/v_{max} \cdot N)$, indicating that $T_n \le v_0 < T_{n+1}$. The optimal threshold $v_0/2$ lies in the range $[T_n/2, T_{n+1}/2)$.

When n is even, $T_{n/2} = \frac{n\cdot\delta}{2} \le \frac{v_0}{2} < \frac{(n+1)\delta}{2} < T_{n/2+1}$, and $v_0/2$ is nearer to $n \cdot \delta/2$, which is approximated as the optimal threshold for post processing. Then, $\rho = \frac{n\cdot\delta/2}{v_0}$ satisfies

$$\frac{n}{2(n+1)} = \frac{n\delta/2}{(n+1)\delta} < \rho = \frac{n\delta/2}{v_0} \le \frac{n\delta/2}{n\delta} = \frac{1}{2}. \quad (7)$$

As $\rho \le 0.5$,

$$d(\rho, \tau) = \tau \cdot \log \frac{1-\rho}{\rho} \quad (8)$$

is a decreasing function of ρ, and reaches its maximum at $\rho = \frac{n}{2(n+1)}$. Accordingly

$$d(\rho, \tau) \le d\left(\frac{n}{2(n+1)}, \tau\right) = \tau \cdot \log \frac{n+2}{n} \le \varepsilon. \quad (9)$$

Solving this inequality leads to

$$n \ge \frac{2}{\exp(\varepsilon/\tau) - 1}. \quad (10)$$

This should hold for every possible value of n, especially the least value $n = int(v_{min}/v_{max} \cdot N)$. Then, the number of thresholds, N, should satisfy

$$N \ge \frac{2}{\exp(\varepsilon/\tau) - 1} \cdot \frac{v_{max}}{v_{min}}. \quad (11)$$

When n is odd, the analysis is similar, and N should satisfy

$$N \ge \left(\frac{2}{\exp(\varepsilon/\tau) - 1} + 1\right) \cdot \frac{v_{max}}{v_{min}}. \quad (12)$$

With $\rho = \frac{n}{2(n+1)}$, $0.5 - \rho = \frac{1}{2(n+1)}$ decreases at large n, so does the deviation $d(\rho, \tau)$. As a result, the maximal deviation appears at v_{min} (with the smallest n), which determines the number of required thresholds.

B. Exponential Thresholds

To reduce the number of thresholds, we try to enlarge the distance between thresholds at large envelope values. These distances can be increased so that $|\rho - 0.5|$ is as large as that at v_{min}. In this way, the minimal number of thresholds is reached and $\Delta\rho = \rho - 0.5$ is a constant. Let v represent the value of a threshold. $\Delta\rho \cdot v = \Delta v$ is designed to be proportional to v.

$$\Delta v \propto v. \quad (13)$$

Solving this differential equation leads to

$$v(n) = k \cdot \alpha^n, \quad (14)$$

where k and $\alpha > 1$ are system parameters.

In the following, we consider the exponential thresholds, $T_n = \alpha^n$, $\alpha^N = v_{max}$, and assume $T_n \le v_0 < T_{n+1}$. Let $2 = \alpha^f$. Then, the optimal threshold $T_m \approx v_0/2$ satisfies

$$\alpha^{n-f} = \frac{\alpha^n}{2} \le T_m < \frac{\alpha^{n+1}}{2} = \alpha^{n+1-f}. \quad (15)$$

Then, $T_m = \alpha^m$, $m = int(n + 1 - f)$, is the optimal threshold for post-processing. The potential range of ρ can be estimated as follows

$$\frac{1}{2\alpha} = \frac{\alpha^{n-f}}{\alpha^{n+1}} < \rho = \frac{T_m}{v_0} < \frac{\alpha^{n+1-f}}{\alpha^n} = \frac{\alpha}{2}. \quad (16)$$

As ρ must satisfy Eq. (6), we get a conservative estimation of α by

$$\frac{1}{1+\exp(\varepsilon/\tau)} \le \frac{1}{2\alpha}, \frac{\alpha}{2} \le \frac{\exp(\varepsilon/\tau)}{1+\exp(\varepsilon/\tau)}. \quad (17)$$

The maximal value of α, α_{max}, is used to reduce the number of thresholds. And the minimal number of thresholds is

$$N \ge \log_{\alpha_{max}} v_{max}. \quad (18)$$

Fig. 3(b) shows the number of required thresholds under two policies (linear thresholds and exponential thresholds). It is clear that at lower COF, more thresholds are required because the envelope is more distorted. By using the exponential policy, the number of thresholds can be reduced by one order compared with the linear policy.

C. Analysis of Delay and Power Consumption

If the counters are implemented in parallel with special hardware, the scale of logic circuits and the cost will be increased. On the other hand, the counters can be emulated by the MCU iteratively increasing each counter when a signal arrives. We consider the latter in this letter.

To meet the realtime requirement, a node should respond within a short time t_0 (e.g., 1 ms) after it receives a wake-up signal. Assume the envelope of a signal has a length T and the sampling period is t_S. The MCU takes a time t_C to update a counter. With N thresholds, the MCU takes a time $N \cdot t_C$ to update all counters per sample. (i) If $N \cdot t_C$ is less than t_S, counters can be updated in real time. Only the estimation of frame length from N counters is left in the post processing, which can be finished within t_0 for a moderate N. Then, the overall delay of wake-up is $T + t_0$, and the power consumption of the MCU is equal to $(T + t_0) \cdot P_M$, where P_M is the power of the MCU. (ii) If $N \cdot t_C$ is greater than t_S, updating counters is conducted in the post processing stage. The MCU will take a time $T/t_S \cdot N \cdot t_C$ to update all the counters, and need to further find the frame length. The overall delay is approximately $T + T/t_S \cdot N \cdot t_C + t_0$. If the MCU starts working after the signal ends, it consumes a power $(T/t_S \cdot N \cdot t_C + t_0) \cdot P_M$ in the post processing.

Consider a typical case with $t_S = 20$ μs, $t_C = 0.5$ μs, and COF = 1 KHz. The exponential policy has a small number of thresholds $N_{exp} = 34$. $N_{exp} \cdot t_C = 17$ μs is less than t_S and counters are updated in real time. In comparison, the linear policy has a large number of thresholds $N_{line} = 346$, and $N_{line} \cdot t_C = 173$ μs is much larger than t_S. As a result, this policy completely runs in the post-processing stage. Its large delay cannot meet the realtime requirement, and may affect the detection of wake-up signals. Table I compares the delay and power consumption of MCU in the two methods under different frame lengths.

TABLE I
COMPARISON OF DELAY AND POWER CONSUMPTION OF MCU
(DELAY UNIT: ms, POWER UNIT: ms · mW, $t_S = 20\ \mu$s, $t_C = 0.5\ \mu$s, $t_0 = 1$ ms, $N_{exp} = 34$, $N_{line} = 346$, $P_M = 1$ mW)

	Expressions	T=2ms	T=5ms
Exponential (delay)	$T + t_0$	3	6
Linear (delay)	$T+T/t_S \cdot N_{line} \cdot t_C + t_0$	20.3	49.3
Exponential (power)	$(T + t_0) \cdot P_M$	3	6
Linear (power)	$(T/t_S \cdot N_{line} \cdot t_C + t_0) \cdot P_M$	18.3	44.3

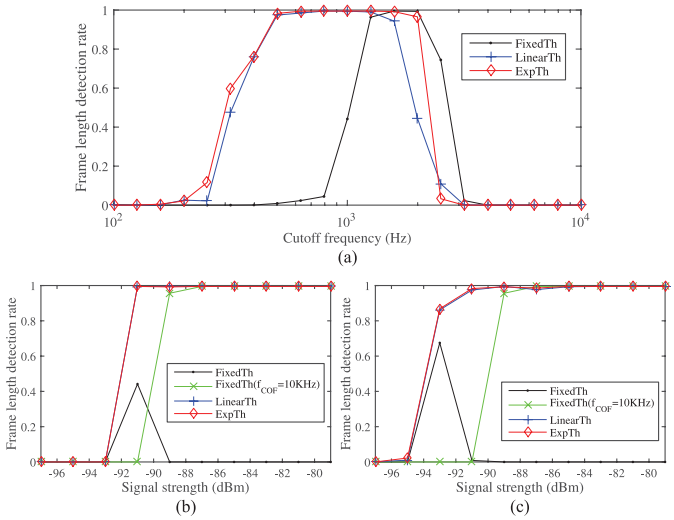

Fig. 5. Frame length detection rate. (a) At different COFs (Signal strength = −91 dBm). (b) At different signal strengths (COF = 1 KHz). (c) At different signal strengths (COF = 500 Hz). The number of thresholds is determined according to Fig. 3(b).

IV. EXPERIMENT RESULTS

We built a simple WuRx for sensor nodes, and did some testbed experiments to evaluate the sensitivity. Interested reader is referred to [8] for more details. IEEE 802.15.4 signal is transmitted by the FSK (frequency-shift keying) modulation, and its frame length is used for the wake-up control of sensor nodes. With this testbed, we obtained envelope samples of signals at different strengths, where the frame length at the transmitter side is equal to 4 ms. The envelope data is processed offline, by using LPF with different COFs. Here, we evaluate (i) One fixed threshold (FixedTh), (ii) Multiple thresholds with equal intervals (LinearTh), and (iii) Multiple thresholds with exponential intervals (ExpTh). The number of thresholds is determined according to Fig. 3(b). The experiments are repeated 20 000 times to compute frame length detection rate. The measured noise level is equal to 760.

Fig. 5(a) shows frame length detection rate at different COFs. A high COF causes a small distortion in frame length but hardly removes noise. A low COF reduces more noise at the cost of large distortion. Therefore, all three schemes work in a certain COF range. Because LinearTh and ExpTh use multiple thresholds to estimate the correct frame length, they are more robust to COF changes and work in a wider COF range than FixedTh. Fig. 5(b) and (c) show frame length detection rate under different signal strengths. Compared with FixedTh (COF = 10 kHz), LinearTh and ExpTh improve the performance at low signal strength, by using a lower COF to remove noise from the envelope. For FixedTh, using a same low COF as in ExpTh, however, is impractical because this will affect the performance at high signal strength. At low signal strength, ExpTh achieves a similar performance as LinearTh, but using much fewer thresholds. Compared with FixedTh (COF = 10 KHz), ExpTh provides a gain of about 2 dB when COF is 1 kHz, and a gain of about 4 dB when COF is 500 Hz at the cost of more thresholds.

V. CONCLUSION

This letter has studied the distortion impact of LPF on frame length detection and suggested using multiple thresholds to find multiple possible frame lengths. Then, via post-processing, the most likely frame length is found. The high accuracy is achieved, which only requires a small number of thresholds. The experiments on a prototype wake-up receiver for sensor nodes confirmed the effectiveness of the proposed scheme. This scheme applies to the wake-up control of Wi-Fi devices as well. Currently, a fixed allowed error in frame length is used. Actually, this depends on the statistical property of background communications. Then, there is a tradeoff between detection accuracy (a large allowed error is desired for a low error rate) and false wake-up (a small allowed error is desired for a low error rate). The study of this tradeoff is left as a future work.

REFERENCES

[1] I. Demirkol, C. Ersoy, and E. Onur, "Wake-up receivers for wireless sensor networks: Benefits and challenges," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 88–96, Aug. 2009.
[2] N. M. Pletcher, S. Gambini, and J. Rabaey, "A 52 uw wake-up receiver with 72 dbm sensitivity using an uncertain-IF architecture," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 269–280, Jan. 2009.
[3] S. Tang, H. Yomo, Y. Kondo, and S. Obana, "Wake-up receiver for Radio-On-Demand wireless LANs," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 42, 2012.
[4] J. Oller, I. Demirkol, J. Casademont, and J. Paradells, "Design, development, and performance evaluation of a low-cost, low-power wake-up radio system for wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 10, no. 1, pp. 1–24, Nov. 2013.
[5] K. Chebrolu and A. Dhekne, "Esense: Communication through energy sensing," in *Proc. ACM MobiCom*, 2009, pp. 85–96.
[6] Y. Kondo *et al.*, "Energy-efficient WLAN with on-demand AP wake-up using IEEE 802.11 frame length modulation," *Comput. Commun.*, vol. 35, no. 14, pp. 1725–1735, Aug. 2012.
[7] S. Tang, H. Yomo, and Y. Takeuchi, "Optimization of frame length modulation-based wake-up control for green WLANs," *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 768–780, Feb. 2015.
[8] S. Tang, H. Yomo, S. Yamaguchi, A. Hasegawa, and S. Obana, "Exploiting frame length of 802.15.4g signals for wake-up control in sensor network," in *Proc. IEEE WCNC*, 2015, pp. 1578–1583.