



Reducing false wake-up in contention-based wake-up control of wireless LANs

Suhua Tang¹ · Sadao Obana¹

© The Author(s) 2018. This article is an open access publication

Abstract

This paper studies the potential problem and performance when tightly integrating a low power wake-up radio (WuR) and a power-hungry wireless LAN (WLAN) module for energy efficient channel access. In this model, a WuR monitors the channel, performs carrier sense, and activates its co-located WLAN module when the channel becomes ready for transmission. Different from previous methods, the node that will be activated is not decided in advance, but decided by distributed contention. Because of the wake-up latency of WLAN modules, multiple nodes may be falsely activated, except the node that will actually transmit. This is called a false wake-up problem and it is solved from three aspects in this work: (i) resetting backoff counter of each node in a way as if it is frozen in a wake-up period, (ii) reducing false wake-up time by immediately putting a WLAN module into sleep once a false wake-up is inferred, and (iii) reducing false wake-up probability by adjusting contention window. Analysis shows that false wake-ups, instead of collisions, become the dominant energy overhead. Extensive simulations confirm that the proposed method (WuR-ESOC) effectively reduces energy overhead, by up to 60% compared with state-of-the-arts, achieving a better tradeoff between throughput and energy consumption.

Keywords Carrier sense multiple access · Wake-up radio · Wake-up latency · False wake-up · Backoff freezing · Early sleep

1 Introduction

To deal with the explosive growth of data traffic in cellular networks, these days most non-realtime traffic is offloaded to wireless local area networks (WLANs) [1]. Accordingly, WLAN modules are embedded into mobile devices such as smartphones and tablets, and widely penetrate into our life. Because a WLAN module is power-hungry while a mobile device is driven by a battery with limited capacity, how to reduce power consumption of a WLAN module becomes an important issue.

In the downlink, a WLAN module expecting to receive messages instantly stays in the constant awake mode (CAM), and the idle waiting leads to much power

consumption. This problem is usually solved from two aspects. One is wake-up scheduling such as power save mode (PSM) [2] and adaptive PSM [3], which is already widely used in mobile devices, although it may cause relatively large delay to real-time messages. The other is asynchronous wake-up by external triggers. In this method, each node uses a low power wake-up radio (WuR) [4–6] to accept remote wake-up request, and activates its power-hungry parts on demand. Recently, the standardization of wake-up control for WLAN module was started as well [7]. In this remote wake-up control [5, 8–12], usually a unique ID is used to identify a specific receiver.

In the up-link, each node needs to monitor the channel before starting its transmission. When many nodes contend for the channel, the idle waiting before actual transmission is non-negligible. For example, when N nodes contend for a channel in the saturation situation, each node is awake all the time but only transmits during $1/N$ of the time. Therefore, it is also necessary to reduce the idle waiting at the transmitter side. In the carrier sense multiple access

✉ Suhua Tang
shtang@uec.ac.jp

¹ Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

(CSMA) method, which node can capture the channel is not decided in advance, but decided by distributed contention. Accordingly, the local wake-up control at the transmitter side also should be contention-based.

Wake-up latency affects the performance of wake-up control. In the remote wake-up control, although a receiver has a wake-up latency and cannot wake up immediately, the transmitter holds the channel before the receiver responds [13]. In the local wake-up control of transmitters, just like the carrier sense function which usually selects only one node to transmit, it is expected that only one node, no more and no less, will be activated. However, due to the wake-up latency, the first node that is to be activated cannot transmit immediately in its wake-up period. Then, WuRs of other nodes detect the channel idleness and falsely decrease their backoff counters. As a result, multiple nodes will be falsely activated.

Previous works mainly use a WuR for the remote wake-up control of receivers [11, 14–16]. In comparison, this work studies the potential problem and performance when tightly integrating a lower-power WuR and a power-hungry WLAN module for the local wake-up control, and suggests (i) using a WuR to perform carrier sense and local wake-up control and (ii) activating a WLAN module only for actual data transmission/reception. The contribution of this paper is two-fold, as follows.

1. To the best of our knowledge, this is the first paper that investigates and analyzes the impact of wake-up latency, considering both the power consumption of WLAN modules in the wake-up period and the power waste due to false wake-ups¹ of non-relevant WLAN modules. On this basis, we *identify false wake-up as a dominant factor of energy overhead* compared with transmission collisions in the multiple access uplink of WLANs.
2. We propose to reduce the impact of false wake-up from three aspects. (i) Recovering the carrier sense mechanism, by resetting the backoff counter of each node in a way as if it is frozen in the wake-up period. (ii) Reducing false wake-up time by inferring false wake-up and immediately putting a falsely activated WLAN module into sleep. (iii) Reducing false wake-up probability by adjusting contention window (CW), which takes a tradeoff between throughput and energy overhead (mainly due to false wake-up) in the optimization process. (i) was reported in a conference

¹ In the literature, usually each time one node is activated by a specific wake-up ID, although the channel errors may lead to false detection of wake-up ID and false wake-ups. In this work, there is no specific wake-up ID. A node that is first activated in the distributed contention is regarded as a true wake-up and subsequent nodes that wake up to find that the channel is already busy are regarded as false wake-ups.

paper [17]. This paper further includes (ii) and (iii), and the method in (ii) allows a more aggressive policy in adjusting CW in (iii) to improve channel efficiency.

Specifically, we show that false wake-up probability increases with wake-up latency. Then, we find the optimal CW to reduce false wake-ups and improve throughput, and suggest a heuristic method to adjust CW, which exploits a WuR to monitor channel contention in an energy efficient way. In the saturation scenario with 10 nodes, when the wake-up latency is equal to 200 μ s, the proposed method (WuR-ESOC) reduces false wake-up probability by 37.2%, and reduces energy overhead (due to both collision and false wake-up) by 62.8%, compared with the method (WuR-MaxEF) using the same optimization but without considering the energy overhead of false wake-up. Extensive simulations lead to the same findings and confirm that the proposed method achieves a better tradeoff between throughput and energy consumption than state-of-the-arts.

In the rest of this paper, Sect. 2 reviews previous methods for collision control and power saving in WLANs. Section 3 first describes system framework, showing how to use a WuR for carrier sense and channel monitoring. Then, we explain the impact of false wake-up due to wake-up latency, introduce the basic solution of resetting backoff counter and point out the necessity of adjusting CW. Section 4 theoretically analyzes the impact of wake-up latency and finds the optimal CW. Simulation evaluation results are analyzed in Sect. 5. Finally, Sect. 6 concludes this paper.

2 Related work

Here, we review collision control, low power listening and asynchronous wake-up control for WLANs.

2.1 Collision control

In the multiple access uplink of a WLAN, nodes contend to transmit to the same AP (access point). The performance heavily depends on the collision probability, which is controlled by the CW. Following the binary exponential backoff policy [2], CW starts from CW_{min} , doubles per transmission failure until it reaches CW_{max} , and resets to CW_{min} per transmission success. Usually CW_{min} and CW_{max} are fixed. This, however, does not work well when the network is congested. Thus, dynamic adjustment of CW is necessary. System throughput depends on the setting of CW [18], and it is possible to adjust CW, based on the number of nodes, to maximize the throughput. System throughput also depends on collision probability, which is correlated with average contention interval [19] and inter-

frame space (IFS). In order to reduce collisions, each WLAN module needs to sense the channel continuously, which wastes much power. In comparison, in this work, a WuR is used to measure in an energy efficient way the length of IFS, based on which CW is adjusted to reduce both collisions and false wake-up events.

2.2 Low power listening

Power waste due to idle listening is a big issue in sensor networks where there is little traffic and each node is almost always idle. Low power listening [20] is a conventional technique for reducing the power waste due to idle waiting, which shifts some overhead from the receiver to the transmitter. The transmitter transmits a long preamble (B-MAC [13]) or periodically repeats the transmission of a short preamble (X-MAC [21]) before each packet, so that a receiver can detect the arrival of this packet. Instead of always being active, a receiver periodically wakes up to sample the channel, and if the energy due to the preamble is detected, remains active to receive the following packet.

A similar power-save mode is adopted in WLANs for the downlink reception. An AP periodically broadcasts a beacon which carries a traffic notification map. Nodes periodically wake up to receive beacons, and if there are packets buffered at the AP, stay awake to receive and otherwise go to sleep immediately. However, duty-cycling [22], which realizes low power consumption by using a long period, usually cannot meet the real-time requirement.

A packet carries a MAC (media access control) header which includes an address specifying the receiver and a duration field indicating when the communication will end. Then, in a further improvement [23, 24], each node at first only decodes the MAC header of a packet, and if this packet is destined to itself, remains active to receive the payload, and otherwise sleeps for a period indicated by the duration field. This avoids the complete decoding of non-relevant packets sent to other nodes.

2.3 Asynchronous wake-up control

Event-triggered sensing requires a real-time response, where a receiver should respond immediately to the transmitter that has detected an event. To this end, very-low power WuR [5] has been extensively studied. Usually it is envelope (energy) detection instead of coherent detection that is used in the WuR to reduce power consumption.

Wake-up control is also applied to WLANs. A low-power WuR, sharing the same channel as its co-located WLAN module, is used to monitor the WLAN channel. In [25], a ZigBee device as a WuR monitors the channel, and

activates its WLAN module if the energy is detected. Without knowing the specific receiver, this usually causes many false wake-ups. A ZigBee module is even used as a secondary radio to obtain the fingerprint of signal strength from nearby APs, and on this basis, provide an energy-efficient background scanning of APs [26].

An elegant method uses a wake-up ID to specify a specific receiver so as to avoid false wake-up. When using a WuR for energy detection of WLAN signals, this can be realized by changing the physical length of a packet [11, 14–16] or applying the on-off-keying modulation on WLAN signals [7]. In the flow-level wake-up control, e.g., an AP is remotely activated by a node initiating a new flow [10, 15], the target (AP) will stay awake until the flow is finished, and then enter the sleep state again. In such cases, the wake-up latency, only at the initial stage, is not a big problem.

Some literature tried to integrate the wake-up control into the MAC protocol, especially for sensor networks [27]. Different from the flow-based wake-up control, here, it is assumed that a node is usually in the sleep state to avoid short-term idle listening, besides long-term idle listening. Then, a wake-up signal is transmitted before each data packet, activating the receiver on demand. The sender node performs carrier sense to avoid potential collision, and if the channel is idle, sends the wake-up signal, which carries the receiver's wake-up ID [28]. The actual data communication is realized by CSMA/CA. The receiving node performs the indicated task according to the received packet and after that enters the sleep state again. A comparison is performed between duty-cycling MAC protocols and the WuR-based MAC protocol, using the parameter of a real wake-up radio [9]. The evaluations under different use cases confirm that the WuR-based MAC protocol reduces power consumption, improves packet delivery rate and reduces delay, compared with duty-cycling based methods.

MAC-level wake-up control can also be included in the WLAN module, if part of its circuit can be used as a low power WuR. An example is illustrated in [8]. A WLAN module in its idle state reduces its clock rate to reduce power consumption. The preamble of each packet is extended so that a node at a low clockrate can also detect the preamble and the wake-up ID inside it, which is similar to B-MAC [13]. If the wake-up ID in the preamble matches its own, a WLAN module recovers its clock-rate to receive the following data packet, using the normal CSMA/CA sequence. This, however, still consumes more power than an optimized WuR, and only applies to remote wake-up control.

2.4 A short comparison

Currently, researchers mainly focus on the power waste due to idle listening at a receiver. In contrast, we investigate the case where multiple nodes contend for the channel and study how to reduce the power consumption due to the carrier sensing (idle waiting) before the actual transmission. In this problem, first, low power listening [20] via duty-cycling is not feasible because the slot for carrier sense is very short. Although overhearing the packet header and going to sleep when a packet is non-relevant help to reduce the power consumption [23, 24], decoding the header of all packets is still expensive. In addition, wake-up latency is not considered in these works.

When activating a WLAN module on-demand, wake-up latency is a big problem. In the remote wake-up control [5, 8–12], the wake-up latency of WLAN modules, although not clearly addressed, can be solved by letting the transmitter hold the channel via a long preamble [13]. But when using a WuR for carrier sensing at the transmitter side, the wake-up is contention-based and the wake-up latency leads to false wake-ups, which has not been studied before. To this end, in this work, we analyze the impact of wake-up latency, considering both the power consumption of WLAN modules in the wake-up period and the power waste due to false wake-ups of non-relevant WLAN modules, and propose to reduce false wake-ups from different aspects. In this way, it is expected that only one node is activated, only for its actual transmission.

3 Tight Integration of WuR and WLAN

WLANs have both control and data planes, but conventionally each WLAN module only has one transceiver. In addition, the power consumption of a transceiver increases with the supported data rate. To facilitate power management, we try to use a low power WuR for the control (both for carrier sense and measuring the IFS for adjusting CW), and use a power-hungry WLAN transceiver only for the high rate data communication. We will discuss the impact of the wake-up latency of a WLAN module and present its solutions.

3.1 System framework

Figure 1 shows the system model. Conventionally (Fig. 1a), a WLAN module consists of a power-hungry RF transceiver and a carrier sense function based on the RF receiving. A WLAN module is idle during most of the time, but its carrier sense function (and the RF receiving) runs all the time and consumes much power. In the

proposed framework (Fig. 1b), the carrier sense function of a WLAN module is separated as an independent part, and is implemented by a low power WuR. A WuR shares the same channel as a WLAN module and continuously detects the energy of a WLAN signal, based on which carrier sense is conducted. Compared with a WLAN module, a WuR consumes much less power because it performs energy detection without requiring a stable carrier. Although in concept a WuR and a WLAN module are independent, in the actual implementation, a WuR can be realized as a part of a WLAN module for a better integration [8].

A WuR and a WLAN module interact as shown in Fig. 1(c). When a WLAN module has a packet to send but the channel is not ready yet, it sets the backoff counter (using current CW) and starts its WuR. Carrier sense by a WuR is performed by energy detection. A WuR detects the envelope of a received signal, and compares it with the carrier sense threshold (CS_{th}) per slot (with a length T_S). A WuR decreases its backoff counter by 1 if the channel is idle in a slot, and leaves it unchanged otherwise. When its backoff counter is decreased to 0, a WuR activates its co-located WLAN module. A WLAN module manages the CW based on the number of slots between two successive transmissions estimated by the WuR, besides initializing the backoff counter.

Both a WuR and a WLAN module have a non-zero wake-up latency due to the hardware constraint. A WLAN module, transiting from the sleep state to the fully functional active state, has to wait until its clock and carrier frequency get stable. Its wake-up period has a relatively long time $T_{WU} = N_{WU} \cdot T_S$ (equal to N_{WU} slots), which may be greater than 100 μs , but its transition from the active state to the sleep state can be finished within a short time T_{SL} . A WuR has a small latency,² which usually only occurs at the beginning of a flow. Although the interaction between a WuR and a WLAN module also causes a delay, this delay can be made small by circuit design. Therefore, for the simplicity, in this work we focus on the wake-up latency of a WLAN module and neglect that of a WuR. In the analysis we take two assumptions. (i) Each WLAN module has the same wake-up latency. (ii) A WuR has the same sensitivity as CS_{th} of a WLAN module.

3.2 Impact of wake-up latency

Without a wake-up latency, a WLAN module is expected to transmit immediately when receiving a wake-up instruction from its WuR. But in the presence of the wake-up latency ($T_{WU} = N_{WU} \cdot T_S$), the first WLAN module that

² By a simple experiment which implements a WuR with off-the-shelf amplifier and envelope detector, we confirmed that the latency of a WuR is usually several μs , less than one contention slot.

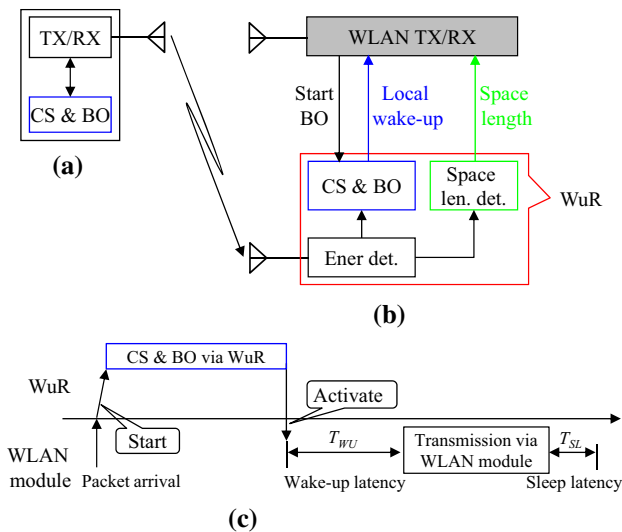


Fig. 1 System model. **a** Conventional WLAN module. **b** Tight integration of a WLAN module and a WuR. **c** Interactions between a WuR and a WLAN module for performing carrier sense, backoff and data transmission

is activated cannot transmit immediately. It is because this WLAN module has to first transit from sleep to the active state. In this wake-up period, the channel, although expected to be busy by the carrier sense mechanism, remains idle. Other WuRs detect the channel idleness and continue the decrease of their backoff counters. Then, multiple backoff counters may reach 0, and their WuRs will activate the co-located WLAN modules. All but the first node are falsely activated.

Figure 2(a) illustrates an example of the multiple access uplink, using a simple WLAN consisting of an AP and three nodes. At first, *A* and *B* set their backoff counters with random initial values 8 and 4, respectively. After the channel is idle for a period of DIFS (Distributed coordination function IFS), WuRs of *A* and *B* decrease their backoff counters per idle slot. WuR *B* first decreases its backoff counter to 0, and activates its WLAN module to transmit a data frame P_{BU} to the AP. When WLAN module of node *B* transits from sleep to the awake state, the channel remains idle. As a result, WuR *A* decreases its backoff counter to 0 per idle slot before node *B* starts transmitting, and activates its WLAN module. Node *A* faces a false wake-up because the channel is already busy due to the transmission from node *B* and stays awake. Next, node *C* contends with node *A* to transmit a frame P_{CU} to the AP, and randomly selects a small value 3 for its backoff counter. Though WLAN module of node *A* selects a relatively large value 6 for its backoff counter, it starts transmission earlier because it is already active and the wake-up latency does not happen. In this round, node *C* faces a false wake-up. Here, the hybrid use of WuRs and WLAN

modules for the backoff control worsens the false wake-up problem.

3.3 Reducing impact of wake-up latency

The impact of wake-up latency is reduced from three aspects, as follows.

- (i) Backoff freezing (BOF) [17]. In the carrier sense mechanism (without wake-up latency), the node whose backoff counter (C_{BO}) first reaches 0 wins the channel and C_{BO} (greater than 0) of other nodes is frozen because the channel gets busy. In this work, the contention-based wake-up control mimics the distributed backoff. Although the value of entering the wake-up period of the first activated node, at that timing the channel is idle and each WuR does not know whether it enters a wake-up period of some node else. Therefore, the detection of false wake-up is postponed.

In this method, it is the WLAN module itself that detects whether it has experienced a false wake-up after it gets fully active. A WLAN module, after being activated, expects the channel to be idle for its own transmission. It performs carrier sense again before the actual transmission, and regards this as a false wake-up if the channel is already busy due to the transmission from other nodes. A falsely activated WLAN module enters the sleep mode again.

C_{BO} is decreased when the channel is really idle (no node is in the wake-up period) or in the wake-up period of a node. As for the latter, the channel is expected to be busy by the carrier sense mechanism. To maintain a consistent C_{BO} , a WuR in the wake-up period continues decreasing its C_{BO} towards minus even after C_{BO} reaches 0, until the channel gets busy again. Because C_{BO} is falsely decreased by N_{WU} in a wake-up period, a node resets its C_{BO} as $C_{BO} + N_{WU}$. In this way, it looks as if C_{BO} is frozen, not changed in the wake-up period.

Figure 2(b) illustrates how this works. C_{BO} of WuR *A* is decreased to 6 when C_{BO} of WuR *B* is decreased 0. Because WLAN module of node *B* does not start transmission immediately, WuR *A* continues decreasing its C_{BO} until the channel gets busy. At this time, C_{BO} of WuR *A* has value 1 and was already falsely decreased by N_{WU} . Therefore, it is reset to $1 + N_{WU} = 6$. After node *B* finishes its transmission, WuR *C* also contends for the channel. C_{BO} of node *C* reaches 0 first. In this wake-up period, WuR *A* decreases its C_{BO} to -2 until the channel gets busy again. Therefore, WuR *A* resets its C_{BO} as $-2 +$

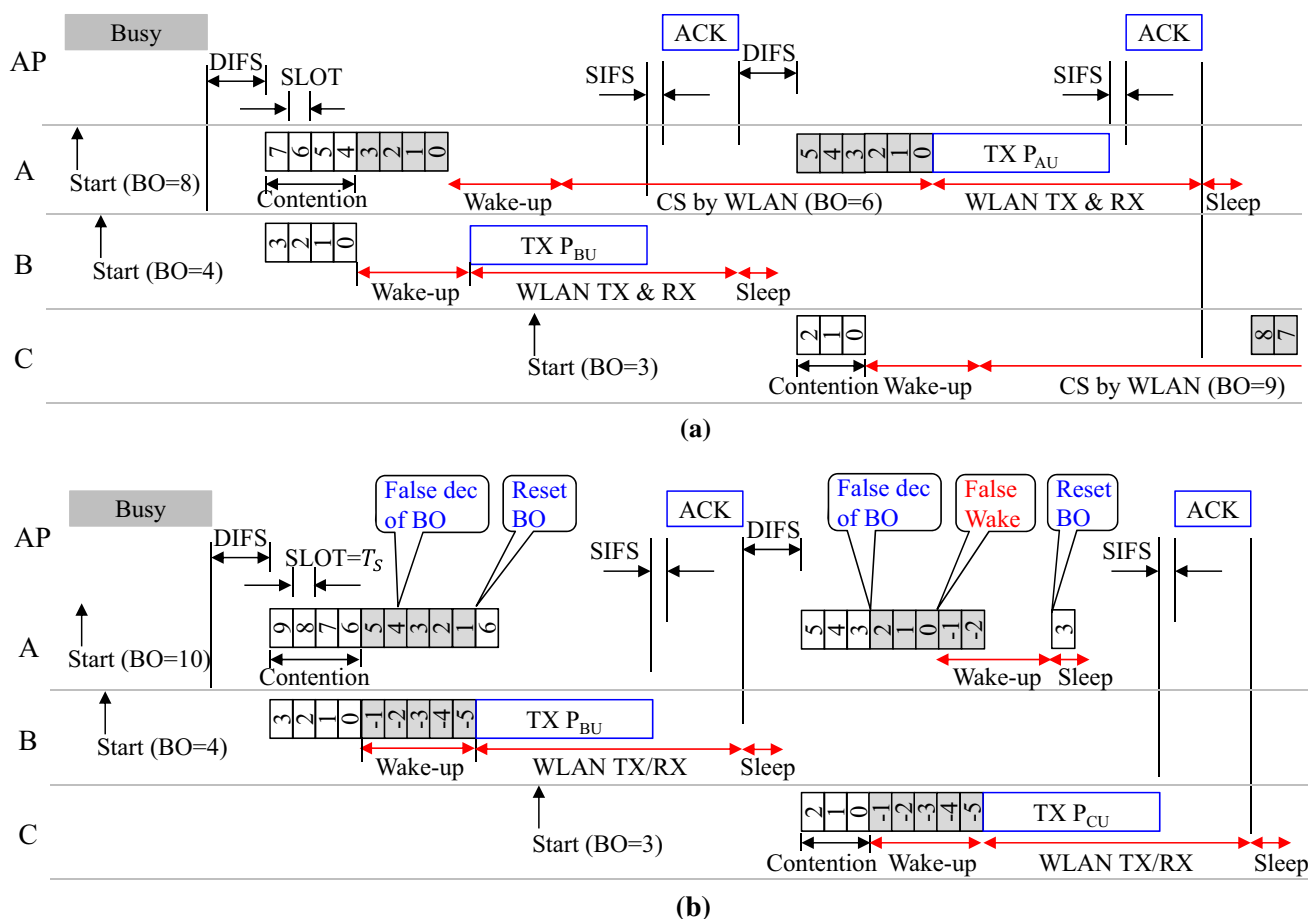


Fig. 2 Example of uplink transmissions in a WLAN with an AP and three nodes *A*, *B* and *C*. **a** The impact of wake-up latency causes multiple nodes with different backoff counters to be activated.

b Simple solution by resetting backoff counters of falsely activated nodes. Wake-up latency is $N_{WU} = 5$ slots

$N_{WU} = 3$ to remove the impact of wake-up latency. Node *A* has to wait until node *C* finishes its transmission, but putting its WLAN module into sleep helps to reduce its power consumption.

- (ii) Reducing false wake-up time. In the BOF method, a falsely activated WLAN module wakes up completely, detecting the channel is busy and entering sleep again (node *A* in Fig. 3a). During most time of the wake-up period, a WLAN module is waiting for its clock and carrier frequency to become stable. Here, a WuR monitors the channel even after its co-located WLAN module is activated, and detects whether it has generated a false wake-up. This detection is triggered by the channel getting busy at the end of the wake-up period of the first activated node, before a falsely activated WLAN module gets fully active.

When a WuR decreases its C_{BO} to 0, it activates its co-located WLAN, with a risk of false wake-up. This gives transmission chance to all nodes in

a distributed way, although there is no guarantee that an activated node can transmit on the channel. A WuR expects its WLAN module to start transmission, N_{WU} slots after the activation, and at that timing its C_{BO} gets equal to $-N_{WU}$. When the channel becomes busy again but its C_{BO} is greater than $-N_{WU}$, it implies that C_{BO} is greater than 0 at the timing of entering the wake-up period of the first activated node. Then, the WuR regards this as a false wake-up and puts its WLAN module into sleep immediately (Fig. 3b), which is called early sleep (ES). The backoff counter is reset in the same way as in (i). If the backoff counter is C_{BO} just before entering the wake-up time, the wake-up time can be reduced from T_{WU} to $T_{WU} - C_{BO} \cdot T_S$. The larger C_{BO} is, the more ES helps to reduce false wake-up time.

- (iii) Reducing false wake-up probability. To completely remove false wake-up events, it is expected that the difference between backoff counters of any two nodes is greater than N_{WU} .

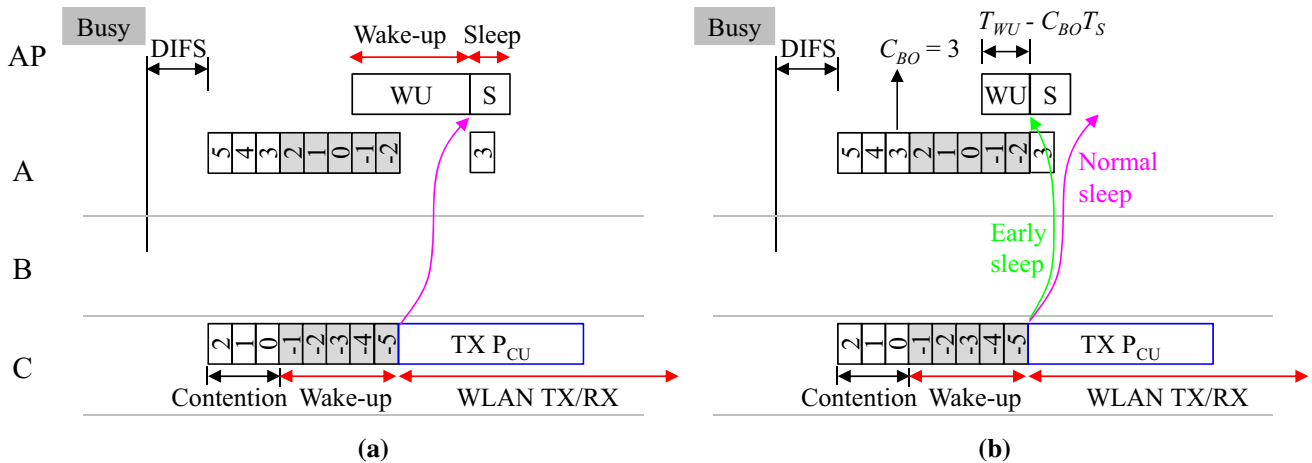


Fig. 3 Sleep control of a WLAN module. **a** Normal sleep vesus. **b** Early sleep. Wake-up latency is $N_{WU} = 5$ slots

In this way, at any timing when a node wakes up, no other nodes will be falsely activated. This can be realized by increasing CW. But too large a CW will degrade channel efficiency. Therefore, a tradeoff should be taken between energy consumption and spectral efficiency.

4 Theoretical analysis

Here, we analyze the energy overhead due to false wake-up and identify its impact compared with the overhead caused by transmission collisions. In the analysis, we consider a WLAN with an AP and N nodes, and assume there are M backoff stages with $CW_{min} = W$, $CW_{max} = 2^M W$, and $CW_i = 2^i W$ for the i th backoff stage. As discussed in Sect. 3.1 and Fig. 1, a WLAN module has a wake-up latency $T_{WU} = N_{WU} \cdot T_S$ and a sleep latency T_{SL} , during which the WLAN module cannot transceive but consumes power. The latency of a WuR is neglected. Main notations are listed in Table 1.

The actual value of T_{WU} varies with devices, and mainly depends on the performance of high frequency oscillator. According to [8], it takes $139 \mu s$ to switch from 1/4 clockrate to full clockrate, and takes about $200 \mu s$ to generate a stable carrier frequency within 50 kHz of the desired value in MAX7032.³ In the following analysis, unless stated otherwise, the wake-up latency is assumed to be $T_{WU} = 200 \mu s$, approximately $N_{WU} = 22$ slots with $T_S = 9 \mu s$.

Table 1 Main notations used in the performance analysis

T_S	Time of a contention slot
T_X	Time of a DATA transmission, including ACK, DIFS, SIFS
L	Packet length (16,000 bits) under frame aggregation
T_C	Transmission time under a collision
T_{WU}	Time taken for a WLAN module to wake up completely
T_{SL}	Time taken for a WLAN module to go to the sleep state
N_{WU}	$N_{WU} = T_{WU}/T_S$, wake-up period in terms of slots
E_I	Power of a WLAN module in the idle period (1 Watt)
E_T	Power of a WLAN module in the transmission (1 Watt)
C_N^k	Number of k -combinations out of N elements

4.1 Basic analysis

By using the backoff freezing method in the wake-up period, the classical Bianchi’s Markov model [29] can be applied here. In the saturation scenario, the probability τ that a node transmits in a randomly chosen slot and the collision probability p under the condition of a transmission can be solved by the following equations.

$$\tau = \frac{2(1 - 2p)}{(1 - 2p)(W + 1) + pW(1 - (2p)^M)}, \tag{1}$$

$$p = 1 - (1 - \tau)^{N-1}. \tag{2}$$

Then, $b_{i,k}$, the probability that a node is at the i th backoff stage with a backoff counter value k , can be computed accordingly from τ and p [29].

Considering all possible backoff stages, the probability that a backoff counter has a value k is equal to

$$B_k = \sum_{i=0}^M b_{i,k}. \tag{3}$$

³ <https://www.maximintegrated.com/en/products/comms/wireless-rf/MAX7032.html/>.

B_k determines the false wake-up probability, and B_0 is equal to the transmission probability τ .

In the slot-based contention, there are three cases at each slot.

- (1) The slot is idle with a probability

$$P_I = (1 - B_0)^N. \quad (4)$$

- (2) Only one node starts a transmission and there is no collision. This probability is equal to

$$P_S = C_N^1 B_0 \cdot (1 - B_0)^{N-1}. \quad (5)$$

The channel occupation time is $T_{WU} + T_X$ for this transmission, but the node will take another time T_{SL} to go to sleep again. At the time a node's backoff counter first reaches 0, other nodes, whose backoff counters are less than N_{WU} , will be falsely activated. The probability that m nodes are falsely activated is equal to

$$P_F(m|S) = C_N^1 B_0 \cdot C_{N-1}^m \left(\sum_{i=1}^{N_{WU}} B_i \right)^m \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-1-m}. \quad (6)$$

And the overall active time of a false wake-up (without early sleep) is $T_{WU} + T_{SL}$.

- (3) $k \geq 2$ nodes transmit simultaneously and a collision occurs, with a probability

$$P_C(k) = C_N^k B_0^k \cdot (1 - B_0)^{N-k}. \quad (7)$$

The channel occupation time is $T_{WU} + T_C$ for this collision. During the wake-up period, m nodes are falsely activated and go to sleep again, with a probability

$$P_F(k, m|C) = C_N^k B_0^k \cdot C_{N-k}^m \left(\sum_{i=1}^{N_{WU}} B_i \right)^m \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-k-m}. \quad (8)$$

4.2 Impact of false wake-up per transmission

The transmission probability per slot is $P_{tr} = 1 - P_I$. Nodes that transmit per contention round in the conventional CSMA are regarded as normal wake-up, and their numbers as successful transmissions (N_S) and collisions (N_C) are computed as

$$N_S = 1 \cdot P_S / P_{tr}, \quad N_C = \sum_{k=2}^N (k \cdot P_C(k)) / P_{tr}. \quad (9)$$

Other nodes that get active due to the wake-up latency are regarded as false wake-up and their number is equal to

$$N_F = \left[\sum_{m=1}^{N-1} m \cdot P_F(m|S) + \sum_{k=2}^N \left(\sum_{m=1}^{N-k} m \cdot P_F(k, m|C) \right) \right] / P_{tr}. \quad (10)$$

On average, total energy consumption of WLAN modules per contention round is composed of three parts: successful transmission ($E_S(W)$), transmission collisions ($E_C(W)$), and false wake-up ($E_F(W)$, without early sleep).

$$E_S(W) = ((T_{WU} + T_{SL}) \cdot E_I + T_X \cdot E_T) \cdot N_S, \quad (11)$$

$$E_C(W) = ((T_{WU} + T_{SL}) \cdot E_I + T_C \cdot E_T) \cdot N_C, \quad (12)$$

$$E_F(W) = (T_{WU} + T_{SL}) \cdot E_I \cdot N_F. \quad (13)$$

Numbers of successful transmissions (N_S), collisions (N_C) and false wake-ups (N_F) are shown in Fig. 4(a), under different values of CW_{min} . With a fixed number of nodes and a fixed wake-up latency, increasing CW_{min} effectively reduces the number of collisions and the number of false wake-ups. An interesting fact here is that N_F is usually much greater than N_C . Actually, the ratio of N_F to N_C increases with CW_{min} and approaches a steady value, which is nothing but N_{WU} (refer to the appendix for more details). This is because at large CW , B_k is almost the same when $k < N_{WU}$ and the wake-up period is N_{WU} times of a collision slot. N_{WU} is typically much greater than 1, which infers that *there are many more false wake-ups than collisions*.

When early sleep is used, the wake-up time of a falsely activated node is reduced from T_{WU} to $T_{WU} - C_{BO} \cdot T_S$, where C_{BO} is the backoff counter value just before the node enters the wake-up period, as shown in Fig. 3(b). C_{BO} can be between 1 and N_{WU} . Let the C_{BO} be k (with a probability B_k). Then, the number of false wake-up slots is equal to $N_{WU} - k$. On average, the number of false wake-up slots is equal to

$$N_{ES} = \sum_{k=1}^{N_{WU}} B_k \cdot (N_{WU} - k) / \sum_{k=1}^{N_{WU}} B_k. \quad (14)$$

Then, a factor, $\alpha_{ES} = (N_{ES} \cdot T_S + T_{SL}) / (T_{WU} + T_{SL})$, should be multiplied to the right side of Eq. (13) when early sleep is used.

4.3 Optimal CW

It should be noted that increasing CW affects the throughput and channel efficiency because it may lead to more idle slots at a larger CW_{min} . Therefore, there is a tradeoff between throughput and energy overhead (due to collision and false wake-up).

Average throughput is computed via Eq. (15) as a ratio of the number of transmitted bits to the average slot length (T_{AVE}),

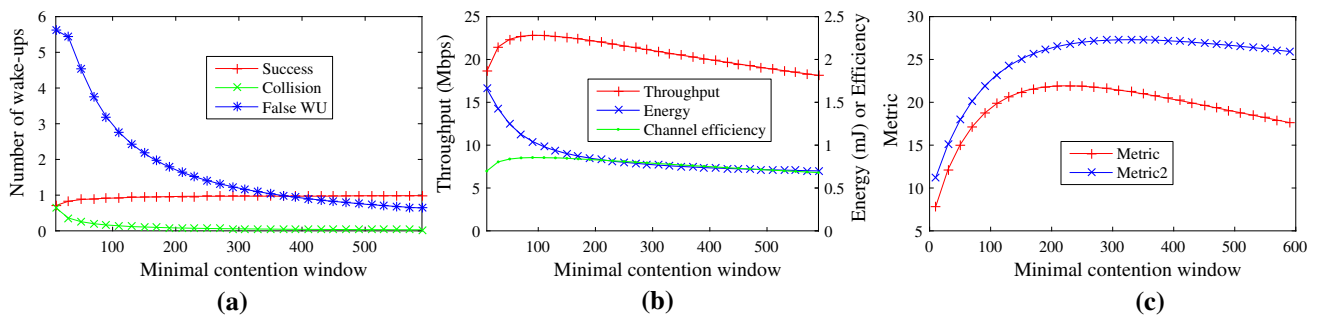


Fig. 4 Performance under different CW_{min} . Saturation scenario; wake-up latency is $N_{WU} = 22$ slots; $N = 10$ nodes. **a** Number of wake-ups. **b** Throughput, energy and channel efficiency. **c** Different metrics

$$\Gamma(W) = \frac{P_S \cdot L}{T_{AVE}},$$

$$T_{AVE} = P_I \cdot T_S + P_S \cdot (T_{WU} + T_X) + P_C \cdot (T_{WU} + T_C), \tag{15}$$

in a similar way as [29], but involves the effect of wake-up latency, where L is packet length in bits.

Spectral energy efficiency $\xi(W)$, defined as the ratio of the number of transmitted bits to the time and the energy consumed in the transmission, is equal to

$$\xi(W) = \frac{\Gamma(W)}{E_S(W) + E_C(W) + E_F(W)}. \tag{16}$$

Channel efficiency $\eta(W)$ is the percentage of time that a channel is actually used.

$$\eta(W) = \frac{P_S \cdot (T_{WU} + T_X)}{T_{AVE}}. \tag{17}$$

CW_{min} should be adjusted in such a way that throughput and channel efficiency are improved (reducing collision and idle slots), but energy consumption is reduced (reducing collision and false wake-up). But according to Fig. 4(b), the peak of throughput ($\Gamma(W)$) appears in the range where energy consumption ($E_S(W) + E_C(W) + E_F(W)$) is relatively high. Then, throughput decreases, but at small CW_{min} , energy consumption decreases faster than throughput. As a result, $\xi(W)$ is maximized at a relatively large CW_{min} . To better stress spectral efficiency, $\eta(W)$ is used together with $\xi(W)$ to find the optimal CW_{min} by the following equation

$$W = \underset{w}{\operatorname{argmax}} \xi(W) \cdot \eta(W). \tag{18}$$

$\xi(W) \cdot \eta(W)$ and $\xi(W)$ are shown as Metric and Metric2, respectively, in Fig. 4(c) under different CW_{min} . Here, by considering channel efficiency, Metric reaches its peak at smaller CW_{min} , where higher throughput can be achieved.

4.4 Comparing different methods

By changing the wake-up latency T_{WU} and finding corresponding optimal CW_{min} , the impact of wake-up latency and the effect of CW adjustment in different methods are investigated. Here, the backoff freezing method (BOF) is regarded as a base. On this basis, functions such as ES (early sleep), OC (optimal CW by applying Eq. (18)) or their combination (ESOC) are studied. MaxTh which finds CW by maximizing the throughput ($\Gamma(W)$) in Eq. (15) and MaxEF which finds CW by maximizing Eq. (18) but without considering the false wake-up energy (i.e. $E_F = 0$) are used for comparison.

Figure 5(a) shows the number of wake-up per transmission under different methods, where S , C and F correspond to N_S , N_C and N_F , respectively. N_C is relatively large in BOF and ES without adjusting CW, but becomes almost negligible when CW is adjusted in the other 4 methods. N_F quickly increases with N_{WU} . Adjusting CW helps to reduce N_F to different degrees. Without considering the energy due to false wake-up in MaxTh and MaxEF, N_F is still relatively large. In comparison, N_F is greatly reduced in OC and ESOC. False wake-up probability, conditioned on a transmission, is computed as N_F/N . This value is 0.540 in BOF, 0.238 in MaxEF, 0.123 in OC, and 0.149 in ESOC, when N_{WU} is equal to 20. ESOC reduces false wake-up probability by 72.4% and 37.2% compared with BOF and MaxEF, respectively. By adopting the early sleep policy, ESOC takes a more aggressive policy than OC and allows more false wake-ups.

Figure 5(b) compares energy consumption per transmission under different methods, where S , C and F correspond to E_S , E_C and E_F , respectively. Here, energy consumption of WuR, being the same in all methods, is not included. Unsurprisingly, overall energy increases with N_{WU} and false wake-up becomes the dominant energy overhead when CW is adjusted in OC, ESOC, MaxTh and MaxEF. MaxEF is a little better than MaxTh by considering the energy due to collisions. But the impact of collision is almost negligible. Therefore, their difference is not

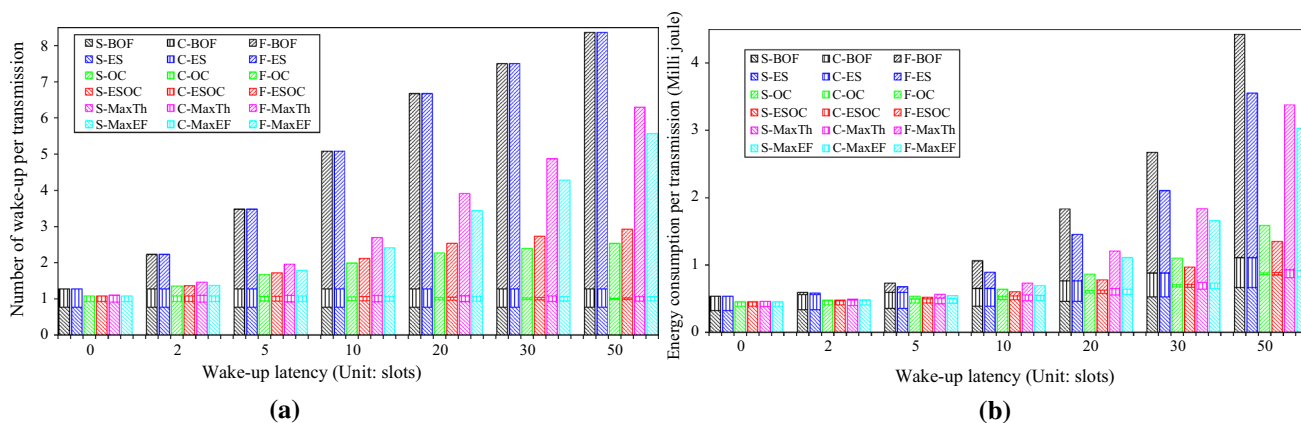


Fig. 5 Impact of wake-up latency under different wake-up latencies. Saturation scenario; $N = 10$ nodes; Energy consumption due to WuR is not considered. **a** Number of successful transmissions, collisions

and false wake-ups. **b** Energy due to successful transmission, collisions and false wake-ups

large. In contrast, energy overhead due to false wake-up is greatly reduced in OC and ESOC, and ES helps to further reduce energy consumption by letting nodes sleep early. When N_{WU} is equal to 20, ESOC reduces energy overhead (due to collision and false wake-up) by 85.2%, 62.8%, 28.0% compared with BOF, MaxEF, OC, respectively.

Spectral energy efficiency ξ is what we care most. Figure 6 shows spectral energy efficiency under different numbers of nodes (without considering energy consumption of WuR). As the number of nodes increases, spectral energy efficiency decreases in BOF and ES because of more collisions. Adjusting CW to reduce both collisions and false wake-ups, spectral energy efficiency in the other four methods approaches steady values. OC and ESOC outperform MaxTh and MaxEF, with ESOC being the best.

Table 2 shows the values of optimal $CW_{min}(W)$ and energy overhead per transmission ($E_C + E_F$) of BOF, ESOC and MaxEF. Compared with MaxEF, ESOC uses larger CW. Although this degrades the throughput of ESOC a little (the difference is approximately 1Mbps), it

helps to reduce the energy overhead greatly by over 60%. This explains why ESOC outperforms MaxEF in spectral energy efficiency in Fig. 6.

It should be noted that a large W may also cause large transmission delay. Average transmission delay (T_D) is equal to the product of the expected number of transmission slots (N_{AVE}) and the average slot length (T_{AVE}).

$$T_D = T_{AVE} \cdot N_{AVE}. \tag{19}$$

Using $1/\tau$ to reflect the number of slots required for a transmission without collision, and using $1/(1-p)$ to account for the number of transmissions under collisions, N_{AVE} can be computed as [30]

$$N_{AVE} = \frac{1}{\tau} \cdot \frac{1}{1-p} = \frac{(1-2p) \cdot (W+1) + pW \cdot (1-(2p)^M)}{2(1-2p)(1-p)}. \tag{20}$$

The delay results are also shown in Table 2. Although ESOC has a larger average slot length (T_{AVE}) than BOF due to a larger contention window, it reduces the collision probability and N_{AVE} effectively. As a result, its delay is even less than that of BOF in the saturation state.

4.5 Heuristics for adjusting CW

The analysis in Sect. 4.3 gives the optimal CW under the saturation scenario. Let $f_X(x)$ represent the distribution B_X of backoff counters ($X = x$) under the saturation scenario, and $F_X(x)$ be its cumulative distribution. Then, the minimal value $X_{min} = \min(X_i)$ of backoff counters of all nodes, follows the distribution

$$f_{X_{min}}(x) = (1 - F_X(x-1))^N - (1 - F_X(x))^N, \tag{21}$$

according to the order statistics [31]. X_{min} decides the number of idle slots, and its expectation $E(X_{min})$ represents

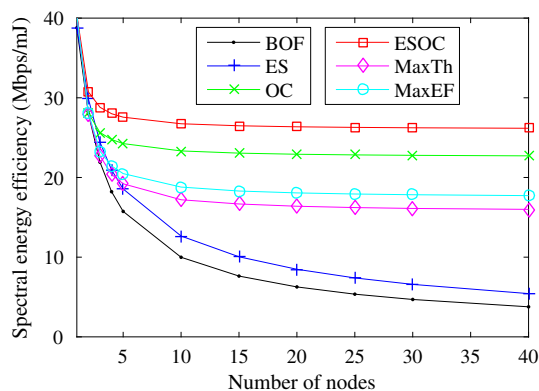


Fig. 6 Spectral energy efficiency under different numbers of nodes. Saturation scenario; wake-up latency is $N_{WU} = 22$ slots; Energy consumption due to WuR is not considered

Table 2 Optimal CW under different numbers of nodes and the corresponding energy overhead and delay

N	BOF			ESOC			MaxEF		
	W	$E_C + E_F$ (Millijoule)	T_D (ms)	W	$E_C + E_F$ (Millijoule)	T_D (ms)	W	$E_C + E_F$ (Millijoule)	T_D (ms)
5	16	0.868	3.6	103	0.209	3.6	58	0.540	3.5
10	16	1.527	8.0	225	0.222	7.3	122	0.631	7.0
15	16	2.013	12.8	347	0.226	11.0	186	0.661	10.6
20	16	2.418	18.0	468	0.229	14.7	250	0.676	14.1
25	16	2.775	23.5	590	0.230	18.4	314	0.685	17.7
30	16	3.099	29.4	711	0.231	22.1	378	0.690	21.2
40	16	3.683	42.0	954	0.232	29.6	506	0.698	28.3

the average number of idle slots per channel contention. These values are very small compared with CW_{min} .

Usually the network runs in a non-saturation case. Here we consider a heuristic algorithm for adjusting CW under the non-saturation case, where less traffic is generated. When the same CW is used as in the saturation scenario, more idle slots will be detected, and CW needs to be decreased in order to improve channel efficiency. The number of consecutive idle slots per contention round can be measured by using a WuR and is denoted as I_{avg} after removing the impact of wake-up latency (N_{WU}) and taking moving average. Then, we adjust $W = CW_{min}$ as follows: increasing W by W_{delta} if I_{avg} is less than $E(X_{min}) - \delta$ and decreasing W by W_{delta} if I_{avg} is greater than $E(X_{min}) + \delta$.

$$W \Leftarrow \begin{cases} W + W_{delta}, & I_{avg} < E(X_{min}) - \delta \\ W - W_{delta}, & I_{avg} > E(X_{min}) + \delta \end{cases} \quad (22)$$

In this way, the number of idle slots per contention, on average, is maintained around $E(X_{min})$.

5 Simulation evaluation

The proposed method is evaluated by network simulation. In the evaluation, the following methods will be compared: (i) CSMA without using WuRs, (ii) WuR-CS [17], using WuRs for carrier sense but without any efforts on solving the false wake-up problem, (iii) WuR-BOF [17], putting a falsely activated WLAN module into sleep and resetting its backoff counter to a proper value after the wake-up period, (iv) WuR-OC, based on WuR-BOF and adjusting CW (using the heuristic method in Sect. 4.5) by monitoring the number of consecutive idle slots via WuRs, (v) WuR-ESOC, based on WuR-OC and is enhanced with the early sleep function, (vi) WuR-MaxEF, similar to WuR-OC except that energy overhead due to false wake-up is not considered, (vii) WuR-CF, a combination of WuR with the contention free method. For a fair comparison, a WLAN

module in CSMA is also put into sleep when there is no packet ready for transmission.

An event-based packet level simulator is built in the Matlab environment. Simulation results are computed as the average of 50 runs. Main simulation parameters are shown in Table 3. In the evaluation, we take IEEE 802.11a as an example, but the proposed method applies to other protocols such as IEEE 802.11n.

A scenario with an AP and a variable number of nodes contending for the uplink transmission is used in the evaluation. We investigate system throughput, energy per packet, spectral energy efficiency, and duty ratio per node (the percentage of active time per WLAN module). Different from the analysis, here energy per packet involves the impact of retransmissions.

5.1 Impact of wake-up latency

First, we investigate the impact of wake-up latency, using a saturation scenario where the number of nodes is fixed to 10. Figure 7(a) shows different components of energy per packet when a WuR is used, where S , C , and F correspond to success, collision, false wake-up, respectively. Energy per packet in WuR-CS increases quickly when the wake-up latency is small, which confirms the necessity of reducing false wake-ups. But it then approaches a constant value, which is a little counter-intuitive. This can be explained as follows: Falsely activated nodes stay awake in WuR-CS, waiting to transmit; Wake-up latency increasing within a certain range decreases the waiting time, and the overall false active time does not change greatly. But a further increase in wake-up latency will lead to the increase of energy per packet. In WuR-BOF, a falsely activated WLAN module is put into sleep again. When N_{WU} is small, this works relatively well, although the wake-up overhead increases with N_{WU} . At a large N_{WU} (e.g., $N_{WU} = 50$) which is much greater than default CW_{min} , in each wake-up period all nodes are activated. As a result, energy per

Table 3 Main parameters for simulation

PHY	IEEE 802.11a, propagation: free-space & two-ray
MAC	CSMA, $T_{SLOT} = 9 \mu s$, $T_{SIFS} = 16 \mu s$, $T_{DIFS} = 34 \mu s$
Packet	$L = 16,000$ bits under frame aggregation
Traffic	Packet arrival time follows Poisson process
Bit rate	Based on SNR (signal to noise ratio)
CW	$W_{delta} = 5$, $\delta = 2$
Latency	Wake-up latency: $22 \times T_{SLOT}$, sleep: $2 \times T_{SLOT}$
Power	WLAN: receiving/idle 1 W, transmission 1 W; WuR: 10 mW

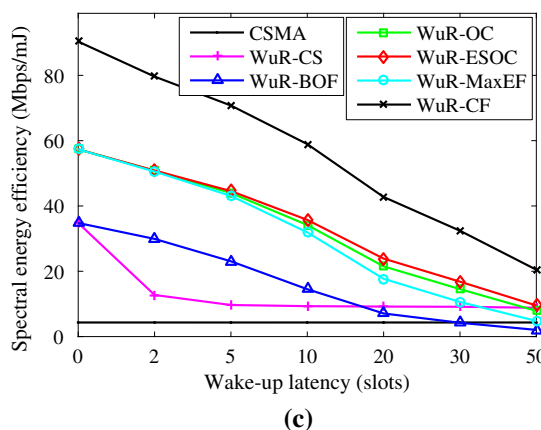
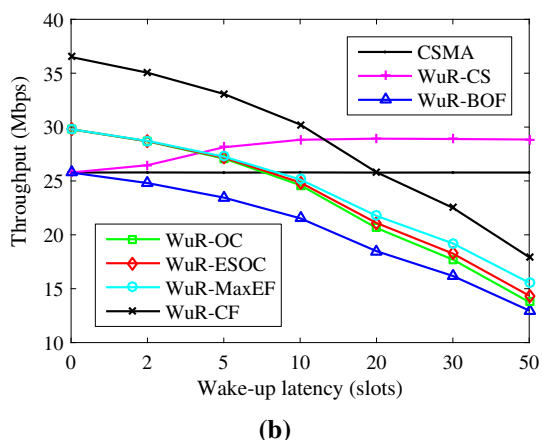
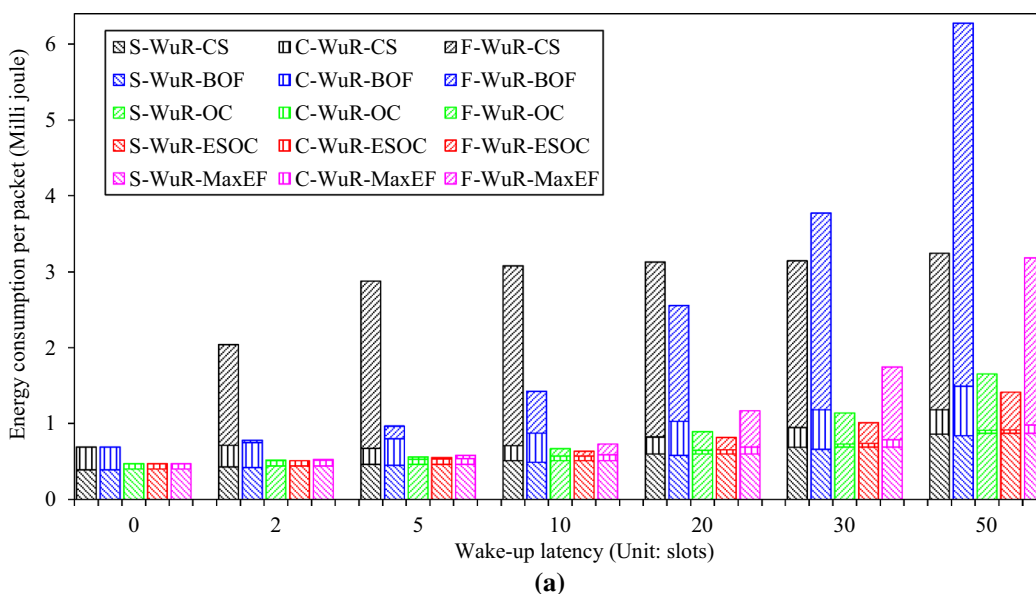


Fig. 7 Performance under different wake-up latencies. Saturation scenario; $N = 10$ nodes. **a** Energy due to successful transmission, collisions and false wake-ups. Energy consumption due to WuR is not considered. **b** Overall throughput. **c** Spectral energy efficiency

packet in WuR-BOF might even get greater than that of WuR-CS. WuR-OC effectively reduces false wake-up events by adjusting CW, and WuR-ESOC further reduces false wake-up time, and their energy overhead due to false wake-up can be kept very low when N_{WU} is no more than 20. WuR-MaxEF reduces collisions very well by adjusting

CW, but the energy overhead due to false wake-up is much larger than that in WuR-ESOC because energy overhead of false wake-up is not considered in the optimization. When $N_{WU} = 20$, WuR-ESOC reduces energy overhead (collision and false wake-up) by 85.0%, 62.4%, 26.8% compared with WuR-BOF, WuR-MaxEF, and WuR-OC,

respectively. This is almost consistent with the analysis result in Sect. 4.4.

When the wake-up latency is relatively large, per-packet sleep/wake-up causes much overhead and degrades the performance of wake-up control. It is possible to further reduce per-packet overhead, e.g., by introducing frame aggregation to enable longer idle time for sleep [32].

Figure 7(b) shows throughput of different methods. Throughput of CSMA is irrelevant of N_{WU} because each node is almost always active, and throughput of other methods except WuR-CS decreases as N_{WU} increases. Throughput of WuR-CS even increases a little with N_{WU} at a small N_{WU} . This is because the wake-up latency leads to false wake-up in WuR-CS. Contention mainly occurs among the falsely activated nodes, which both reduces the impact of wake-up latency and controls the collision degree, although at the cost of more active time. When N_{WU} further increases, severe collision among almost all nodes also degrades the throughput of WuR-CS. Throughput of WuR-MaxEF is a little greater than that of WuR-ESOC.

Combining the results in Fig. 7(a) and (b) leads to spectral energy efficiency in Fig. 7(c). Because throughput decreases while energy per packet increases as N_{WU} increases, spectral energy efficiency decreases very fast. At small wake-up latencies, energy overhead is very small and the superiority of WuR-ESOC over WuR-MaxEF in spectral energy efficiency is not very obvious. The gap between WuR-ESOC and other methods increases with the wake-up latency.

5.2 Impact of the number of nodes

Next, we investigate how system performance varies with the number of nodes. Here, the wake-up latency is fixed to $N_{WU} = 22$ slots, and a saturation scenario is used.

Figure 8(a) shows different components of energy per packet when a WuR is used. Without adjusting CW (WuR-CS, WuR-BOF), energy per packet increases almost linearly with the number of nodes. By adjusting CW, both collisions and false wake-ups can be greatly reduced.

Figure 8(b) shows throughput of all methods. Throughput of CSMA decreases as the number of nodes increases, due to more collisions. Throughput of WuR-CS can even be greater than that of WuR-CF, because the wake-up latency leads to false wake-up in WuR-CS and the contention is mainly among the falsely activated nodes without wake-up latency, as explained before. As the number of nodes further increases, the contention among falsely activated nodes also degrades the throughput of WuR-CS. In WuR-BOF, it is always all nodes that contend for the channel, and the curve has a similar trend as in CSMA, but the throughput is lower due to the wake-up

latency. WuR-OC, WuR-ESOC and WuR-MaxEF achieve stable and higher throughput than WuR-BOF by adjusting CW. The gap between WuR-ESOC and WuR-CF is mainly due to the idle contention slots, and is partially due to unresolved collisions in WuR-ESOC.

Figure 8(c) shows spectral energy efficiency of different methods. Except WuR-CF, WuR-ESOC achieves the highest performance, and has a clear superiority over other methods. Its superiority over WuR-OC confirms the effectiveness of the early sleep method. WuR-ESOC far outperforms WuR-MaxEF, which confirms the necessity of considering energy overhead of false wake-up in optimizing CW. As the number of nodes increases, the transmission interval of each node increases, so does the energy consumption of WuR per packet. As a result, this leads to a little decrease in spectral energy efficiency of WuR-ESOC.

Figure 8(d) shows duty ratio per node. Duty ratio per node in CSMA is nearly 100% because each WLAN module always performs carrier sense for the next transmission in the saturation scenario. As the number of nodes increases, this value decreases to nearly 0.5 in WuR-CS and decreases greatly to about 0.2 in WuR-BOF. It decreases to almost 0.025 (the inverse of the number of nodes) in WuR-ESOC, because each WLAN module is only used for actual transmissions. There is an obvious difference between WuR-BOF and WuR-OC/WuR-ESOC, which confirms the effect of dynamic adjustment of CW on reducing false wake-up events.

5.3 Different traffic rates

We also investigated the performance of these methods under different traffic rates. Here, the number of nodes is fixed to 10. The results are shown in Fig. 9, where the horizontal axis is the overall rate of traffic from all nodes. WuR is used in all methods except in CSMA. In times of light traffic (no more than 500 packet/s), spectral energy efficiency increases with the traffic rate, when a WuR is used. This is because the number of bits divided by the transmission time is nearly constant, but an increase in traffic rate reduces the transmission interval, which further reduces the energy consumption of WuR for each transmission. Spectral energy efficiency is higher in CSMA than in other methods, because a WLAN module is put into sleep in CSMA and it has no extra power consumption for WuR. The superiority of WuR-ESOC over CSMA will get more obvious if nodes in CSMA always listen to the channel in the idle state.

As traffic rate further increases, the network starts to saturate, spectral energy efficiency slightly decreases in WuR-ESOC because a large contention window is used, which increases the idle waiting. Spectral energy efficiency greatly degrades in CSMA due to severe collisions and

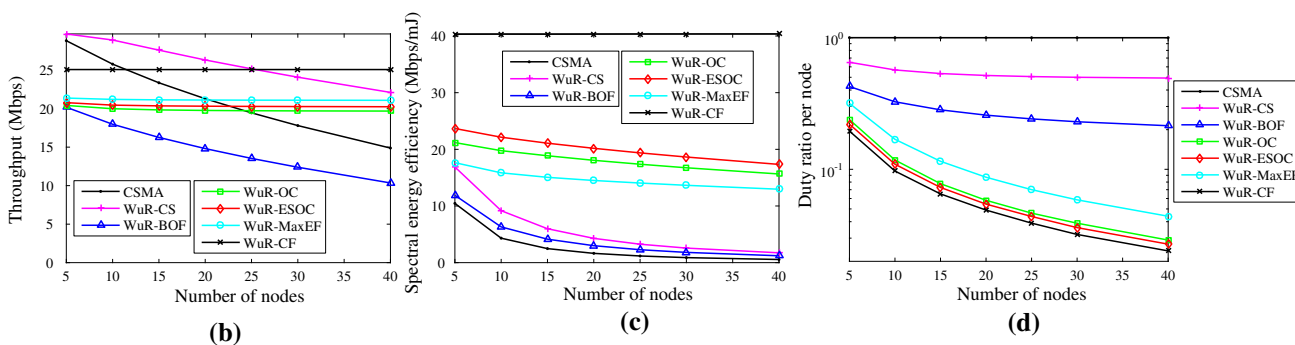
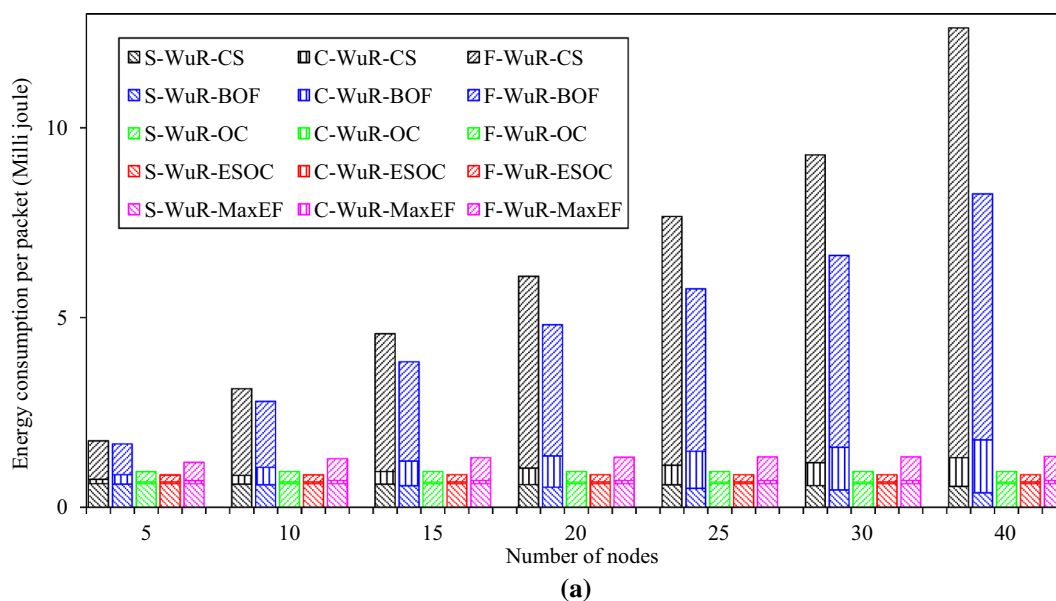


Fig. 8 Performance under different numbers of nodes. Saturation scenario; Wake-up latency is fixed to $N_{WU} = 22$ slots. **a** Energy due to successful transmission, collisions and false wake-ups. Energy

consumption due to WuR is not considered. **b** Total throughput. **c** Spectral energy efficiency (Energy consumption due to WuR is considered). **d** Duty ratio per node

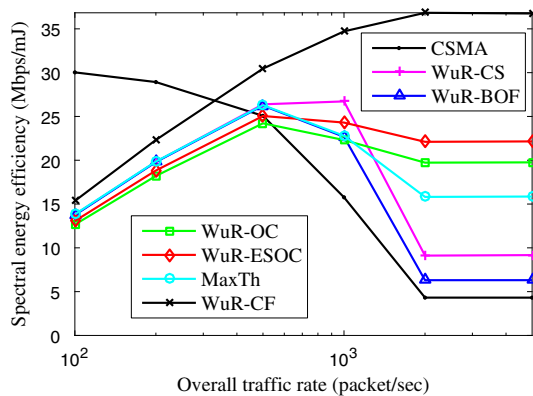


Fig. 9 Spectral energy efficiency under different traffic rates. Wake-up latency is fixed to $N_{WU} = 22$ slots; $N = 10$ nodes

degrades in WuR-BOF, WuR-CS due to both collisions and false wake-up.

According to the results in Figs. 7–9, we give the following remarks: Wake-up control for reducing the power

consumption of WLAN modules in the short-term idleness before actual transmission generally works well at a small wake-up latency. At a relatively large wake-up latency, however, false wake-up is a dominant factor of energy overhead compared with transmission collisions in the multiple access uplink of WLANs. In the proposed method, the adjustment of CW helps to reduce false wake-up events and early sleep helps to reduce false wake-up time. When the wake-up latency is very large compared with packet transmission time, the per-packet sleep/wake-up overhead will be too large and degrade the performance of wake-up control. The per-packet overhead can be reduced, e.g., by introducing frame aggregation to enable longer idle time for sleep [32].

6 Conclusion

To improve energy efficiency of WLANs, we have studied the potential problem and performance when tightly integrating a low power WuR with a power-hungry WLAN module. Specifically, a WuR performs carrier sense and activates its co-located WLAN module when the channel gets ready for actual transmission. We investigated the false wake-up problem caused by the wake-up latency, and found that it was the main energy overhead. We solved this problem from three aspects: (i) recovering the carrier sense mechanism by resetting backoff counters properly, (ii) performing early sleep to reduce false wake-up time by inferring false wake-up, and (iii) reducing the probability of false wake-up by adjusting the contention window. In this way, the impact of wake-up latency is mitigated and the throughput is retained. Theoretical analysis and simulation evaluations confirm that the proposed method effectively reduces energy overhead (mainly caused by false wake-up), by more than 60% compared with the state-of-the-arts, achieving a better tradeoff between spectral and energy efficiency. In the future, we will further study the impact of wake-up latency in the multi-hop transmission.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: The ratio of N_F to N_C

Both N_F and N_C decrease as CW_{min} increases. At a large CW_{min} (W), transmission probability B_0 approaches 0 and the probabilities that are related to false wake-up, B_i , $i \leq N_{WU} \ll W$, are approximately equal to B_0 . In Eq. (10), N_F consists of two parts. Each of the numerators can be rewritten as follows

$$\begin{aligned}
 Part1 &= \sum_{m=1}^{N-1} m \cdot P_F(m|S) \\
 &= NB_0 \sum_{m=1}^{N-1} (m \cdot C_{N-1}^m) \left(\sum_{i=1}^{N_{WU}} B_i \right)^m \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-1-m} \\
 &= NB_0(N-1) \left(\sum_{i=1}^{N_{WU}} B_i \right) \sum_{m=1}^{N-1} C_{N-2}^{m-1} \left(\sum_{i=1}^{N_{WU}} B_i \right)^{m-1} \\
 &\quad \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-2-(m-1)} \\
 &= N(N-1)B_0 \left(\sum_{i=1}^{N_{WU}} B_i \right) (1-B_0)^{N-2} = O(B_0^2).
 \end{aligned} \tag{23}$$

$$\begin{aligned}
 Part2 &= \sum_{k=2}^N \left(\sum_{m=1}^{N-k} m \cdot P_F(k,m|C) \right) \\
 &= \sum_{k=2}^N C_N^k B_0^k \sum_{m=1}^{N-k} (m \cdot C_{N-k}^m) \left(\sum_{i=1}^{N_{WU}} B_i \right)^m \\
 &\quad \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-k-m} = \sum_{k=2}^N C_N^k B_0^k (N-k) \left(\sum_{i=1}^{N_{WU}} B_i \right) \\
 &\quad \sum_{m=1}^{N-k} C_{N-k-1}^{m-1} \left(\sum_{i=1}^{N_{WU}} B_i \right)^{m-1} \cdot \left(1 - \sum_{i=0}^{N_{WU}} B_i \right)^{N-k-m} \\
 &= \sum_{k=2}^N C_N^k B_0^k (N-k) \left(\sum_{i=1}^{N_{WU}} B_i \right) (1-B_0)^{N-k-1} = O(B_0^3).
 \end{aligned} \tag{24}$$

As for N_C , its numerator in Eq. (9) can be written as

$$\begin{aligned}
 \sum_{k=2}^N k \cdot P_C(k) &= \sum_{k=2}^N k \cdot C_N^k B_0^k \cdot (1-B_0)^{N-k} \\
 &= N(N-1)B_0^2(1-B_0)^{N-2} + O(B_0^3).
 \end{aligned} \tag{25}$$

Then, N_F/N_C is computed as

$$\begin{aligned}
 \lim_{W \rightarrow \infty} \frac{N_F}{N_C} &= \lim_{B_0 \rightarrow 0} \frac{N(N-1)B_0 \left(\sum_{i=1}^{N_{WU}} B_i \right) (1-B_0)^{N-2} + O(B_0^3)}{N(N-1)B_0^2(1-B_0)^{N-2} + O(B_0^3)} \\
 &= \lim_{B_0 \rightarrow 0} \frac{\sum_{i=1}^{N_{WU}} B_i}{B_0} = N_{WU},
 \end{aligned} \tag{26}$$

which approaches N_{WU} as CW_{min} (W) increases.

References

1. Lee, K., Lee, J., Yi, Y., Rhee, I., & Chong, S. (2013). Mobile data offloading: How much can WiFi deliver? *IEEE/ACM Transactions on Networking*, 21(2), 536–550.
2. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. (2012). IEEE Std. 802.11.
3. He, Y., & Yuan, R. (2009). A novel scheduled power saving mechanism for 802.11 wireless LANs. *IEEE Transactions on Mobile Computing*, 8(10), 1368–1383.
4. Bahl, P., Adya, A., Padhye, J., & Wolman, A. (2004). Reconsidering wireless systems with multiple radios. *ACM SIGCOMM Computer Communication Review*, 34(5), 39–46.
5. Demirkol, I., Ersoy, C., & Onur, E. (2009). Wake-up receivers for wireless sensor networks: Benefits and challenges. *IEEE Wireless Communications Magazine*, 16(4), 88–96.
6. Jelacic, V., Magno, M., Brunelli, D., Bilas, V., & Benini, L. (2012). Analytic comparison of wake-up receivers for WSNs and benefits over the wake-on radio scheme. In *Proceedings of the ACM PM2HW2N'12* (pp. 99–106).
7. Park, M., Azizi, S., Stacey, R., Tian, B., Vegt, R. D., Jones, V., Au, E., Yang, D. X., Yu, R. J., & Yang, Y. (2016). Proposal for wake-up receiver (WUR) study group. IEEE, Tech. Rep. doc.: IEEE 802.11-16/0722r1.
8. Zhang, X., & Shin, K. G. (2012). E-MiLi: Energy-minimizing idle listening in wireless networks. *IEEE Transactions on Mobile Computing*, 11(9), 1441–1454.
9. Oller, J., Demirkol, I., Casademont, J., Paradells, J., Gamm, G., & Reindl, L. (2016). Has time come to switch from duty-cycled MAC protocols to wake-up radio for wireless sensor networks? *IEEE/ACM Transactions on Networking*, 24(2), 674–687.
10. Tang, S., Yomo, H., Kondo, Y., & Obana, S. (2012). Wake-up receiver for radio-on-demand wireless LANs. *EURASIP Journal on Wireless Communications and Networking*, 2012, 42.
11. Tang, S., Yomo, H., & Takeuchi, Y. (2015). Optimization of frame length modulation-based wake-up control for green WLANs. *IEEE Transactions on Vehicular Technology*, 64(2), 768–780.
12. Qin, H., & Zhang, W. (2014). ZigBee-assisted power saving management for mobile devices. *IEEE Transactions on Mobile Computing*, 13(12), 2933–2947.
13. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the ACM SenSys'04* (pp. 95–107).
14. Chebrolu, K., & Dhekne, A. (2009). Esense: Communication through energy sensing. In *Proceedings of the ACM MobiCom'09* (pp. 85–96).
15. Kondo, Y., Yomo, H., Tang, S., Iwai, M., Tanaka, T., Tsutsui, H., et al. (2012). Energy-efficient WLAN with on-demand AP wake-up using IEEE 802.11 frame length modulation. *Computer Communications*, 35(14), 1725–1735.
16. Tang, S., Yomo, H., & Obana, S. (2015). Dynamic threshold selection for frame length-based wake-up control. *IEEE Wireless Communications Letters*, 4(6), 609–612.
17. Tang, S., & Obana, S. (2016). Tight integration of wake-up radio in wireless LANs and the impact of wake-up latency. In *Proceedings of the IEEE Globecom*.
18. Hong, K., Lee, S., Kim, K., & Kim, Y. (2012). Channel condition based contention window adaptation in IEEE 802.11 WLANs. *IEEE Transactions on Communications*, 60(2), 469–478.
19. Deng, D.-J., Ke, C.-H., Chen, H.-H., & Huang, Y.-M. (2008). Contention window optimization for IEEE 802.11 DCF access control. *IEEE Transactions on Wireless Communications*, 7(12), 5129–5135.
20. Hill, J. L., & Culler, D. E. (2002). Mica: a wireless platform for deeply embedded networks. *IEEE Micro*, 22(6), 12–24.
21. Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the ACM SenSys'06* (pp. 307–320).
22. Anastasi, G., Conti, M., Francesco, M. D., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537–568.
23. Baiamonte, V., & Chiasserini, C.-F. (2006). Saving energy during channel contention in 802.11 WLANs. *Mobile Networks and Applications*, 11(2), 287–296.
24. Liang, J., & Shoemake, M. (2008). Low power packet detector for low power WLAN devices. US Patent 7,403,511. [Online]. <https://www.google.com/patents/US7403511>.
25. Mishra, N., Chebrolu, K., Raman, B., & Pathak, A. (2006). Wake-on-WLAN. In *Proceedings of the ACM WWW'06* (pp. 761–769).
26. Zhou, R., Xiong, Y., Xing, G., Sun, L., & Ma, J. (2010). Zifi: Wireless LAN discovery via ZigBee interference signatures. In *Proceedings of the ACM MobiCom'10* (pp. 49–60).
27. Djiroun, F. Z., & Djenouri, D. (2017). MAC protocols with wake-up radio for wireless sensor networks: A review. *IEEE Communications Surveys & Tutorials*, 19(1), 587–618.
28. Ansari, J., Pankin, D., & Mahonen, P. (2009). Radio-triggered wakeups with addressing capabilities for extremely low power sensor network applications. *International Journal of Wireless Information Networks*, 16(3), 118.
29. Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 535–547.
30. Chatzimisios, P., Vitsas, V., & Boucouvalas, A. C. (2002). Throughput and delay analysis of IEEE 802.11 protocol. In *Proceedings 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications* (pp. 168–174).
31. David, H. A., & Nagaraja, H. N. (2003). *Order statistics* (3rd ed.). New York: Wiley-Interscience.
32. Dogar, F. R., Steenkiste, P., & Papagiannaki, K. (2010). Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the ACM MobiSys'10* (pp. 107–122).



Suhua Tang received the B.S. degree in Electronic Engineering in 1998 and the Ph.D. degree in Information and Communication Engineering in 2003, both from University of Science and Technology of China. From October 2003 to March 2014, he was with Adaptive Communications Research Laboratories, ATR, Japan. Since April 2014, he is an assistant professor with Graduate School of Informatics and Engineering, The University of Electro-Communications, Japan. His research interests include green communications, ad hoc and sensor networks, inter-vehicle communications and high precision positioning. He is a member of IEEE, IEICE and IPSJ.



Sadao Obana received the B.E., M.E. and Ph.D. Degrees from Keio Univ. Tokyo, Japan, in 1976, 1978 and 1993 respectively. After joining KDDI (former KDD) in 1978, he was engaged in R&D in the field of packet exchange systems, network architecture, OSI (Open Systems Interconnection) protocols, database, distributed processing, network management and ITS (Intelligent Transport Systems). In 2004, he

joined Advanced Telecommunication Research Institute International (ATR) and was a director of

Adaptive Communications Research Laboratories in ATR. Since 2011, he is a professor with Graduate School of Informatics and Engineering, The University of Electro-Communications, Japan. His current research areas are wireless mobile ad-hoc network, ITS, sensor networks and low-power communications. He received an Award of Minister of Education, Culture, Sports, Science and Technology in 2001. Dr. Obana is a member of The Institute of Electronics, Information and Communication Engineers (IEICE) and a Fellow of Information Processing Society of Japan (IPSI).