An Analytical MAC Model for IEEE 802.15.4 Enabled Wireless Networks With Periodic Traffic

Xianghui Cao, *Member, IEEE*, Jiming Chen, *Senior Member, IEEE*, Yu Cheng, *Senior Member, IEEE*, Xuemin (Sherman) Shen, *Fellow, IEEE*, and Youxian Sun

***Abstract*—The IEEE 802.15.4 standard, which supports low-cost communications, has been applied in a variety of wireless networks. Developing accurate analytical models for the IEEE 802.15.4 medium access control (MAC) protocol is critical for the design and performance evaluation of such networks. Periodic traffic is a common traffic pattern generated in many practical application scenarios, for which most existing analytical models assuming either saturated or random network traffic patterns become inapplicable. In this paper, we develop an accurate and scalable analytical model to analyze the IEEE 802.15.4 MAC protocol with the periodic traffic. Our model can accurately capture the protocol stochastic behavior in each period in scenarios such as with or without retransmissions and with single clear channel assessment (CCA) or double CCAs. Extensive simulations are conducted to validate the proposed model by both transient and aggregate performance evaluations, and the results show that the model captures MAC behavior with periodic traffic accurately. We also discuss about extending the proposed model to account for heterogeneous scenarios and the hidden node problem.**

***Index Terms*—IEEE 802.15.4, periodic traffic, CSMA/CA, analytical model, throughput, energy consumption.**

I. INTRODUCTION

THE IEEE 802.15.4 standard specifies a suite of medium access control (MAC) and physical layer protocols for low-power, low-cost and low-data-rate applications [1]. Established based on this standard, some protocols such as ZigBee [2], WirelessHART [3], MiWi [4] and ISA100.11a [5] gain popularity in various fields. WirelessHART, MiWi and ISA100.11a have been applied in wireless instrumentation and industrial control applications [6], while ZigBee has become one of the most prevalent techniques in wireless sensor networks (WSNs) and machine-to-machine (M2M) networks (e.g., home automation networks, on-body sensor networks, and smart meter networks) [7]–[9]. Some recent developments also show that ZigBee can be readily integrated into mobile phones to work jointly with Wi-Fi and cellular modules [10], making it potentially feasible for saving energy in mobile applications such as the so-called outband device-to-device (D2D) communications [11].

In IEEE 802.15.4-enabled wireless networks, the MAC protocol plays a vital role in data transmission scheduling and greatly affects the network performance. Accurately modeling the MAC behavior is critical for network performance evaluation and optimization. The MAC layer specifies a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism which allows network nodes to contend for channel access in a distributed manner. Most existing models of IEEE 802.15.4 focus on either saturated or random traffic (or unsaturated traffic) patterns [12]–[20]. In applications where the network is likely to be congested, the MAC service rate of a node may be much lower than its data traffic arriving rate, resulting in the *saturated traffic* pattern. In this case, a node always has packets in its buffer for transmitting. Whereas, the *unsaturated traffic* pattern, generated in many low-load applications, assumes that a node has much less packets which arrive at *random* time instances.

Unlike those two patterns, the data traffic in many practical applications may be generated periodically, resulting in the *periodic traffic* pattern. Periodic traffic can be observed in WSNs. For example, in environment monitoring applications, sensors are usually configured to periodically sense the environment and report data to a remote center [21]. In duty-cycled networks, each node may periodically wake up for communication in order to save energy [22]. In wireless control networks, discrete-time control strategies with strict and synchronous sensor data reporting periods are the common way to implement closed-loop control systems [23], [24]. Periodic traffic is also a major traffic pattern generated by neighbor discovery and network maintenance processes in IEEE 802.15.4-enabled wireless networks [25]. For example, in a neighbor discovery process, a node broadcasts a request packet to its neighbors who will contend for replying the sender node upon receiving the request. Such a communication mechanism is also commonly used in WSNs for data aggregation applications [26]. Note that those neighbors may start contention at the same time after receiving the request, which is an important feature of periodic traffic.

Manuscript received August 12, 2014; revised December 13, 2014 and March 24, 2015; accepted May 5, 2015. Date of publication May 19, 2015; date of current version October 8, 2015. This work was supported in part by the National Science Foundation under Grants CNS-1117687 and CNS-1320736, by the National Natural Science Foundation of China under Grants 61203036 and 61222305, and by the 111 Program under Grant B07031. The associate editor coordinating the review of this paper and approving it for publication was C.-F. Chiasserini.

X. Cao is with the School of Automation, Southeast University, Nanjing 210096, China, and also with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: xh.cao@ieee.org).

J. Chen and Y. Sun are with the Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: jmchen@iipc.zju.edu.cn; yxsun@iipc.zju.edu.cn).

Y. Cheng is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: cheng@iit.edu).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: xshen@bbcr.uwaterloo.ca).

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/TWC.2015.2435006

In this paper, we develop an accurate analytical model of the CSMA/CA protocol with periodic traffic for IEEE 802.15.4-enabled wireless networks. A special challenge is that the MAC statistics change along time in the contention period, which is hard to capture by either saturated or unsaturated models found in the literature. Moreover, existing models for periodic traffic often make strong assumptions as explained in detail in the next section. Our main contributions can be summarized as follows. (1) We propose an accurate model for the IEEE 802.15.4 CSMA/CA protocol, by which we explore the time-varying feature of the MAC statistics under periodic traffic. Our model is scalable by modeling the behavior of a typical tagged node. It has a clean structure, including *parallel updating* operations to capture the behavior of the tagged node and *cross updating* operations to describe the probabilities of clear channel assessment (CCA) failures and collisions in transmissions. We analyze both situations with and without retransmissions. Our model considers the scenario of double CCAs, while it can be readily reduced to the case of single CCA. (2) Based on the model, we develop analytical expressions for the MAC performance in terms of aggregate throughput of the data packets and the energy consumption in one contention period. (3) Through extensive simulations, we demonstrate the accuracy and scalability of our model. Moreover, we show that existing models for either saturated or unsaturated traffic patterns generally do not apply in the periodic traffic case. We compare the performance between the cases of single CCA and double CCAs and show that the latter has much higher throughput than the former. We also discuss the impact of the contention period length on the throughput. (4) Finally, we discuss extensions of our model to accommodate the heterogeneous network environment where nodes use diverse parameters and the hidden node problem.

The remainder of this paper is organized as follows. Section II reviews existing models of IEEE 802.15.4 MAC. Section III introduces the CSMA/CA protocol. Our analytical model is presented in Section IV, following which the MAC performance is analyzed in Section V. Simulation results are presented in Section VI. Section VII presents extensions and discussions. Finally, Section VIII concludes this paper.

II. Related Work

Developing analytical models of the IEEE 802.15.4 MAC protocol (particularly for the CSMA/CA mode) has been a hot research topic in recent years. Most existing models can be divided into two categories according to the traffic patterns, i.e., saturated traffic based models and random traffic (or unsaturated traffic) based models. Inspired by the model primarily proposed for IEEE 802.11 [27], a number of analytical models for saturated traffic have been proposed [12], [13], [16]. The main ideas are to apply either fixed-point techniques [14], [15] or Markov chain based model [12] for stationary statistics analysis. The models for saturated traffic have been extended to accommodate unsaturated traffic patterns, assuming either Poisson [17]–[19] or uniform [20] random traffic, where Markov chain based approaches are often adopted. Alternatively, a renewal theory based approach is proposed in [14], which has low computation complexity.

TABLE I
Basic Notations

Notation	Definition
n	Number of nodes
K	Total number of slots in a contention period
s	Backoff stage. $s \in \{0, 1, \ldots, M\}$
i_s	Backoff exponent in stage s. $i_s \in \{BE_{\min}, \ldots, BE_{\max}\}$
$BE_{\min}$	Minimum backoff exponent
$BE_{\max}$	Maximum backoff exponent
c	Index of CSMA/CA re-initialization times in each transmission/retransmission round. $c \in \{0, \ldots, C\}$
r	Index of retransmission times. $r \in \{0, 1, \ldots, R\}$
L	The length (in slots) of each packet

In the literature, there are very few models for the CSMA/CA protocol with periodic traffic. Pollin *et al.* extend their Markov chain based MAC model for saturated traffic to an unsaturated traffic case which basically seeks the stationary statistics of the MAC operations. However, we will show in this paper that, under periodic traffic, the MAC statistics do not follow Markov chain and no stationary statistics can be obtained [12]. Moreover, the traffic pattern in [12] assumes each transmission is followed by a fixed idle period, which is different from ours. Although the model proposed in [28] can be adapted to the periodic traffic case, it focuses on a purely contention based MAC without carrier sensing.

With periodic traffic, the MAC statistics of each node will vary along time during each period, which introduces a special challenge for MAC performance analysis. Existing techniques based on steady-state analysis for either saturated or unsaturated traffic cannot be directly extended to the periodic traffic scenario; thus non-stationary techniques to analyze the transient behavior within each period are needed. The transient behavior of the CSMA/CA MAC with periodic traffic is studied in [29], [30] by approaches based on state transitions. Such models are further improved by extending to a high-dimensional Markov chain based model [26]. Haghighi *et al.* consider a very similar problem in studying flooding attack problems and propose an analytical CSMA/CA model [31]. However, these works assume single CCA (i.e., CW = 1) while the CSMA/CA protocol defines CW = 2, where CW is the contention window size. We will show later in this paper that this assumption significantly changes the network performance under periodic traffic. Moreover, few of them consider retransmission, an important option offered by IEEE 802.15.4. Although the model proposed in [32] successfully accounts for double CCAs, it does not allow protocol re-initialization and retransmissions and the model is heavyweight. Specifically, to capture the transient behavior of the network, it maintains a much bigger set of variables than our model does, which will incur higher computation complexity.

III. Preliminaries

In this section, we overview the IEEE 802.15.4 CSMA/CA protocol, and indicate the assumptions taken in this paper. Table I summarizes the main notations in this paper. We consider a network consisting of one coordinator (e.g., the sink node in WSNs) and n nodes. The nodes periodically send data packets (or *packets* for short) to the coordinator (which broadcasts beacons to control nodes' periods).

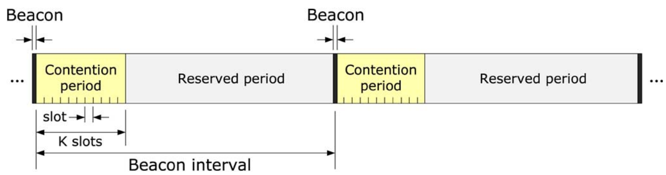


Fig. 1. Structure of beacon intervals (or periods).

As shown in Fig. 1, a period consists of a contention period and a reserved period. The coordinator periodically broadcasts beacons to notify the nodes the beginning of the contention periods. Upon receiving a beacon, each node applies the beacon-enabled mode of IEEE 802.15.4 (and hence with slotted version of CSMA/CA protocol) to contend for sending one packet to the coordinator. According to IEEE 802.15.4, the contention period is equally divided into K unit slots, which are called backoff slots or slots for short. The length of the contention period is fixed while that of the reserved period can be variable. In [33], the authors propose a similar structure as the basis of a hybrid MAC protocol for M2M networks, where the reserved period is allocated for those nodes who have succeeded in the contention period. However, the contention period adopts a p-persistent CSMA MAC and the modeling approach does not take periodic traffic into account.

We consider a one-hop network with all nodes locating within each other's communication ranges such that no hidden node problem presents [27], [32]. No capture effect is assumed such that a packet will get lost whenever it is collided with another packet (including the acknowledgement sent from the coordinator). We assume that beacons and acknowledgement packets sent by the coordinator are loss-free. Since all the nodes receive a beacon simultaneously, they start to contend for channel access simultaneously.[1] For simplicity, we assume homogeneous nodes such that their MAC protocols are configured with the same parameters and all the packets are of the same length L. Extension to heterogeneous scenarios where the nodes can use different parameters is discussed later in Section VII.

A. IEEE 802.15.4 MAC

As shown in Fig. 2, the CSMA/CA protocol defines two basic operations, i.e., random backoff and CCA, both of which should be launched at slot boundaries. A CCA lasts for one slot and is conducted after a backoff. We say that a CCA is successful if the channel is clear (i.e., unoccupied by other nodes or the coordinator) in that slot. A node transmits its packet only after two successive successful CCAs (denoted by CCA1 and CCA2, respectively). A backoff is performed after either the protocol initialization or a CCA failure, and lasts for a random number of slots uniformly chosen from $\{0, 1, \ldots, W(i_s) - 1\}$, where

$$W(i_s) \triangleq 2^{i_s}, \tag{1}$$

i_s represents the backoff exponent and s is the backoff stage. s is initialized at 0 and increases by 1 if a new backoff is performed.

[1]The signal propagation delay and data preparation time are assumed negligible.

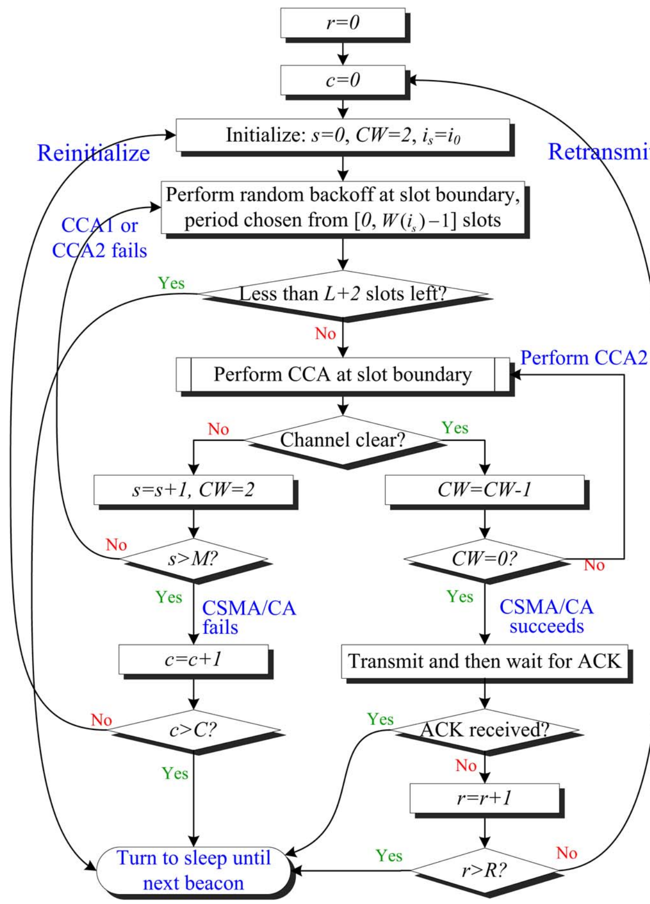


Fig. 2. Slotted version of the IEEE 802.15.4 CSMA/CA protocol with re-initializations and retransmissions.

$i_{s+1} = \min\{i_s + 1, BE_{\max}\}$ and $i_0 = BE_{\min}$. The CSMA/CA protocol fails if s has reached its maximum M but two successive successful CCAs have not been achieved. If more than one node transmits packets simultaneously, collisions happen at the coordinator side. In this case, we can enable the retransmission option which triggers the following procedure. The coordinator will reply an acknowledgement packet (ACK) to the sender upon successfully receiving a packet; the sender retransmits the packet if ACK is not received during a certain period of time after the previous transmission. The retransmission rounds are counted by $r \in \{0, 1, \ldots, R\}$ where R is the maximum number of retransmission times. We call the first time transmission round as the 0-th retransmission round.

In this paper, since all the packets arriving within the contention period are valid to the coordinator, once CSMA/CA fails, it is a good option for a node to re-initialize the protocol unless the remaining time of the contention period is less than[2] $L + 2$, i.e., it is impossible to complete a transmission (including performing two CCAs and transmitting a packet) in that time. Let c be the re-initialization count and C be the maximum number of re-initialization times for each retransmission round. The special case with $c = 0$ indicates that the protocol is

[2]The coordinator can encapsulate the length of the contention period in the beacons to be broadcasted over the network. With the knowledge of K, each node can know whether to permit a protocol re-initialization.

TABLE II
VARIABLES DEFINED FOR THE CASES WITHOUT RETRANSMISSION

Variable	Definition
POS	Protocol operation status. POS$=(c,s)$ means that the preceding backoff is in stage s and the protocol has been re-initialized c times, where $s\in\{0,\ldots,M\}$ and $c\in\{0,\ldots,C\}$
τ_k	$\Pr\{\mathcal{D}^\dagger$ performs CCA1 in k-th slot$\}$
$\alpha_{1,k}$	$\Pr\{$a CCA1 succeeds $\mid$ it is performed in k-th slot$\}$
$\alpha_{2,k}$	$\Pr\{$a CCA2 succeeds $\mid$ it is performed in k-th slot (i.e., a CCA1 in $(k-1)$-th slot succeeded)$\}$
α_k	$\Pr\{$two successive CCAs succeed $\mid \mathcal{D}^\dagger$ performed CCA1 in $(k-1)$-th slot$\}$
$\beta^c_{s,k}$	$\Pr\{\mathcal{D}^\dagger$ is currently performing CCA1 and POS$=(c,s)\}$
$\lambda^c_{s,b,k}$	$\Pr\{\mathcal{D}^\dagger$ performs CCA1 in slot k and POS$=(c,s)\mid$ preceding backoff length $=b\}$, where $b\in\{0,1,\ldots,W(i_s)-1\}$

initialized when a retransmission is just started, which is called the 0-th re-initialization for convenience.

As shown in Fig. 2, a node will turn to sleep in a contention period in two general cases: 1) it finishes serving the packet, i.e., one of the following happens: the packet has been successfully transmitted and the corresponding ACK has been received, the re-initialization count c exceeds its maximum C, and the retransmission count r exceeds its maximum R; 2) there are less than L available slots in the contention period when the node is not transmitting (i.e., the residual slots are not enough for transmitting a packet).

IV. ANALYTICAL MAC MODEL

In this section, our analytical model is presented in detail. To demonstrate the basic modeling steps, we first focus on the simple case that the retransmission option is disabled, i.e., $R=0$. Extensions are made in the second part of this section to accommodate retransmission cases.

A. Without Retransmissions

As aforementioned, compared with either saturated or random traffic patterns, the periodic traffic pattern does not render stationary statistics of each node's performance in a per-slot basis; whereas, it incurs that the statistics of each node in one contention period may vary from slot to slot. For example, a node will not contend for channel access after having transmitted its packet in a contention period. Without loss of generality, we focus on one contention period where the slots are indexed by $k=0,\ldots,K-1$. The variables used in this subsection are defined in Table II. Since the MAC protocol for each node is fair, we can concentrate on an arbitrary node, namely $\mathcal{D}^\dagger$. We focus on the event that $\mathcal{D}^\dagger$ executes a CCA since only at this time the channel information is observed by $\mathcal{D}^\dagger$. If $\mathcal{D}^\dagger$ performs CCA1 in slot k, the POS (see definition in Table II) of $\mathcal{D}^\dagger$ can be described by the tuple (c,s).

As mentioned in Section III, CCA1 will not be performed if the number of residual slots in current contention period is no more than $L+1$. Therefore, $\forall k\ge K-L-1$, define $\tau_k=0$ and $\beta^c_{s,k}=0$. Noticing that $\alpha_{1,k},\alpha_{2,k}$ and α_k are conditional probabilities, we can simply define them to be 0 if CCA1 is not performed, i.e., $\alpha_{1,k}=0$ if $\tau_k=0$ and $\alpha_{2,k}=\alpha_k=0$ if

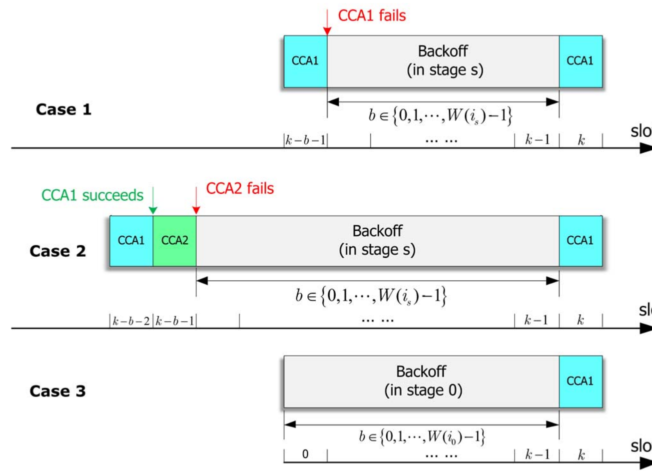


Fig. 3. The three cases that cause $\mathcal{D}^\dagger$ to perform CCA1 in k-th slot.

$\tau_{k-1}=0$. Our analytical model updates the above probabilities recursively. For the convenience of applying recursion, define $\tau_k=\alpha_{1,k}=\alpha_{2,k}=\alpha_k=\beta^c_{s,k}=0$ for all $k<0$.

Note that whether $\mathcal{D}^\dagger$ performs CCA1 in k-th slot is independent of other nodes' activities in the same slot, but the successfulness of that CCA1 is dependent. Such independence and dependence are accounted by the *parallel updating* and *cross updating*, respectively, as follows. In the former one, $\mathcal{D}^\dagger$ updates the probabilities τ_k and $\{\beta^c_{s,k}\}$ based on their historical values of its own. The latter process updates the probabilities $\alpha_{1,k},\alpha_{2,k}$ and α_k by taking into account the interactions between $\mathcal{D}^\dagger$ and other nodes.

1) Parallel Updating: First, consider the probability $\lambda^c_{s,b,k}$. Depending on the POS (i.e., (c,s)), $\lambda^c_{s,b,k}$ can be calculated by considering the following three scenarios.

- $s\ge 1$: since s is reset to 0 when the protocol is re-initialized, $s\ge 1$ means $\mathcal{D}^\dagger$ has conducted backoff at least twice after c-th re-initialization. Therefore, only if previously either a CCA1 fails (with probability $\beta^c_{s-1,k-b-2}\alpha_{1,k-b-2}$, corresponding to case 1 in Fig. 3) or a CC2 fails (with probability $\beta^c_{s-1,k-b-2}\alpha_{1,k-b-2}(1-\alpha_{2,k-b-1})$, corresponding to case 2 in Fig. 3), $\mathcal{D}^\dagger$ will enter a new backoff stage and perform CCA1 after a backoff. Given the current POS as (c,s), the POS when $\mathcal{D}^\dagger$ was performing the previous CCA1 or CCA2 is $(c,s-1)$. Therefore,

$$\lambda^c_{s,b,k}=\underbrace{\beta^c_{s-1,k-b-1}(1-\alpha_{1,k-b-1})}_{\text{Case 1}}+\underbrace{\beta^c_{s-1,k-b-2}\alpha_{1,k-b-2}(1-\alpha_{2,k-b-1})}_{\text{Case 2}}. \quad (2)$$

- $s=0, c\ge 1$: in this scenario, $\mathcal{D}^\dagger$ is performing CCA1 for the first time after c-th re-initialization, which is the consequence of the following events: in $(k-b)$-th slot, either a CCA1 or a CCA2 fails (when the POS is $(c-1,M)$), i.e., $\mathcal{D}^\dagger$ fails to get channel access and its backoff stage already reaches M; hence it performs a new re-initialization (i.e., the c-th re-initialization). Similar to the above scenario,

$$\lambda_{0,b,k}^{c} = \beta_{M,k-b-1}^{c-1}(1-\alpha_{1,k-b-1}) + \beta_{M,k-b-2}^{c-1}\alpha_{1,k-b-2}(1-\alpha_{2,k-b-1}). \quad (3)$$

- $s = c = 0$: in this special case, $\mathcal{D}^{\dagger}$ performs CCA1 for the first time in the contention period (see case 3 in Fig. 3). Therefore, the probability $\lambda_{0,b,k}^{0}$ depends on the probability of the preceding backoff length b. Since b is uniformly chosen from $\{0, \ldots, W(i_0)-1\}$,

$$\lambda_{0,b,k}^{0} = \begin{cases} \frac{1}{W(i_0)}, & \text{if } k \in \{0, \ldots, W(i_0)-1\}, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

With $\lambda_{s,b,k}^{c}$ obtained as above, $\beta_{s,k}^{c}$ can be then calculated by the total probability formula with all possible backoff length b taken into account. Thus,

$$\beta_{s,k}^{c} = \sum_{b=0}^{W(i_s)-1} \lambda_{s,b,k}^{c} \Pr\{\text{backoff length} = b\} = \frac{1}{W(i_s)} \sum_{b=0}^{W(i_s)-1} \lambda_{s,b,k}^{c}. \quad (5)$$

Moreover, by definition, τ_k can be calculated by taking into account all possible POS as follows.

$$\tau_k = \sum_{c=0}^{C}\sum_{s=0}^{M} \beta_{s,k}^{c}. \quad (6)$$

2) Cross Updating: To calculate $\alpha_{1,k}$, i.e., the success probability of a CCA1, let's consider the corresponding failure probability $1-\alpha_{1,k}$, i.e., the conditional probability that the channel is busy (at least one of the other nodes is transmitting) when $\mathcal{D}^{\dagger}$ is performing CCA1 during k-th slot. With the packet length L, the transmission(s) causing the CCA failure at slot k may start in slots $k-L+1, \cdots, k-1, k$. As a result, if $\tau_k \neq 0, \forall k \in \{0, 1, \ldots, K-L-2\}$,

$$\begin{aligned} 1-\alpha_{1,k} &= \sum_{l=0}^{L-1} \Pr\{\text{At least one of the other nodes starts to transmit in slot } k-l\} \\ &= \sum_{l=1}^{L} \Pr\{\text{At least one of the other nodes performs CCA1 in slot } k-l-1\} \times \alpha_{k-l} \\ &= \sum_{l=1}^{L} \left[1-(1-\tau_{k-l-1})^{n-1}\right]\alpha_{k-l}, \end{aligned} \quad (7)$$

where we have used the fact that the events that the nodes start to transmit in the same slot are dependent. For example, if one node starts to transmit in slot $k-l$ (indicating that the channel is free in the previous two slots), another node will also start to transmit if it performs CCA1 in slot $k-l-1$. This explains the deduction from the second line to the third of (7). If $\tau_k = 0$, $\alpha_{1,k}$ is defined as 0. Note that we have defined that $\tau_k = 0$ and $\alpha_k = 0$ if $k < 0$. Therefore, in case $k \le L$, only the terms with $1 \le l \le k-1$ in the summation in (7) may be non-zero. This is in consistence with the fact that transmission(s) of the other node(s) only starting at slot $1, \ldots, k-1$ can cause the channel busy in slot k.

Once a node has transmitted its packet, it will enter the sleep mode, which means that, for one run, the number of active nodes (denoted as n_a) is decreasing. Moreover, n_a is a stochastic variable along time. However, the above equation uses n instead of n_a because the probability τ_k already contains the information of a node being active. In fact, τ_k can also be interpreted as the probability that the tagged node is active and performs CCA1 in slot k.

Similarly, the event that a CCA2 fails in k-th slot (its preceding CCA1 succeeds) happens only when there are other nodes that start transmitting exactly in k-th slot. Therefore, if $\alpha_{1,k-1} \neq 0$,

$$\begin{aligned} \alpha_{1,k-1}(1-\alpha_{2,k}) &= \left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1} \\ \Rightarrow \quad \alpha_{2,k} &= 1 - \frac{1}{\alpha_{1,k-1}}\left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1}. \end{aligned} \quad (8)$$

Otherwise, $\alpha_{2,k} = 0$. By definition, if $\tau_{k-1} \neq 0$,

$$\begin{aligned} \alpha_k &= \alpha_{1,k-1}\alpha_{2,k} \\ &= \alpha_{1,k-1} - \left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1} \\ &= 1 - \sum_{l=1}^{L+1}\left[1-(1-\tau_{k-l-1})^{n-1}\right]\alpha_{k-l}, \end{aligned} \quad (9)$$

where we have used (7). $\alpha_k = 0$ if $\tau_{k-1} = 0$. Since $\mathcal{D}^{\dagger}$ cannot perform CCA2 in the first slot, we define $\alpha_{2,0} = \alpha_0 = 0$.

B. With Retransmissions

A transmission after two consecutive successful CCAs may collide with other simultaneous transmissions. When the retransmission of a collided packet is allowed (unless the number of retransmission times has reached its maximum), a node will wait L_w slots for receiving the coordinator's ACK after each transmission. If no ACK is received when L_w expires, a collision is indicated and the node will apply the CSMA/CA protocol to try transmitting this packet again. Let L_{ack} denote the length (in slots) of an ACK. According to IEEE 802.15.4,

$$L_w = L_{turn} + L_{ack}, \quad (10)$$

where L_{turn} represents the time (in slots) for the coordinator's transceiver to switch between transmitting and receiving modes. $L_{turn} < 2$ according to the standard.

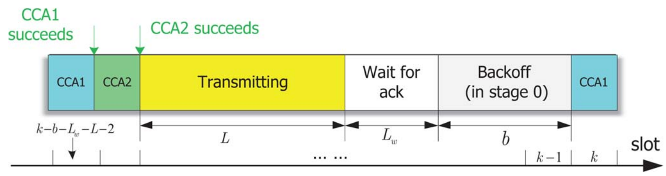


Fig. 4. A case that causes $\mathcal{D}^{\dagger}$ to perform CCA1 in k-th slot when retransmissions are allowed.

With retransmissions, the POS of $\mathcal{D}^{\dagger}$ is now redefined as (r, c, s) which means: $\mathcal{D}^{\dagger}$ attempts to retransmit the packet for r-th time, the preceding backoff is in stage s and the protocol has been re-initialized c times, where $r \in \{0, \ldots, R\}$, $c \in \{0, \ldots, C\}$ and $s \in \{0, \ldots, M\}$. Thus, $\beta^{c}_{s,k}$ is redefined as $\beta^{c,r}_{s,k}$, i.e., the probability that $\mathcal{D}^{\dagger}$ is performing CCA1 and POS $= (r, c, s)$. Accordingly, $\lambda^{c}_{s,b,k}$ is redefined as $\lambda^{c,r}_{s,b,k}$.

The extended analytical model also consists of parallel and cross updating processes as follows.

1) Parallel Updating: Consider the probability $\lambda^{c,r}_{s,b,k}$. It is determined by taking into account of the following scenarios:

- $s \geq 1$: in this case, $\mathcal{D}^{\dagger}$ has conducted backoff at least twice, which implies that the retransmission time as well as the re-initialization time do not change. Therefore, similar to (2),

$$\lambda^{c,r}_{s,b,k} = \beta^{c,r}_{s-1,k-b-2}\alpha_{1,k-b-2}(1-\alpha_{2,k-b-1}) + \beta^{c,r}_{s-1,k-b-1}(1-\alpha_{1,k-b-1}). \quad (11)$$

- $s = 0, c \geq 1$: $\mathcal{D}^{\dagger}$ is performing CCA1 for the first time after c-th re-initialization, which implies that the retransmission time does not change. Similar to (3),

$$\lambda^{c,r}_{0,b,k} = \beta^{c-1,r}_{M,k-b-2}\alpha_{1,k-b-2}(1-\alpha_{2,k-b-1}) + \beta^{c-1,r}_{M,k-b-1}(1-\alpha_{1,k-b-1}). \quad (12)$$

- $s = c = 0$, $r \geq 1$: this scenario indicates that a collision happened, and the protocol resets its parameters to $s = c = 0$ and increases the retransmission round to r. Let ω_k denote the collision probability, i.e., the probability that $\mathcal{D}^{\dagger}$ and at least one of the other nodes starts to transmit at k-th slot simultaneously, conditioned on that they perform CCA1 two slots before. Hence,

$$\omega_k = \left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1}. \quad (13)$$

  As shown in Fig. 4, given that the preceding backoff length is b, the collided transmission started $L + b + L_w$ slots ago. Since $\mathcal{D}^{\dagger}$ will start to transmit whenever two successive CCAs succeed, its POS right before the collided transmission can be any except for that the retransmission round is $r - 1$. Therefore,

$$\lambda^{0,r}_{0,b,k} = \sum_{c=0}^{C}\sum_{s=0}^{M} \beta^{c,r-1}_{s,k-b-L_w-L-2}\omega_{k-b-L_w-L}. \quad (14)$$

- $s = c = r = 0$: $\mathcal{D}^{\dagger}$ performs CCA1 for the first time in the contention period. Therefore,

$$\lambda^{0,0}_{0,b,k} = \lambda^{0}_{0,b,k}. \quad (15)$$

  where $\lambda^{0}_{0,b,k}$ has been defined in (4).

By considering all possible values of the preceding backoff length b and all possible POS, we can calculate $\beta^{c,r}_{s,k}$ and τ_k in a similar way as shown in (5) and (6), respectively.

$$\beta^{c,r}_{s,k} = \frac{1}{W(i_s)} \sum_{b=0}^{W(i_s)-1} \lambda^{c,r}_{s,b,k}, \quad (16)$$

$$\tau_k = \sum_{r=0}^{R}\sum_{c=0}^{C}\sum_{s=0}^{M} \beta^{c,r}_{s,k}. \quad (17)$$

2) Cross Updating: Consider that $\mathcal{D}^{\dagger}$ is performing CCA1 in slot k. Aside from the situation that the channel is occupied by other nodes, this CCA1 may also end up with a busy channel observation if the coordinator is busy sending an ACK. Let $\upsilon_{k,k-j}$ be the probability that an ACK is sent by the coordinator from slot $k - j$ on (which corresponds to that one of the other $n - 1$ nodes successfully transmitted a packet) conditioned on that $\mathcal{D}^{\dagger}$ is performing CCA1 in k-th slot.[3]

$$\upsilon_{k,k-j} = (n-1)\tau_{k-j-L-L_{turn}-2-j}\alpha_{k-j-L-L_{turn}-1} \times \left(1-\tau_{k-j-L-L_{turn}-2}\right)^{n-2}. \quad (18)$$

Therefore, if $\tau_k \neq 0$, the counterpart of (7) becomes

$$1-\alpha_{1,k} = \sum_{l=1}^{L}\left[1-(1-\tau_{k-l-1})^{n-1}\right]\alpha_{k-l} + \sum_{j=0}^{L_{ack}-1} \upsilon_{k,k-j}. \quad (19)$$

The counterparts of $\alpha_{2,k}$ and α_k can be obtained similarly. If $\alpha_{1,k-1} \neq 0$,

$$\alpha_{2,k} = 1 - \frac{\left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1} + \upsilon_{k,k}}{\alpha_{1,k-1}}, \quad (20)$$

and if $\tau_{k-1} \neq 0$,

$$\alpha_k = \alpha_{1,k-1} - \left[1-(1-\tau_{k-2})^{n-1}\right]\alpha_{k-1} - \upsilon_{k,k}. \quad (21)$$

V. MAC Performances Analysis

In the coordinator's point of view, all the packets successfully received during each contention period are valid and hence are of its concern. Therefore, we take one contention period as the analysis window and study the MAC performance in terms of the throughput of packets and energy consumption.

A. Without Retransmissions

1) Throughput: In slot k, the coordinator successfully decodes the packet from $\mathcal{D}^{\dagger}$ (i.e., only $\mathcal{D}^{\dagger}$ succeeds in performing

[3] $\mathcal{D}^{\dagger}$ cannot be transmitting simultaneously as the other nodes because otherwise it is currently waiting for ACK other than performing CCA1.

two CCAs and transmits this packet subsequently) with probability

$$\eta_k = \tau_{k-L-1}\alpha_{k-L}(1-\tau_{k-L-1})^{n-1}. \tag{22}$$

Thus, by counting all the slots across the contention period, the packet reception rate is $\sum_{k=0}^{K-1}\eta_k$. Define throughput (denoted as φ) as the total number of received packets during one contention period. In other words, φ denotes the number of nodes that succeed in the contention. Therefore,

$$\varphi = n\sum_{k=0}^{K-1}\eta_k. \tag{23}$$

2) Energy Consumption: Since the coordinator is always in high power for receiving or sending ACKs in the contention period, we shall only focus on nodes' energy consumption. Specifically, we study the *energy consumption rate*, i.e., per slot energy consumption of a node. Consider the state of $\mathcal{D}^\dagger$ in k-th slot. Let $p_k^{\text{cca}}, p_k^{\text{tx}}, p_k^{\text{bk}}$ and p_k^o be the probabilities that $\mathcal{D}^\dagger$ is conducting CCA, transmitting, performing backoff and sleeping, respectively. By definition, it is easy to verify that

$$p_k^{\text{tx}} = \sum_{l=1}^{L}\tau_{k-l-1}\alpha_{k-l}, \tag{24}$$

$$p_k^{\text{cca}} = \tau_k + \tau_{k-1}\alpha_{1,k-1}. \tag{25}$$

To calculate the probability p_k^{bk}, let's re-examine the probability $\lambda_{s,b,k}^c$. By definition, $\lambda_{s,b,k}^c$ is identical to the probability that $\mathcal{D}^\dagger$ starts a backoff with parameters s, c and b at slot $(k-b)$. Given s, c and b ($b \geq 1$), the present backoff period that covers slot k must start at slot $(k-b')$, where $b' \in \{0,\ldots,b-1\}$, and the corresponding probability is the same as $\lambda_{s,b',k}^c$. Then, taking all possible c, s, b and b' into account, one can easily verify that

$$p_k^{\text{bk}} = \sum_{c=0}^{C}\sum_{s=0}^{M}\sum_{b=1}^{W(i_s)-1}\frac{1}{W(i_s)}\sum_{b'=0}^{b-1}\lambda_{s,b',k}^c. \tag{26}$$

Finally, since there are no other states in which $\mathcal{D}^\dagger$ can reside, we immediately have

$$p_k^o = 1 - p_k^{\text{cca}} - p_k^{\text{tx}} - p_k^{\text{bk}}. \tag{27}$$

Denote by $\epsilon^{\text{cca}}, \epsilon^{\text{tx}}, \epsilon^{\text{bk}}$ and ϵ^o the energy consumption rates when $\mathcal{D}^\dagger$ is in the above four states, respectively. The average energy consumption rate is calculated by

$$e = \frac{1}{K}\sum_{k=0}^{K-1}\left(p_k^{\text{cca}}\epsilon^{\text{cca}} + p_k^{\text{tx}}\epsilon^{\text{tx}} + p_k^{\text{bk}}\epsilon^{\text{bk}} + p_k^o\epsilon^o\right). \tag{28}$$

B. With Retransmissions

Because $L_{turn} < 2$, after a successful transmission, the coordinator will start to transmit ACK before any node can finish two successive CCAs. In order words, a packet cannot collide with any ACK. Therefore, (22), (23) still hold. Also, it is easy to verify the validity of (24) and (25). Similar to (26), the backoff probability can be calculated as

$$p_k^{\text{bk}} = \sum_{r=0}^{R}\sum_{c=0}^{C}\sum_{s=0}^{M}\sum_{b=1}^{W(i_s)-1}\frac{1}{W(i_s)}\sum_{b'=0}^{b-1}\lambda_{s,b',k}^{c,r}. \tag{29}$$

Compared with the energy consumption rate in (28), there is an extra amount of energy consumed by a node for receiving the ACKs during the waiting periods. For the ACK receiving energy, we introduce p_k^{rx} to denote the ACK waiting probability. Because a node will not enter the waiting period if the maximum number of retransmissions has been used, the waiting probability is

$$\begin{aligned}
p_k^{\text{rx}} &= \sum_{l=1}^{L_w}\sum_{r=0}^{R-1}\sum_{c=0}^{C}\sum_{s=0}^{M}\beta_{s,k-L-l-1}^{c,r}\alpha_{k-L-l} \\
&= \sum_{l=1}^{L_w}\left(\sum_{r=0}^{R}\sum_{c=0}^{C}\sum_{s=0}^{M}\beta_{s,k-L-l-1}^{c,r} - \sum_{c=0}^{C}\sum_{s=0}^{M}\beta_{s,k-L-l-1}^{c,R}\right) \\
&\quad \times \alpha_{k-L-l} \\
&= \sum_{l=1}^{L_w}\left(\tau_{k-L-l-1} - \sum_{c=0}^{C}\sum_{s=0}^{M}\beta_{s,k-L-l-1}^{c,R}\right)\alpha_{k-L-l},
\end{aligned} \tag{30}$$

where we have used (17).

Let ϵ^{rx} be the energy consumption rate when $\mathcal{D}^\dagger$ is in the waiting period. The average energy consumption rate with retransmissions can be calculated by

$$\begin{aligned}
e = \frac{1}{K}\sum_{k=0}^{K-1}\Big[&p_k^{\text{cca}}\epsilon^{\text{cca}} + p_k^{\text{tx}}\epsilon^{\text{tx}} + p_k^{\text{bk}}\epsilon^{\text{bk}} + p_k^{\text{rx}}\epsilon^{\text{rx}} \\
&+ \left(1 - p_k^{\text{cca}} - p_k^{\text{tx}} - p_k^{\text{bk}} - p_k^{\text{rx}}\right)\epsilon^o\Big].
\end{aligned} \tag{31}$$

VI. Simulation and Numerical Results

We develop OMNeT++ [34] simulation programs to evaluate the proposed analytical model. In our simulations, the physical layer data rate is 250 kbps. The number of slots in each contention period is $K = 1536$ (corresponding to that the superframe order is 6 according to the IEEE 802.15.4 standard). The maximum backoff stage M is fixed at $BE_{\max} - BE_{\min}$. All the packets are of the same length $L = 6$ (in number of slots). In simulating energy cost, we adopt the following energy model originally proposed for Mica2 wireless nodes and has been used for simulating IEEE 802.15.4 in OMNeT++ [35]. The CPU current when it is in active and inactive modes is 7.6 mA and 0.237 mA, respectively, while the transceiver current when it is in listening/receiving, transmitting (with transmit power 0 dBm), idle and sleep modes is 9.6 mA, 17.0 mA, 1.38 mA and 60 μA, respectively. In the following figures, we report the mean values of 10^5 independent simulation runs for each parameter setting.

A. Model Validation

1) Comparison With Existing Models: Four existing models are chosen for comparison: the saturated-traffic model (SAT model) and the random-traffic model (RND model) both proposed in [14], the unsaturated-traffic model (UnSAT model)

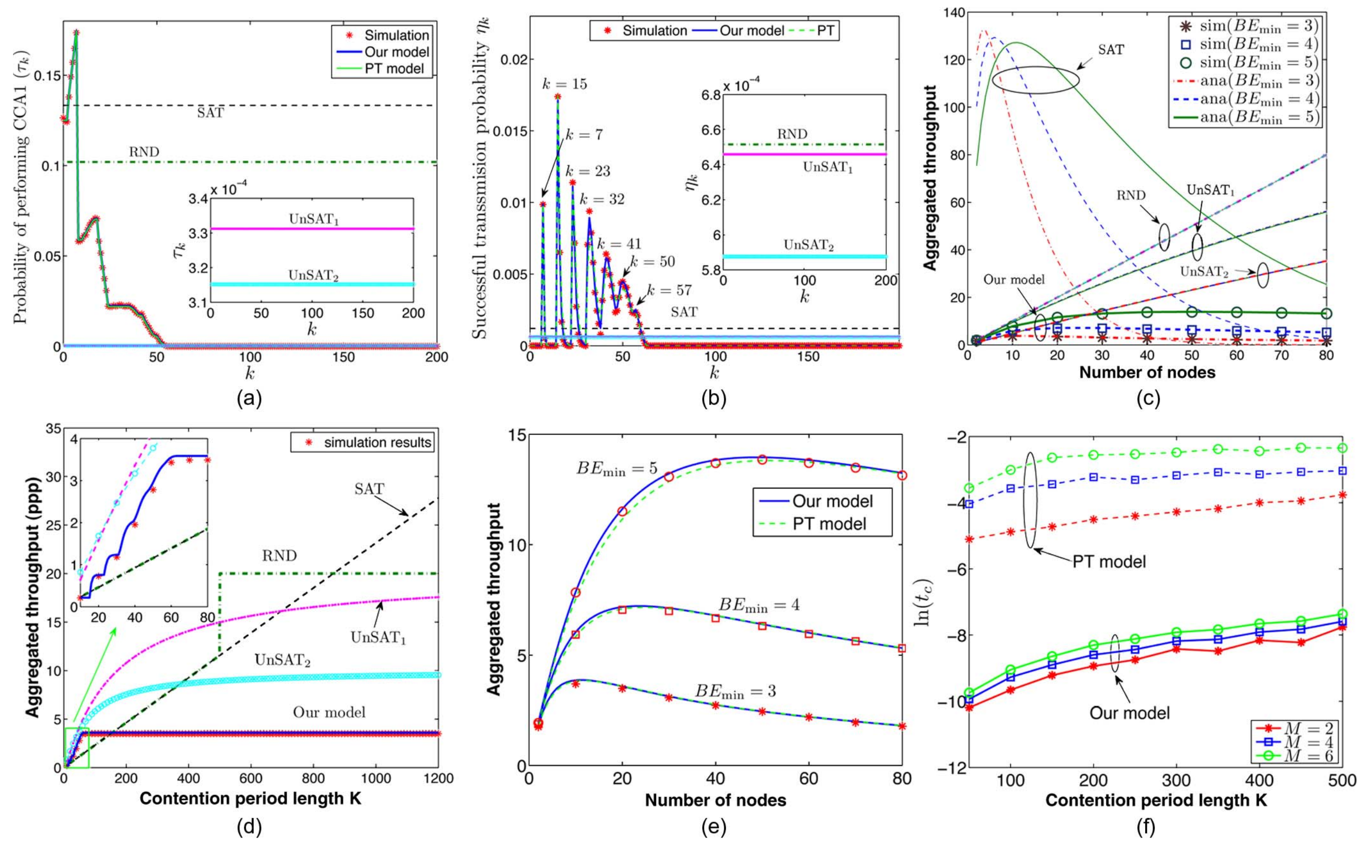


Fig. 5. Comparison with existing models where $C = 0$ and $R = 0$. In (a), (b), (d) and (f), $BE_{\min} = 3, BE_{\max} = 5$ and $n = 20$; in (c) and (e), $K = 1536$. In (f), t_c means computation time in seconds. (a) Probability of performing CCA1 (τ_k). (b) Successful transmission probability η_k. (c) Aggregate throughput vs. n. (d) Aggregate throughput vs. K. (e) Aggregate throughput vs. n. (f) Computation time vs. K.

proposed in [12], and the periodic-traffic model (PT model) proposed in [32]. Because neither retransmission nor protocol re-initialization are taken into account in those models, we fix $R = 0$ and $C = 0$ in our model. For the RND model, since each node has only one packet for transmission in every K slots, the average packet arrival rate is set as $1/K$. The UnSAT model uses three additional parameters X_1, X_2 and X_3 to account for periodic traffic. Specifically, if a node's transmission attempt fails, it shall wait for X_3 number of slots before starting a new transmission attempt. Otherwise, after a transmission, it shall wait for $X_2 + X_3$ (or $X_1 + X_2 + X_3$) number of slots before starting a new transmission attempt if the previous transmission collided (or succeeded). To best fit the periodic traffic considered in our study, we choose to use X_1 and X_3 and consider two settings of the UnSAT model: $X_1 = K$ while $X_2 = X_3 = 0$, in which case the model is called UnSAT_1 (which is also the periodic traffic case considered in the simulations of [12]). Since UnSAT_1 does not consider X_3, a node will immediately start a new transmission attempt if its previous attempt fails. Therefore, we consider another setting that $X_1 = X_3 = K$ while $X_2 = 0$, in which case the model is called UnSAT_2.

In the following, we first demonstrate the MAC transient behavior of a network with $n = 20$ nodes; then we show the network aggregate throughput and analyze the scalability of our model. As shown in Fig. 5(a) and (b), SAT, RND and UnSAT models predict that the probabilities of performing CCA1 (i.e., τ_k) and a successful transmission (i.e., η_k) are independent of time, since they pursue stationary statistics of the MAC performance. In contrast, both our model and the PT model accurately predict the evolutions of τ_k and η_k along time. As shown in Fig. 5(a), after the first backoff which expires with equal probabilities in the first 8 slots ($W(i_0) = 8$), a node which experiences unsuccessful CCA(s) will perform another backoff which expires with probabilities equally distributed in future slots. Continuing in this way, we can predict that τ_k first increases during the first 8 slots. The peak value is reached at $k = 7$, from which a quick drop takes place. The reason is two-fold: the expiration probability of a newly performed backoff decreases as the backoff exponent increases; and the probability of a node turning to sleep increases since it may have already transmitted its packet. Basically, τ_k will ultimately decrease to 0 providing that the contention period is sufficiently long. As shown in Fig. 5(b), it is interesting to observe that the probability η_k oscillates when $k < 60$, and the period is around 8 slots (as indicated by the locations of the peak values shown in this figure). In fact, with τ_k shown in Fig. 5(a), the term $\tau_{k-L-1}(1 - \tau_{k-L-1})^{n-1}$ in (22) does not show an oscillating pattern as η_k does; thus, the probability α_k is the main reason for the oscillation of η_k. From (9), we have

$$\alpha_k - \alpha_{k-1} = \left[1 - (1 - \tau_{k-L-3})^{n-1}\right]\alpha_{k-L-2} - \left[1 - (1 - \tau_{k-2})^{n-1}\right]\alpha_{k-1}.$$

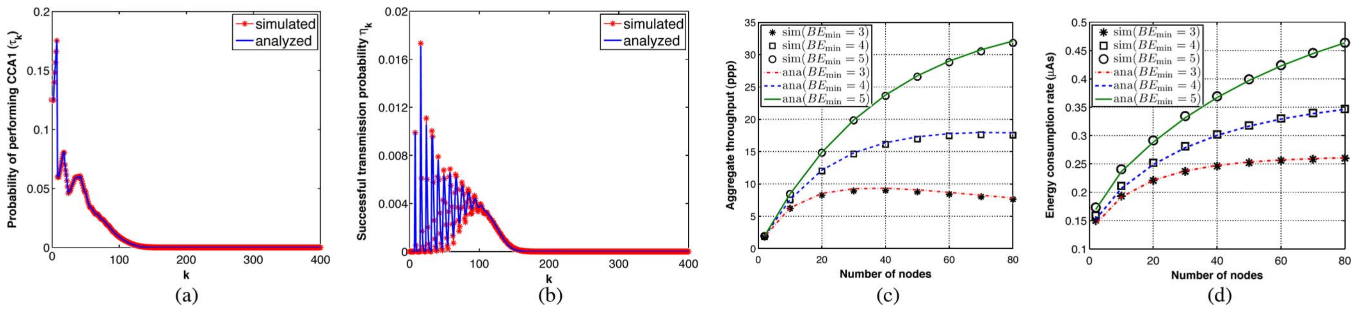


Fig. 6. MAC performance when with re-initialization ($C = 5$) but without retransmission ($R = 0$). In (a) and (b), $BE_{\min} = 3, BE_{\max} = 5$. (a) Probability of performing CCA1 (τ_k). (b) Successful transmission probability η_k. (c) Aggregate throughput. (d) Energy consumption rate (μAs).

Consider the extreme case with $n \to \infty$. Then, the above equation will reduce to $\alpha_k \to \alpha_{k-L-2}$, which clearly indicates that α_k will be periodic and the period is $L + 2 = 8$. Moreover, since $\alpha_1 = 1$ and $\forall k \le 0$, $\alpha_k = 0$, we can see that α_k will oscillate between 0 and 1 with oscillating period 8 in the extreme case. In normal cases, the oscillating behavior of α_k also can be roughly explained by the above equation. However, this equation does not fully capture the exact behavior of α_k (and further η_k), since the recursive updating of τ_k is complicated and the impact of τ_k on calculating α_k and η_k cannot be ignored.

We further compare the performance of these models in terms of aggregate throughput φ (i.e., total number of packets successfully transmitted in one contention period, as defined in (23)) under various parameter settings. The unit of φ is packet per period (ppp). From Fig. 5(c), we can observe that our model is both accurate and scalable. Without retransmissions, increasing the backoff exponents, which in turn increases the average backoff periods and hence can mitigate channel access contention intensity, will benefit the network in terms of increasing the aggregate throughput. As studied in [16], the throughput based on the SAT model shows a parabola shape as n increases, and, due to intensified contention, it drops very fast after its peak value has been reached. For the RND model with $K = 1536$, the average packet arrival rate is significantly low, resulting in that the channel contentions among the nodes are in a very low level and each node can eventually successfully transmit its packet during one period. Therefore, the predicted throughput is almost the same as n increases.

Under periodic traffic, all the nodes will stop contending after some period of time. Therefore, further increasing the contention period length (i.e., K) will not increase the aggregated throughput, as shown in Fig. 5(d). However, increasing K will impact the performance of other models. Specifically, for the SAT model, although it is independent of K, increasing K will increase the aggregated throughput since the latter counts all packets successfully transmitted in one contention period. The reason that the performance of the RND model degrades as K increases is similar, except that, when K is large enough, the aggregated throughput will stay unchanged (as above mentioned, each node will eventually transmit its packet if K is enough long. As shown in Fig. 5(d), when $K > 550$, the aggregated throughput will be 20, the same as the number of nodes). For the UnSAT_1 model, although increasing K will cause longer waiting time (i.e., X_1) and thereby lower rate of transmission attempt, the aggregated throughput will still increase due to the increase of K.[4] The impact of K on UnSAT_2 is similar.

Fig. 5(e) shows that both PT and our models are accurate in terms of aggregated throughput. However, the PT model requires much more computation time than ours, as shown in Fig. 5(f). In fact, the PT model uses a state transition based approach, where a state is defined as $\{BC_c, BC_s, CW, t\}$ representing the tuple of the backoff timer count, backoff stage, the value of CW (1 or 2) and the current slot index. The model should maintain at least the following set of variables in order to capture all the protocol operation details: the probability of each state (i.e., $\Pr\{BC_c = c, BC_s = i, CW = w, t = j\}$), the probabilities of finding the channel busy in the first and second CCAs, and the probability of finding the channel free in two successive slots. Therefore, for each slot t, the number of maintained variables is

$$2\sum_{s=0}^{M} W(i_s) + 3 = 2^{M+2} + 1. \tag{32}$$

In contrast, our model takes a recursive approach to directly update the variables. Note that, by combining (2)–(5), we can see that the probabilities $\{\lambda_{s,b,k}\}$ are just intermediate variables for updating $\{\beta_{s,t}\}$. Therefore, our model only needs to maintain the probabilities $\beta_{s,k}$, τ_k, $\alpha_{1,k}$ and $\alpha_{2,k}$. For each slot k, the number of maintained variables is $M + 4$, which can be significantly smaller than $2^{M+2} + 1$ if M is large. In this sense, our model is light-weight.

In sum, for modeling the MAC performance under periodic traffic, the SAT, RND and UnSAT models become inaccurate, regardless of the backoff exponents and the contention period length. Although the PT model is as accurate as ours, it requires much longer computation time. Therefore, in the following we shall focus on the performance of our model only.

2) When Protocol Re-Initialization Is Allowed: Consider a more complicated scenario in which the protocol can be re-initialized but retransmissions are still disabled. The maximum number of re-initialization times C is set to 5. Fig. 6(a) and (b) show both simulated and analytically predicted probabilities

[4]In [12], the values X_1 and X_3 can be tuned to fit application scenarios. In UnSAT_1 and UnSAT_2, we have used the maximal possible values of X_1 and X_3. For smaller X_1 and X_3, our simulations suggest that the performance gap between UnSAT and our model will be larger.

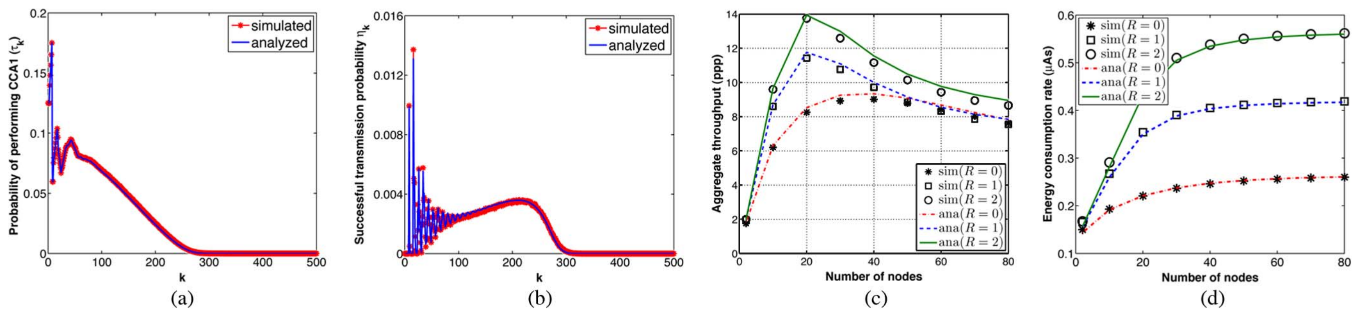


Fig. 7. MAC performance when with retransmission. $C = 5$, $BE_{\min} = 3$ and $BE_{\max} = 5$. In (a) and (b), $R = 2$. (a) Probability of performing CCA1 (τ_k). (b) Successful transmission probability η_k. (c) Aggregate throughput. (d) Energy consumption rate (μAs).

of τ_k and η_k, respectively. The figures clearly show that our model is accurate. Further, the trajectory of the probability η_k delivers the time-dependent distributions of packet successful reception rate and successful reception time. This information is useful for determining the length of the contention period. For example, from Fig. 6(b), all the nodes almost turn to sleep (and hence no packet will be received) after around 150 slots. We call the earliest time in a contention period when all nodes are sleeping the protocol *hibernation time*. If the network configuration is fixed, it is desirable for the coordinator to terminate the contention period after that time in order to save energy consumed for channel listening. On the other hand, a delayed hibernation time usually indicates higher throughput. Comparing Figs. 6(a) and 5(a), we know that by allowing protocol re-initialization, nodes will be given more chances for transmitting their packets, resulting in that the hibernation time moves right along the horizontal axis. By using higher backoff exponents, the backoff period is prolonged and the hibernation time can also be delayed. In both cases, the network throughput is improved, as can be seen by comparing Figs. 5(c) and 6(c).

As mentioned above, increasing the backoff exponents will improve the throughput, but at the cost of higher energy consumption. Such trends are clearly shown in Fig. 6(c) and (d).

3) When Retransmission Is Enabled: We consider that the maximum number of allowable retransmissions R varies in $\{0, 1, 2\}$, and C is set to 5. $L_{turn} = L_{ack} = 1$. As shown in Fig. 7(a) and (b), the shapes of the τ_k and η_k curves are similar as above. However, the protocol hibernation time is postponed because more channel access opportunities are given to nodes whose previous transmissions are collided.

In general, both throughput and energy consumption rate grow as R increases, as shown in Fig. 7(c) and (d). As n increases, the throughput first increases and then slowly declines, yielding parabola-like shapes. It is easy to understand that, when n is small, the contention intensity is low such that having more nodes to access the channel will improve the throughput. As n continues to increase, the channel contention is intensified, which means that, for any node, an increasing amount of bandwidth is wasted in performing backoff, failed CCAs and collided transmissions. For a large n, the channel will be congested for a long time after the beginning of each contention period. Although there are still chances to transmit data after that congestion period, many nodes may use up its limited numbers of protocol re-initializations and retransmissions and will sleep before the congestion period ends. Based on Figs. 6 and 7, when n is large (e.g., $n \geq 40$), increasing the backoff exponents is more efficient than providing more retransmission times. However, in view of the obvious performance improvement by retransmissions when n is relatively small, we still enable retransmissions in the following analysis.

B. Single CCA vs. Double CCA

In the literature, single CCA is usually assumed for modeling the IEEE 802.15.4 MAC performance. In [14], the MAC performance with both single CCA (SS) and double CCAs (DS) is compared under saturated traffic assumptions, and the results prefer DS over SS for higher successful transmission probability while the opposite opinion is given regarding to network throughput. With periodic traffic, our model can readily work with SS after simple modifications. Taking the non-retransmission case for example, we drop all the terms containing $\alpha_{2,k}$ from (2), (3), (8) and (9), since CCA2 is not performed any more. Also, since transmission is now started after one successful CCA, (7) changes to $1 - \alpha_{1,k} = \sum_{l=1}^{L}[1 - (1 - \tau_{k-l})^{n-1}]\alpha_{k-l}$.

To compare SS and DS, we hereby introduce another metric—energy efficiency which is the ratio of aggregate throughput over total energy consumption of all nodes. Numerical results are shown in Fig. 8, where we can see that there are significant gaps between SS and DS in terms of both throughput and energy consumption rate. We observe that DS outperforms SS in terms of throughput under various parameter settings. Compared with DS, a node with SS becomes more easily to transmit packet and the collision probability becomes higher. Since the ACK is sent one slot ($L_{turn} = 1$) after a successful transmission, a packet may also collide with the ACK under SS, which, however, does not happen in DS case. Under saturated traffic, though the reliability of each transmission is lower, SS allows a node to serve more packets than DS, which may in turn increase the network throughput. Whereas, in our case, since there is only one packet to be served in each contention period, lower reliability can imply lower throughput. Fig. 8(b) shows DS incurs much higher energy than SS. From Fig. 8(c), we can see that DS has obvious advantage over SS in terms of energy efficiency when the number of nodes is small. However, when $n \geq 20$, the energy efficiency of the two schemes become very close.

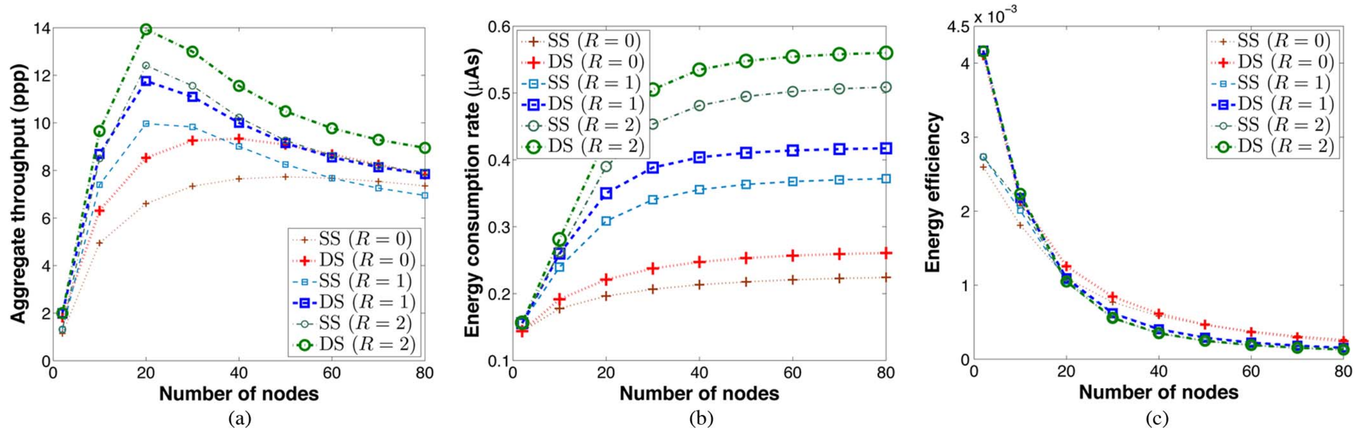


Fig. 8. Performance comparisons between single CCA and double CCAs, where $BE_{\min} = 3$, $BE_{\max} = 5$ and $C = 5$. (a) Aggregate throughput. (b) Energy consumption rate (μAs). (c) Energy efficiency.

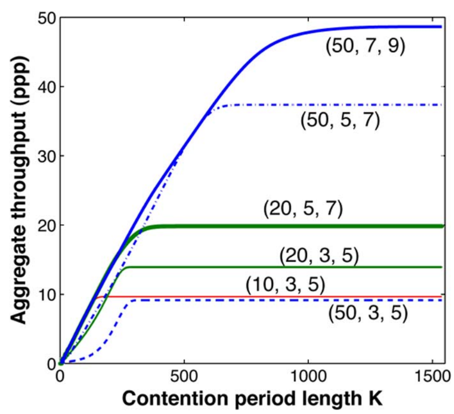


Fig. 9. Impact of contention period length on the throughput. In the figure, the three numbers inside each pair of parentheses mean number of nodes, $BE_{\min}$ and $BE_{\max}$, respectively. We fix $C = 5$ and $R = 2$.

C. The Impact of Contention Period Length

With the proposed analytical model, we can optimize the parameters (e.g., maximum and minimum backoff exponents, maximum number of retransmission times, and the length of the contention period) to improve the MAC performance. Fig. 9 shows the impact of contention period length K on the aggregate throughput of packets. For each parameter setting, we can determine a critical length (denoted as K^c, which is consistent with the value of the protocol hibernation time mentioned above) for the contention period such that the throughput changes very little for any $K > K^c$. Such results can guide the design of the contention period length: the coordinator should, in average, wait at least for a period equal to K^c if it wants to get as many data as possible. Moreover, in order to save the coordinator's energy spent for idle-listening, its average waiting period can be set to K^c.

VII. Discussions

A. Extension to Heterogeneous Scenarios

So far, we have focused on a homogeneous network environment where all nodes are configured with the same set of MAC parameters. In this extension, we consider a class of heterogeneous scenarios where the nodes still apply the IEEE 802.15.4 MAC protocol but use different parameters, e.g., packet length, maximum numbers of re-initialization and retransmission times, backoff exponents, and even contention period. Considering many factors such as mobility and diverse types of the nodes, it would be possible to have such heterogeneous networks. For example, during a contention period, a node may move out of others' communication ranges, which can be viewed as the case that this node uses a shorter contention period.

To accommodate such heterogeneous scenarios, our model should be adapted by carrying out a few modifications as follows. Each of the probabilities is no longer the same for different nodes, so we need to associate with each node a distinct set of the probabilities, e.g., τ_k should be specified to i-th node as $\tau_k^{(i)}$. For each node, depending only on its own historical probabilities, the parallel updating process does not change, while the cross updating process should be modified. Specifically, the conditional probabilities for successful CCAs in (7)–(9) and (19)–(21) should be updated by taking every other node's state into account. For instance, (7) for node 1 will become to $1 - \alpha_{1,k}^{(1)} = \sum_{l=1}^{L}[1 - \prod_{i=2}^{n}(1 - \tau_{k-l-1}^{(i)})]\alpha_{k-l}^{(1)}$. Since each node should be treated separately, as a drawback of the new model, the computational complexity will increase as the network scale increases, which will be investigated in our future work.

B. The Hidden Node Problem

Consider that two nodes are out of each other's communication ranges. If they transmit packets at the same time, collisions may happen because they are not aware of each other's activities even if they perform CCAs. The proposed model has to incorporate the following revisions to accommodate such a hidden node problem. Similar as the above, a distinct set of variables for each node should be defined. Moreover, for each node i, let $\mathcal{N}_i$ and $\mathcal{H}_i$ be the sets of nodes within its interference range and hidden nodes, respectively. Thus, for node i, the probabilities of successful CCAs should take the activities of the nodes only in $\mathcal{N}_i$ into account. For example, without retransmissions, (7) will become: $1 - \alpha_{1,k}^{(i)} = \sum_{l=1}^{L}[1 - \prod_{j\in\mathcal{N}_i}(1 - \tau_{k-l-1}^{(j)})]\alpha_{k-l}^{(i)}$. With this modification, the key impact due to the hidden node problem is on the calculation of successful transmission probability $\eta_k^{(i)}$. Specifically, $\eta_k^{(i)}$ will depend on: the probability that node i transmits a packet, the probability that the channel is occupied by neither the coordinator (which transmits ACKs) nor the nodes in both $\mathcal{N}_i$ and $\mathcal{H}_i$. Since the ACK transmission probability depends on the successful probability of corresponding packet transmitted previously, the key difficulty lies in calculating the probability that none of the nodes in $\mathcal{H}_i$ is transmitting given that node i is transmitting. The challenge is that the transmission probability of the nodes (especially if they lie in each other's communication ranges) in $\mathcal{H}_i$ may be interdependent, which significantly increases the model complexity and is worth investigating in the future.

VIII. Conclusion

We have presented an analytical model for the IEEE 802.15.4 MAC protocol with periodic traffic, a major traffic pattern generated in many practical wireless networks. We establish the model based on recursive updating with well-defined initial conditions. The model includes the MAC characteristics in a variety of scenarios, e.g., whether retransmissions are enabled, with single CCA or double CCAs. Extensive simulations have verified that the model accurately captures performance in terms of both transient and aggregate throughput and energy consumption.

With the proposed model, the time-dependent distributions of packet reception rate and reception time can be explored, which are almost impossible for steady-state based models. Our model is able to accurately predict the impact of many designable parameters such as MAC protocol parameters and contention period length, and thus can facilitate optimization and energy efficient design of IEEE 802.15.4-enabled wireless networks.

References

[1] IEEE Computer Society, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std. 802.15.4-2006, 2006.

[2] "ZigBee Alliance." [Online]. Available: http://www.zigbee.org

[3] J. Song *et al.*, " WirelessHART: Applying wireless technology in real-time industrial process control ," in *Proc. IEEE RTAS*, 2008, pp. 377–386.

[4] Microchip, "MiWi wireless networking protocol stack." [Online]. Available: http://ww1.microchip.com/downloads/en/AppNotes/AN1066-MiWiAppNote.pdf

[5] *Wireless Systems for Industrial Automation: Process Control and Related Applications*, Std. ISA-100.11a-2009, 2009.

[6] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Ind. Electron. Mag.*, vol. 5, no. 4, pp. 23–34, Dec. 2011.

[7] Z. M. Fadlullah *et al.*, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 60–65, Apr. 2011.

[8] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Commun.*, vol. 19, no. 3, pp. 96–104, Jun. 2012.

[9] D. M. Shila *et al.*, "Ghost-in-the-wireless: Energy depletion attack on zigbee," Illinois Instit. Technol., Chicago, IL, USA, Tech. Rep., 2014.

[10] T. Jin, G. Noubir, and B. Sheng, "WiZi-Cloud: Application-transparent dual Zigbee-WiFi radios for low power Internet access," in *Proc. IEEE INFOCOM*, 2011, pp. 1593–1601.

[11] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, 4th Quart. 2014.

[12] S. Pollin *et al.*, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, Sep. 2008.

[13] J. He, Z. Tang, H.-H. Chen, and Q. Zhang, "An accurate and scalable analytical model for IEEE 802.15.4 slotted CSMA/CA networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 440–448, Jan. 2009.

[14] X. Ling, Y. Cheng, J. W. Mark, and X. Shen, "A renewal theory based analytical model for the contention access period of IEEE 802.15.4 MAC," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2340–2349, Jun. 2008.

[15] Y. Cheng, X. Ling, and W. Zhuang, "A protocol-independent approach for analyzing the optimal operation point of CSMA/CA protocols," in *Proc. IEEE INFOCOM*, 2009, pp. 2070–2078.

[16] X. Cao, J. Chen, Y. Sun, and X. Shen, "Maximum throughput of IEEE 802.15.4 enabled wireless sensor networks," in *Proc. IEEE Globecom*, Miami, FL, 2010, pp. 1–5.

[17] I. Ramachandran, A. K. Das, and S. Roy, "Analysis of the contention access period of IEEE 802.15.4 MAC," *ACM Trans. Sens. Netw.*, vol. 3, no. 1, p. 4, Mar. 2007.

[18] C. Jung, H. Hwang, D. Sung, and G. Hwang, "Enhanced Markov chain model and throughput analysis of the slotted CSMA/CA for IEEE 802.15.4 under unsaturated traffic conditions," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 673–678, Jan. 2009.

[19] K. Lu, J. Wang, D. Wu, and Y. Fang, "Performance of a burst-frame-based CSMA/CA protocol: Analysis and enhancement," *Wireless Netw.*, vol. 15, pp. 87–98, 2009.

[20] P. Park, P. D. Marco, P. Soldati, C. Fischione, and K. H. Johansson, "A generalized Markov chain model for effective analysis of slotted IEEE 802.15.4," in *Proc. IEEE MASS*, Macau, China, 2009, pp. 130–139.

[21] F. Ingelrest *et al.*, "SensorScope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sens. Netw.*, vol. 6, no. 2, p. 17, 2010.

[22] A. A. Shabaneh *et al.*, "Review of energy conservation using duty cycling schemes for IEEE 802.15.4 wireless sensor network (WSN)," *Wireless Pers. Commun.*, vol. 77, no. 1, pp. 589–604, Jul. 2014.

[23] F. Zhang, K. Szwaykowska, W. Wolf, and V. Mooney, "Task scheduling for control oriented requirements for cyber-physical systems," in *Proc. IEEE RTSS*, 2008, pp. 47–56.

[24] X. Cao, P. Cheng, J. Chen, and Y. Sun, "An online optimization approach for control and communication codesign in networked cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 439–450, Feb. 2013.

[25] W. Rukpakavong, I. Phillips, and L. Guan, "Neighbour discovery for transmit power adjustment in IEEE 802.15.4 using RSSI," in *Proc. IFIP Int. Conf. NTMS*, 2011, pp. 1–4.

[26] X. Li and D. Hunter, "Four-dimensional Markov chain model of single-hop data aggregation with IEEE 802.15.4 in wireless sensor networks," *Wireless Netw.*, vol. 18, no. 5, pp. 469–479, Jul. 2012.

[27] G. Bianchi, "Performance analysis of the IEEE 802. 11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[28] M. Rabi, L. Stabellini, A. Proutiere, and M. Johansson, "Networked estimation under contention-based medium access," *Int. J. Robust Nonlinear Control*, vol. 20, no. 2, pp. 140–155, 2010.

[29] K. Leibnitz, N. Wakamiya, and M. Murata, "Modeling of IEEE 802.15.4 in a cluster of synchronized sensor nodes," in *Proc. ITC-19*, 2005, pp. 1–10.

[30] F. Shu, T. Sakurai, M. Zukerman, and H. Vu, "Packet loss analysis of the IEEE 802.15.4 MAC without acknowledgements," *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 79–81, Jan. 2007.

[31] M. Sayad Haghighi, K. Mohamedpour, V. Varadharajan, and B. Quinn, "Stochastic modeling of hello flooding in slotted CSMA/CA wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1185–1199, Dec. 2011.

[32] C. Buratti, "Performance analysis of IEEE 802.15.4 beacon-enabled mode," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 2031–2045, May 2010.

[33] Y. Liu, C. Yuen, X. Cao, N. Ul Hassan, and J. Chen, "Design of a scalable hybrid MAC protocol for heterogeneous M2M networks," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 99–111, Feb. 2014.

[34] "OMNeT++: A discrete event simulation system." [Online]. Available: http://www.omnetpp.org

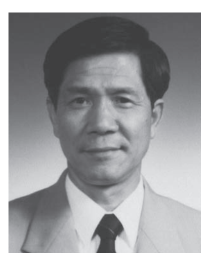
[35] F. Chen, I. Dietrich, R. German, and F. Dressler, "An energy model for simulation studies of wireless sensor networks using OMNeT++," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 32, no. 2, pp. 133–138, 2009.

Xianghui Cao (S'08–M'11) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. During 2007–2009, he was a Visiting Scholar with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. Since 2012, he has been a Senior Research Associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. In 2015, he joined the School of Automation, Southeast University, Nanjing, China, as an Associate Professor. His research interests include cyberphysical systems, wireless network performance analysis, networked estimation and control, and network security. He is a TPC Member for a number of conferences, including IEEE Globecom, IEEE ICC, and IEEE VTC and a Publicity Cochair for ACM MobiHoc 2015. He serves as an Associate Editor of several journals, including *KSII Transactions on Internet and Information Systems, Security and Communication Networks (Wiley)*, and the *International Journal of Ad Hoc and Ubiquitous Computing*. He was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc 2014.

Jiming Chen (M'08–SM'11) received the B.Sc. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2000 and 2005, respectively. He was a Visiting Researcher at INRIA in 2006, at the National University of Singapore in 2007, and at the University of Waterloo from 2008 to 2010. He is currently a Full Professor with the Department of Control Science and Engineering, the Coordinator of the Group of Networked Sensing and Control in the State Key Laboratory of Industrial Control Technology, and the Vice Director of the Institute of Industrial Process Control at Zhejiang University. His research interests include sensor networks and networked control. He currently serves as an Associate Editor of several international journals, including the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *IEEE Network*, and the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. He has been a Guest Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and other journals.

Yu Cheng (S'01–M'04–SM'09) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003. From September 2004 to July 2006, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA, where he is currently an Associate Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking. He served as a Cochair for the Wireless Networking Symposium of IEEE ICC 2009; a Cochair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011; a Cochair for the Signal Processing for Communications Symposium of IEEE ICC 2012; a Cochair for the Ad Hoc and Sensor Networking Symposium of IEEE GLOBECOM 2013; and a Technical Program Committee Cochair for WASA 2011, ICNC 2015, and IEEE/CIC ICCC 2015. He is a Founding Vice Chair of the IEEE ComSoc Technical Subcommittee on Green Communications and Computing. He is also an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the New Books & Multimedia Column Editor of *IEEE Network*. He was a recipient of the Best Paper Award from the conferences QShine 2007 and IEEE ICC 2011 and the Best Paper Runner-Up Award from ACM MobiHoc 2014. He was also a recipient of the National Science Foundation CAREER Award in 2011 and the IIT Sigma Xi Research Award in the junior faculty division in 2013.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees from Rutgers University, New Brunswick, NJ, USA, in 1987 and 1990, respectively, all in electrical engineering. He is currently a Professor and a University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. He was the Associate Chair for Graduate Studies from 2004 to 2008. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an Elected Member of the IEEE ComSoc Board of Governors and the Chair of the Distinguished Lecturers Selection Committee. He is also a Fellow of IEEE, The Engineering Institute of Canada, and the Canadian Academy of Engineering and a Distinguished Lecturer of the IEEE Vehicular Technology and IEEE Communications Societies. He served as the Technical Program Committee Chair/Cochair for IEEE Infocom'14 and IEEE VTC'10 Fall; the Symposia Chair for IEEE ICC'10; the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08; the Technical Program Committee Chair for IEEE Globecom'07; the General Cochair for ACM Mobihoc'15, Chinacom'07, and QShine'06; and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications and P2P Communications and Networking. He also serves/served as the Editor-in-Chief of IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Computer Networks*, and *ACM/Wireless Networks*; and the Guest Editor of IEEE JSAC, *IEEE Wireless Communications*, *IEEE Communications Magazine*, and *ACM Mobile Networks and Applications*. He was a recipient of the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo; the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada; and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is also a Registered Professional Engineer in Ontario, Canada.



Youxian Sun received the Diploma degree from Zhejiang University, Hangzhou, China, in 1964. In 1964, he joined the Department of Chemical Engineering, Zhejiang University. From 1984 to 1987, he was an Alexander Von Humboldt Research Fellow and a Visiting Associate Professor at the University of Stuttgart, Stuttgart, Germany. Since 1988, he has been a Full Professor with Zhejiang University, where he was elevated to an Academician of the Chinese Academy of Engineering, Beijing, China, in 1995 and is currently the Director of the Institute of Industrial Process Control and the National Engineering Research Center of Industrial Automation. He has authored/co-authored over 450 journal and conference proceeding papers. His current research interests include the modeling, control, and optimization of complex systems and robust control design and its applications.