

# CAMA: Efficient Modeling of the Capture Effect for Low-Power Wireless Networks

BEHNAM DEZFOULI and MARJAN RADI, Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia; Networking Protocols Department, Institute for Infocomm Research (I<sup>2</sup>R), A\*STAR, Singapore  
KAMIN WHITEHOUSE, Department of Computer Science, University of Virginia, USA  
SHUKOR ABD RAZAK, Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia  
HWEE-PINK TAN, Networking Protocols Department, Institute for Infocomm Research (I<sup>2</sup>R), A\*STAR, Singapore

Network simulation is an essential tool for the design and evaluation of wireless network protocols, and realistic channel modeling is essential for meaningful analysis. Recently, several network protocols have demonstrated substantial network performance improvements by exploiting the capture effect, but existing models of the capture effect are still not adequate for protocol simulation and analysis. Physical-level models that calculate the signal-to-interference-plus-noise ratio (SINR) for every incoming bit are too slow to be used for large-scale or long-term networking experiments, and link-level models such as those currently used by the NS2 simulator do not accurately predict protocol performance. In this article, we propose a new technique called the *capture modeling algorithm* (CAMA) that provides the simulation fidelity of physical-level models while achieving the simulation time of link-level models. We confirm the validity of CAMA through comparison with the empirical traces of the experiments conducted by various numbers of CC1000 and CC2420-based nodes in different scenarios. Our results indicate that CAMA can accurately predict the packet reception, corruption, and collision detection rates of real radios, while existing models currently used by the NS2 simulator produce substantial prediction error.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*; C.4 [Performance of Systems]: Modeling Techniques; I.6.1 [Simulation and Modeling]: Simulation Theory—*Systems theory*

General Terms: Design, Algorithms, Performance, Measurement, Verification

Additional Key Words and Phrases: Wireless sensor networks, radio interference, packet collisions

## ACM Reference Format:

Behnam Dezfouli, Marjan Radi, Kamin Whitehouse, Shukor Abd Razak, and Hwee-Pink Tan. 2014. CAMA: Efficient modeling of the capture effect for low-power wireless networks. *ACM Trans. Sensor Netw.* 11, 1, Article 20 (July 2014), 43 pages.  
DOI: <http://dx.doi.org/10.1145/2629352>

---

Authors' addresses: B. Dezfouli, M. Radi, Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor 81310, Malaysia; Networking Protocols Department, Institute for Infocomm Research (I<sup>2</sup>R), A\*STAR, Singapore 138632, Singapore; email: {dezfouli, radi}@ieee.org; K. Whitehouse, Department of Computer Science, University of Virginia, Charlottesville, VA, USA; email: whitehouse@virginia.edu; S. A. Razak, Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor, Malaysia; email: shukorar@utm.my; H.-P. Tan, Institute for Infocomm Research (I<sup>2</sup>R), A\*STAR, Singapore 138632, Singapore; email: hptan@i2r.a-star.edu.sg.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2014 ACM 1550-4859/2014/07-ART20 \$15.00

DOI: <http://dx.doi.org/10.1145/2629352>

## 1. INTRODUCTION

Low-power wireless communication is enabling an endless number of new applications that use mobile, mesh, and ad hoc networks to connect embedded devices. As wireless communication evolves to support this expanding horizon of applications, higher-layer protocols are increasingly exploiting subtle physical-layer protocols in order to realize performance gains. For example, *the capture effect*, also called cochannel interference tolerance [Kim and Lee 1999], is the ability of a radio to receive a signal even in the presence of interfering signals, as long as its signal-to-interference-plus-noise ratio (SINR) is above a threshold  $T$ . The capture effect is supported by the most common low-power wireless transceivers, and it has a large effect on the packet reception rate when packets are subject to collision. Recent studies have revealed that this physical-layer property has significant performance impact on the performance of several higher-layer protocols, such as increasing the packet reception probability when multiple packets collide at a receiver; causing packet reception unfairness [Son et al. 2006; Whitehouse et al. 2005; Gezer et al. 2010; Firner et al. 2010; Lee et al. 2007]; enabling collision detection and packet recovery during a collision [Whitehouse et al. 2005; Yun and Seo 2007; Dezfouli et al. 2014b]; reducing the packet losses that are caused by collision [Firner et al. 2010]; enhancing flood propagation [Lu and Whitehouse 2009; Dutta et al. 2010]; enabling acknowledgments to broadcast packets [Dutta et al. 2010]; and improving packet reception through interference management [Xu et al. 2010]. Consequently, the capture effect in particular has recently become an important component in protocol design, specifically for low-power radio networks.

Unfortunately, experimental evaluation is costly and time consuming, particularly for dense and large-scale networks. Furthermore, the true channel conditions cannot be easily controlled in an empirical experiment [Dezfouli et al. 2014a]. Network simulation allows the experimenter to perform repeatable experiments and to use a diverse set of network topologies for exhaustive testing and analysis. However, existing models of the capture effect are insufficient to explore cross-layer optimizations such as those described previously. There are two approaches to design a capture-enabled model for packet reception: (1) link-level modeling (packet-level modeling) is computationally efficient but does not accurately reproduce many aspects of the capture effect, and (2) physical-level modeling has high fidelity but is computationally infeasible for simulating long, multipacket network traces or large-scale networks. Link-level models typically perform one or two SINR computations per packet. For example, the *capture threshold model* (CTM) and *additive interference model* (AIM)<sup>1</sup> utilized in the NS2 network simulator [NS-2 2014] decide about packet reception and collision on a per-packet basis. From the simulation point of view, packet-level modeling requires a lightweight algorithm and provides fast simulation. Unfortunately, this approach does not permit simulation and analysis of subpacket operations. For example, wireless collision and partial packet recovery techniques depend on the SINR values during specific fields or bits of the packet, and therefore cannot be analyzed with packet-level SINR estimates. Consequently, link-level models cannot be used to develop, test, or analyze cross-layer protocol optimizations. In addition, link-level capture models do not necessarily translate from one link-layer protocol to another, because the impact of the capture effect changes with each protocol. For example, when long preambles are used with the low-power listening protocols [Polastre et al. 2004], packets may be received even in the presence of partial corruption during their preamble, whereas other protocols would lose packets with any level of preamble corruption. Therefore, MAC protocols and subpacket link-layer operations often require custom link-level capture

<sup>1</sup>In this paper, we refer to NS2.32 and its predecessors as NS2 CTM, and NS2.33 and its successors as NS2 AIM.

models, which makes it difficult to compare different techniques and protocols in simulation. On the other hand, physical-level modeling performs SINR calculations at the bit level, therefore providing the upper-layer protocols with a high fidelity estimate of the capture effect's impact. Unfortunately, physical-level modeling is very expensive from the computational point of view: it involves a comparison of the signal strength of every possible transmitter at all possible receivers for every bit. Therefore, the execution time scales with the number of bits per packet, the number of packets sent, and the *square* of the number of nodes in the network. This approach is particularly infeasible in dense mesh networks with low-power MAC protocols, many of which use extremely long preambles or packet cycling techniques. This approach is not scalable for protocol-level analysis where thousands of packets are being sent in networks with hundreds or possibly thousands of nodes, and each packet contains hundreds to thousands of bits. Consequently, new techniques are needed to incorporate realistic models of capture into existing simulation tools.

In this paper, we propose a new simulation technique that moves the modeling of the capture effect down to the physical layer while achieving simulation times comparable to link-level models. This technique is called CAMA (Capture Modeling Algorithm). The proposed technique is independent of the MAC-layer implementation and can be used to simulate and analyze bit-level, byte-level, or field-level packet operations such as collision detection or partial packet recovery. CAMA's efficiency is achieved through two basic mechanisms. First, we prove an intuitive and particularly useful property of SINR: at any time instance, at most one signal's SINR value can be higher than  $T$  for a given receiver, where  $T$  is the SINR threshold required for successful signal reception. While we use this property to dramatically reduce the number of SINR computations required for physical-level modeling of the capture effect, it also reduces the overhead of the simulation engine's event management. Once we find the captured signal for each receiver, we do not perform SINR computations for any other signals, thereby making the number of comparisons linear with the number of nodes, instead of quadratic. Second, we leverage a physical-layer property of low-power radios: the need to use the preamble for radio synchronization. Similar to the actual radio transceivers where correct packet reception requires successful preamble reception, CAMA does not start data reception unless synchronization is successful. Radio transceivers define a minimum number of bits required for successful packet synchronization, referred to as the *settling bits*, and CAMA decides about radio synchronization only with respect to the last settling bits of the preamble. This approach makes capture modeling time independent of the preamble size, which is particularly important, for example, when long preambles are used by low-power listening MAC protocols. Furthermore, if synchronization is not successful, then the signal is thereafter considered interference, precluding the need for any additional SINR computations for that signal.

We implement CAMA with the OMNeT++ simulation framework using an efficient state machine design. Instead of continuously monitoring and reprocessing all SINR values, we embed signal strength information into the packets and evaluate SINR at specific times (e.g., at each bit or byte boundary). Our state machine has only four states and is integrated with the simulator's discrete-event messaging mechanisms to efficiently schedule SINR evaluations during packet reception. We validated CAMA using two low-power radios with different physical layers: the CC1000 FSK radio [Chipcon CC1000 2014] operating in the sub-1GHz band, and the 802.15.4-compliant CC2420 DSSS-OQPSK radio operating in the 2.4GHz band [Chipcon CC2420 2014]. We use small-scale experiments (with three and four nodes) to validate the accuracy of CAMA in reproducing the empirical traces obtained by the variations in transmission time and signal power. In addition, we use larger-scale experiments (36 nodes) to validate the accuracy of CAMA in real applications. We also analyze the capture models currently

used in NS2 AIM and NS2 CTM. In order to provide a fair comparison, we augmented the capture model of NS2 AIM with support for receiver sensitivity computations and collision detection. In contrast to CAMA, our results confirm considerable inaccuracy of the NS2 capture models. In terms of performance, we found that CAMA is slower than existing NS2 models, but always less than 30% slower and only when simulating with a minimum-size preamble (6 bytes in our experiments). As the preamble size grows, the simulation speed of CAMA quickly approaches that of NS2's link-level capture models.

Due to the lack of accurate capture-enabled simulation tools, there is currently no analysis on the influence of various network parameters on the capture effect. In this paper, we demonstrate how CAMA can be used to analyze and characterize packet collision and the capture effect in sensor networks. More specifically, we perform a sensitivity analysis of the capture effect to various network parameters. Such analysis allows protocol designers to predict the effects of design approaches and network parameters on the performance of cross-layer protocol design.

## 2. RELATED WORK

In this section, first, we review the efforts made to model the capture effect. Then, we present the studies that have investigated the capture effect in wireless networks.

### 2.1. Modeling the Capture Effect

The rudimentary proposed capture models use the arrival time of the second packet [Davis and Gronemeyer 1980], the reception power of the second packet [Arnbak and Van Blitterswijk 1987], or a combination of time and power [Cheun and Kim 1998] to determine whether the firstly arrived packet involved in a collision can be received. These models assume a specific time interval after the start of the first packet during which every collision can be resolved in favor of the first packet. Therefore, they disregard the ability of the radio transceivers to be retrained with a new incoming signal during a packet reception [Ware et al. 2001; Whitehouse et al. 2005]. More specifically, the empirical studies of Kochut et al. [2004] and Lee et al. [2007] showed that the capture effect not only happens in stronger-first collisions but also exists in stronger-last collisions.

In NS2 CTM, when a sample packet  $S_i$  arrives, the received signal strength of the packet (denoted by  $\Psi(S_i)$ ) is first compared with the receiver sensitivity value ( $RxThresh$ ) to check whether it is eligible for reception. If  $\Psi(S_i) > RxThresh$  and no interference occurs during the packet reception, the packet is assumed to be successfully received. If a new signal  $S_n$  arrives during a packet reception, NS2 CTM uses the *capture threshold model*: it compares the reception power of the packet currently being received with the reception power of the newly arrived packet. If  $\Psi(S_i)/\Psi(S_n) \geq CpThresh$ , packet reception is continued; otherwise, it discards both packets. The state machine of this model is demonstrated in Figure 1. The drawbacks of this model can be summarized as follows [Lee et al. 2010; Iyer et al. 2009; Al-Bado et al. 2012; Hamida et al. 2009; Chen et al. 2006, 2007; Hu and Hou 2005]: (1) A packet reception requires the radio to be in the *Searching* state upon its arrival. Therefore, this model does not support stronger-last capture, because it does not start a packet reception when the arrived signal collides with another signal. (2) There is no difference between preamble reception and MPDU reception. Therefore, this algorithm does not model radio synchronization. (3) The effect of additive interference on packet reception is neglected. (4) As this is a link-level model, it does not support collision detection and partial packet recovery. (5) This model neglects signal variations during packet reception.

Based on the observations reported by Kochut et al. [2004], the inaccuracy of NS2 CTM for the performance evaluation of 802.11 networks has been specifically pointed

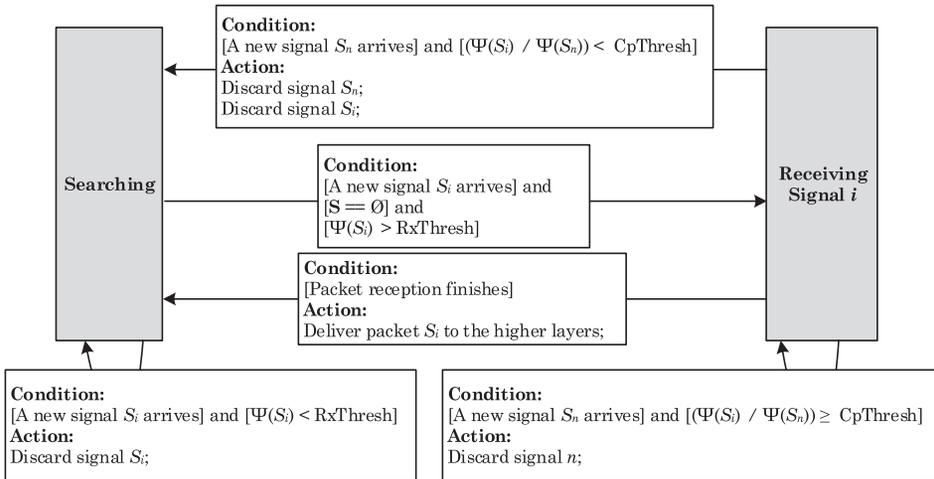


Fig. 1. The state machine of the capture model employed by NS2 CTM.

out by Chen et al. [2007]. To address the aforementioned problems, they proposed a new link-level model to support the SINR model and stronger-last capture in 802.11 networks. As this model considers additive interference, it is referred to as NS2 AIM in this paper. The physical layer includes the *PHY State Manager* sublayer that implements a state machine with the following states. The state machine is in the *Searching* state when there is no signal at the radio or when the power of each signal is lower than the required threshold value. If a signal arrives and its SINR is higher than the threshold value, the state machine goes into the *PreRXing* state. If this signal's SINR remains higher than the threshold value for the preamble duration, the state changes into *RXing*, during which the packet body can be received. Arrival of a signal during the *PreRXing* or *RXing* states may start a new preamble reception duration if the new signal's SINR is higher than the threshold value. The state machine of this model is shown in Figure 2. Considering a mobile ad hoc network (MANET) scenario, Chen et al. [2007] have evaluated this capture model against NS2 CTM. Their results validate the influence of the capture effect on packet reception rate, especially in dense networks where the hidden-terminal problem occurs frequently. Nevertheless, the main shortcomings of this model can be identified as follows [Lee et al. 2010; Iyer et al. 2009; Al-Bado et al. 2012; Hamida et al. 2009; Chen et al. 2006, 2007; Hu and Hou 2005]: (1) Although this model supports stronger-first and stronger-last captures, it goes into the *PreRXing* state if a signal's SINR value is higher than a certain threshold upon its arrival. Therefore, when the preamble size is larger than the settling bits, a packet cannot be received if it cannot provide enough SINR upon its arrival but its SINR value is large enough during the settling bits. This happens when the preamble bytes of a packet are partially collided and overpowered by one or multiple signals. (2) As this is a link-level model, it does not support partial packet recovery and collision detection. In order to overcome these problems, separate state machines are required for each incoming packet instead of utilizing one state machine for the packet currently being received. Moreover, extra states should be added to the state machines to support partial packet recovery. These accuracy improvements require considerable changes to the physical layer and induce significant overhead and complexity. (3) Since SINR evaluations are performed at signal arrival times, signal variations during packet reception cannot be precisely modeled. Specifically, instead

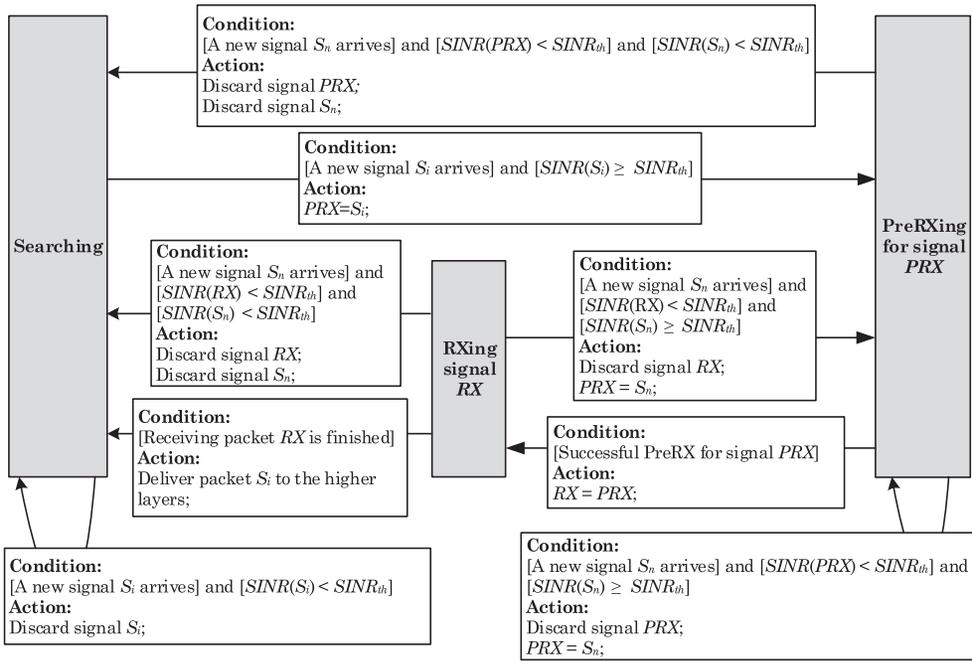


Fig. 2. The state machine of the capture model employed by NS2 AIM.

of using a probabilistic packet reception approach, threshold SINR values have been defined to decide about preamble and MPDU reception.

The approach employed by Lu and Whitehouse [2009] is to use the traces of packet reception from a 48-node testbed to construct a capture-aware simulator. For each node, the *transmitter sets*, in which multiple neighboring nodes transmit concurrently, are identified. Then, the empirical traces are used to determine whether a receiver can receive a packet from each of its transmitter sets. These results are saved into a *capture map* that acts as a look-up table during the simulation process to find out about packet reception at the nodes. In order to build a large-scale simulation platform, they have used multiple copies of the testbed and virtually tied them together. Since this simulator relies on experimental traces, it is only valid for the same network configuration. Specifically, any change in the network deployment, radio parameters, or packet transmission makes the simulator invalid. Therefore, this approach does not provide a flexible simulation platform because experimental traces are required prior to simulator development. Reis et al. [2006] and Al-Bado et al. [2012] also tried to improve packet reception accuracy through adding empirical measurements to a simulation platform. Unfortunately, no higher accuracy is observable with these contributions when the traffic pattern or MAC protocol changes. A more general approach is proposed by Zhang et al. [2007]. Although their approach can be used for performance evaluation of a given network under various scenarios, the model is only valid for one network, and any change in node placement or number of nodes invalidates the model.

As an alternative approach, analytical models have been proposed for evaluating network parameters in 802.11 [Abukharis et al. 2011; Daneshgaran et al. 2008; Hadzi-Velkov and Spasenovski 2002] and low-power [Gezer et al. 2010] wireless networks. However, as these approaches try to augment a specific MAC model with a capture-enabled physical model, they are not general and should only be used under specific assumptions.

Table I. Notations and their Descriptions

Description	Symbol
Set of the signals currently being received at a node	$\mathbf{S}$
A signal transmitted by node $i$	$S_i$
Output power of node $i$	$\Omega_i$
Received power at node $j$ , corresponding to signal $S_i$	$\Psi_j(S_i)$
Average noise floor	$\tilde{\Psi}$
Number of settling bits	$L_{settle}$
SINR value at node $j$ , corresponding to signal $S_i$	$SINR_j(S_i)$
SINR value at node $j$ , corresponding to signal $S_i$ , at time $t$	$SINR_j(S_i, t)$
Received signal strength	$RSSI$
Noise bandwidth	$B$
Radio speed	$R$
Path-loss exponent	$\eta$
Standard deviation of multipath channel variations	$\sigma_{ch}$
SINR threshold	$SINR_{th}$
Initial contention window	$CW_i$
Congestion contention window	$CW_c$
Carrier-sensing threshold	$CS_{th}$

## 2.2. Analyzing the Capture Effect

Regarding the investigation of the capture effect in wireless networks, some works have revealed the influence of the capture effect on packet reception, throughput, delay, and fairness [Abukharis et al. 2011; Kochut et al. 2004; Lee et al. 2007; Ganu et al. 2006]. However, they have not studied the benefits of collision detection and recovery. Besides, there is no analysis on the influence of various network parameters on the capture effect. Son et al. [2006] have conducted empirical studies in which a sender transmits to a receiver in the presence of one or more interferers. Through this study, they confirm and characterize the capture effect with CC1000 radio, as well as confirming the existence of the capture effect with CC2420 radio. The most significant study of the capture effect in low-power wireless networks was conducted by Whitehouse et al. [2005]. Through providing a decoding scheme at the MAC layer, they enabled Mica2 nodes to support the capture effect and packet recovery. To analyze the influence of the capture effect on packet reception rate, they used a three-node experiment in which two nodes transmit their data to a common receiver with a predetermined timing difference. Whitehouse et al. also provided some preliminary results about the influence of transmission power on collision detection and recovery in a 36-node testbed. As the findings of Son et al. and Whitehouse et al. merely rely on empirical studies, no simulation implementation is provided by them.

## 3. BACKGROUND

Realizing the operation of radio transceivers and low-power wireless links is essential for accurate modeling of wireless communications. In this section, we introduce radio synchronization and the capture effect with regard to the SINR model. In addition, we study the essential models of link unreliability and asymmetry, which have been integrated with CAMA and used in the developed simulation tool of this article. The concepts presented in this section are also required for the investigations presented in Section 7. Table I shows the notations used in this article.

### 3.1. Radio Synchronization

In wireless communications, the radio should obtain certain information regarding the incoming signal before receiving data bytes. To this aim, data packets begin with a predefined training sequence, called *preamble*, which enables the receiver to learn

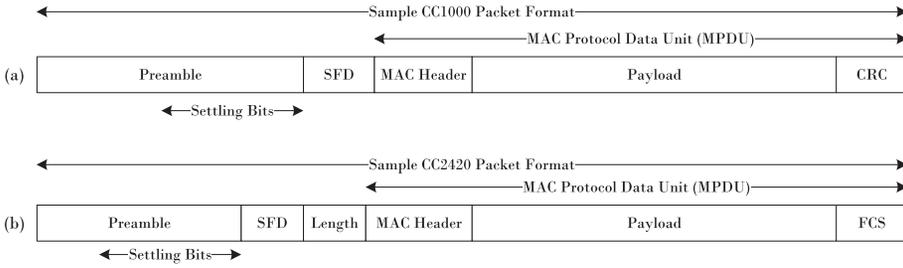


Fig. 3. Sample packet formats using (a) CC1000 and (b) CC2420 radios. We refer to the last  $L_{settle}$  bits of the preamble field as the *settling bits*.

about the parameters of the transmitter and lock on the incoming signal. In addition to the preamble bytes, the receiver should be able to detect the start of a data frame. Therefore, specific frame synchronization bytes should be transmitted between the preamble bytes and data bytes. These bytes are usually called *sync word*, *start bytes*, or *start of frame delimiter* (SFD). The length and formatting of the preamble and start bytes may be different for various radios, and also depend on the MAC protocol implementation. If at least  $L_{settle}$  bits are required for a proper synchronization, we refer to the last  $L_{settle}$  bits of the preamble field as the *settling bits*. Figure 3 shows sample packet formats and the positions of the settling bits.

Using CC1000 radio, successful synchronization requires 98 bauds (equal to 49 bits in Manchester mode). The default number of preamble bytes in CC2420 is 4. While this value provides compliance with the 802.15.4 standard, employing a lower number of preamble bytes is not recommended because it causes incorrect frame detection. The number of settling bits has been reduced to as low as 4 bits in newer radios such as CC1120 [CC1120 2014] to decrease the cost of packet transmission, especially with low-power MAC protocols.

Although using the minimum required preamble size (i.e.,  $L_{settle}$ ) reduces packet transmission overhead, usually longer preamble sizes are used in real applications. Specifically, since most of the proposed MAC protocols for low-power wireless networks utilize the low-power listening technique for duty cycle reduction, the length of the preamble field is usually much longer than the settling bits.

Notice that radio synchronization not only happens during the settling bits but also begins as soon as the radio starts receiving the preamble bits with enough SINR. However, since the settling bits reflect the minimum number of required bits for synchronization, in this article we only consider radio synchronization during the settling bits. We will show in Sections 4.3 and 6 that this assumption results in correct packet reception and capture modeling.

### 3.2. Interference Model

As wireless nodes share the same transmission medium, internode interference affects packet reception performance. Therefore, various interference models have been proposed [Cardieri 2010; Dezfouli et al. 2014a]: (1) *interference range model*, (2) *protocol model*, (3) *capture threshold model*, and (4) *signal-to-interference-plus-noise ratio (SINR) model* (a.k.a., *physical interference model* [Gupta and Kumar 2000]). Among these models, the SINR model provides the highest accuracy. Although the empirical studies of Son et al. [2006] with Mica2 nodes showed that the interference power is lower than the addition of interfering signals, the studies of Iyer et al. [2009] and Maheshwari et al. [2008] clearly confirm the additivity of interference. These studies state that the nonadditivity reported by Son et al. [2006] was due to their system noise.

Let  $\mathbf{S} = \{S_1, S_2, S_3, \dots, S_n\}$  represent the set of the signals currently being received at a node; the SINR value for signal  $S_i$  is defined as

$$SINR(S_i) = \frac{\Psi(S_i)}{\bar{\Psi} + \sum_{S_j \in \mathbf{S} \setminus \{S_i\}} \Psi(S_j)}, \quad (1)$$

where  $\Psi(S_k)$  is the received power corresponding to signal  $S_k$ , and  $\bar{\Psi}$  is the noise power. Having an acceptable SINR level (i.e., higher than a certain threshold), radio can be synchronized with the incoming signal.

### 3.3. The Capture Effect

When the radio is synchronized with a signal, a new signal arrival may result in one of the three following situations:

- If the SINR of the synchronized signal is high enough to maintain radio synchronization, the radio continues packet reception.
- If the SINR of the newly arrived signal is higher than a certain threshold, the radio starts synchronizing with the new signal.
- If the SINR values of both signals are lower than the threshold value, destructive collision happens and no packet can be received.

In the following, we present sample cases for the first two scenarios to show how packet reception can be achieved when multiple signals exist at a receiver.

A stronger-first capture scenario is demonstrated in Figure 4(a). When signal  $S_1$  arrives, it can synchronize the radio since  $SINR(S_1, t_0) > SINR_{th}$ . At  $t_1$ , signal  $S_2$  arrives while packet  $S_1$  is being received. Suppose  $S_1$ 's power is higher than  $S_2$ 's power, and  $SINR(S_1, t_1) > SINR_{th}$  (we will show in Section 4.2 that if  $SINR(S_1, t_1) > SINR_{th}$ , then  $SINR(S_2, t_1) < SINR_{th}$ ). Therefore, the arrival of  $S_2$  does not change the radio synchronization, and the radio continues receiving  $S_1$ . In this case, the MAC layer receives the data bytes of  $S_1$  and is unaware of the existence of  $S_2$ .

Now assume that  $S_2$ 's power is higher than  $S_1$ 's power, and  $SINR(S_2, t_1) > SINR_{th}$ . Figure 4(b) shows this stronger-last capture scenario. When signal  $S_2$  arrives at  $t_1$ , radio is synchronized with signal  $S_1$ ; however, since  $SINR(S_2, t_1) > SINR_{th}$ , the receiver starts synchronizing with  $S_2$ . Using CC1000 radio, the MAC layer receives the SFD bytes of  $S_2$  while it was expecting to receive the data bytes of  $S_1$ . This enables the software to perform collision detection. Using CC2420 radio, this condition causes the SFD pin of this chip to go low and high during an ongoing reception. While this feature of CC2420 simplifies collision detection, this radio also allows reading of the received bytes and performing collision detection by software.

It should be noted that the capture effect does not require the second signal to arrive after the preamble bytes of the first packet. In fact, as soon as a signal can provide an acceptable SINR value during its preamble duration, and regardless of the number of interfering signals, the radio starts synchronizing with that signal. However, collision detection capability depends on the capture timing. For example, in Figure 4(b), if  $S_2$  arrives during the preamble reception of  $S_1$ , the collision cannot be detected because the MAC layer cannot discriminate between the preamble bytes of  $S_1$  and  $S_2$ . Hence, the potential collisions for collision detection are those stronger-last collisions that happen after SFD reception.

### 3.4. Modeling Link Unreliability and Asymmetry

Studies on low-power radio communication have revealed three main characteristics: path loss, multipath fading, and hardware heterogeneity. In this section, we present these models and show how they can be integrated with the SINR model.

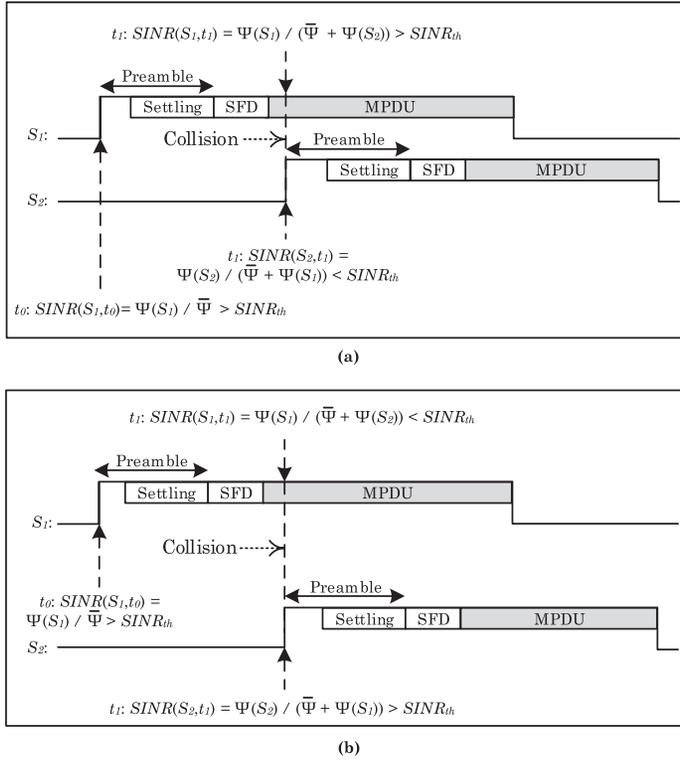


Fig. 4. (a) A stronger-first capture scenario. Signal  $S_2$  arrives at  $t_1$ . Because  $SINR(S_1, t_1) > SINR_{th}$ , the radio continues receiving  $S_1$ . (b) A stronger-last capture scenario. Because  $SINR(S_1, t_1) < SINR_{th}$ , the arrival of  $S_2$  at  $t_1$  causes the radio to lose its synchronization with  $S_1$ . The radio starts synchronizing with  $S_2$ , because  $SINR(S_2, t_1) > SINR_{th}$ .

Existing studies confirm that three regions can be identified around a transmitter: connected, transitional, and disconnected. While the packet reception rate in the connected region is above 90%, it varies between 90% and 10% for those links in the transitional region. It has been shown that the path loss and link unreliability caused by the multipath channel can be accurately modeled using the log-normal shadowing model:

$$PL(d) = PL(d_0) + 10\eta \log_{10} \left( \frac{d}{d_0} \right) + N(0, \sigma_{ch}), \quad (2)$$

where  $PL(d)$  is the signal path loss (in dB) at distance  $d$ ,  $PL(d_0)$  is the path loss at reference distance  $d_0$ ,  $\eta$  is the path-loss exponent,  $\sigma_{ch}$  is the standard deviation of signal power variations caused by the multipath channel, and  $N(0, \sigma_{ch})$  is a zero-mean Gaussian random variable with standard deviation  $\sigma_{ch}$ . The log-normal shadowing model holds for both indoor and outdoor environments and provides a more accurate multipath fading model than the Rayleigh and Nakagami distributions [Rappaport 2002; Nikookar and Hashemi 1993].

In addition to the multipath channel variations, the studies conducted by Zamalloa and Krishnamachari [2007] and Zhou et al. [2006] showed that hardware heterogeneity is the major cause of link asymmetry. In particular, these studies show that identical devices do not exhibit similar transmission powers and noise floors. Due to the correlation between transmission power and noise floor, the proposed method by Zamalloa and

Krishnamachari [2007] is to use a multivariate Gaussian distribution to model hardware heterogeneity. Therefore, using transmission power variance ( $\sigma_{tx}^2$ ), noise floor variance ( $\sigma_{rx}^2$ ), and their correlation factor ( $\sigma_{tx,rx}$ ), we compute the adjusted transmission power ( $\Omega_i^{adj}$ ) and noise floor ( $\Psi_i^{adj}$ ) for each node.

According to the aforementioned models, we can compute the received signal power at node  $j$ , corresponding to the signal sent by node  $i$ , as

$$\Psi_j(S_i) = \Omega_i^{adj} + N(0, \sigma_{ch}) - PL(d_0) - 10\eta \log_{10} \left( \frac{d}{d_0} \right), \quad (3)$$

where  $d$  is the distance between node  $i$  and node  $j$ , and  $\Omega_i^{adj}$  is the transmission power of node  $i$ .

Let  $\mathbf{S}$  show the signal set currently being received at node  $j$ ; the SINR corresponding to the signal sent by node  $i$  is computed as

$$SINR_j(S_i) = \frac{10^{\frac{\Psi_j(S_i)}{10}}}{10^{\frac{\Psi_j^{adj}}{10}} + \sum_{S_k \in \mathbf{S} \setminus \{S_i\}} 10^{\frac{\Psi_j(S_k)}{10}}}. \quad (4)$$

Having a SINR value, we can use the equations that map SINR into bit error rate (BER). For example, for the FSK modulation used in CC1000, the bit error rate corresponding to signal  $S_i$  is [Rappaport 2002]

$$\Pr(\overline{\text{bit}})_{NCFSK} = \frac{1}{2} \exp \frac{-SINR(S_i) \times B}{R}. \quad (5)$$

Using BER, the probability of correct reception of a block of bits of size  $n$  with Manchester encoding is

$$\Pr(\text{block}) = (1 - \Pr(\overline{\text{bit}}))^{2 \times n}. \quad (6)$$

Bit error rate equations of the OQPSK modulation used in CC2420 can be found in IEEE Computer Society [2012] and Rappaport [2002]. It is worth mentioning that some simulators employ an SINR-BER or SINR-PRR map to compute packet reception probability [Dezfouli et al. 2014a]. For example, in TOSSIM [Levis et al. 2003; TOSSIM 2014], packet reception rate computation for CC2420 radio is based on an equation that fits the empirical traces of SINR-PRR.

#### 4. MODELING THE CAPTURE EFFECT

This section introduces CAMA; however, before presenting this algorithm, Sections 4.1 and 4.2 present prerequisites for the design and implementation of CAMA. These prerequisites, in particular, reduce the overhead of this algorithm and improve its implementation efficiency. Afterward, the design and operation of CAMA are described through its state machine and a sample collision scenario.

##### 4.1. Modeling Signal Arrival and Reception

In order to design and implement an accurate capture modeling algorithm, each node should be aware of the signals it is currently receiving. Information such as signal duration, signal power, and the start and end of the settling duration should be known to the capture algorithm. In the context of software development, this information can be kept in the entries of a data structure located at the physical layer module. Since each node deals with the reception and ending of many signals, the efficiency of the underlying data structure is of great importance. Assuming fixed-size packets, the insertion and deletion operations can be performed at the two ends of a data

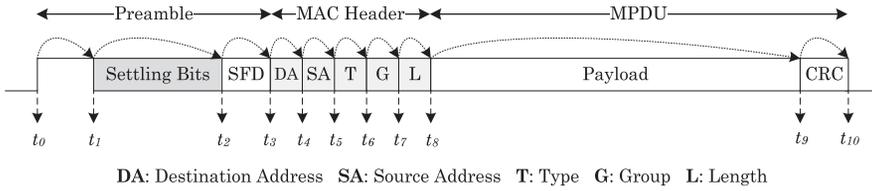


Fig. 5. Utilizing the messaging and self-messaging mechanisms for modeling radio synchronization and packet reception. These mechanisms are integrated with the state machine of CAMA to reduce the number of states and the overhead of fine-grained packet reception.

structure, because a first-arrived first-finished pattern exists. Therefore, a linked list would be a suitable data structure, since the complexity of insertion and deletion at the two ends of the list is  $O(1)$ . However, in wireless communications, packet sizes are not necessarily fixed. For example, a network-layer protocol may change the payload size based on the link quality. More importantly, low-power wireless networks usually utilize long and various preamble lengths determined by the MAC protocol. Therefore, since a deletion operation is not guaranteed to be from the end of the list, a linked list cannot be considered as an efficient data structure, because its lookup complexity is  $O(n)$ . In NS2, the *Power Monitor* module uses a linked list to record the end time and the received power of each signal. This linked list is organized in an ordered fashion based on the signals' expiry time (*viz.*, if signal  $S_i$  is before signal  $S_j$ , then  $S_i$  finishes earlier than  $S_j$ ). Therefore, the signal insertion complexity is  $O(n)$ . Although using data structures for keeping track of the incoming signals is a feasible solution, it affects the simulation performance, especially when fine-grained packet reception is required. Nevertheless, NS2 has adopted this mechanism, because it evaluates the SINR value of an incoming signal once (in NS2 CTM) or twice (in NS2 AIM), and it does not provide fine-grained packet reception. However, we will later show that this approach imposes considerable overhead to the simulation engine (cf. Sections 4.3 and 6.5). In addition, adopting this approach in CAMA imposes significant overhead in terms of data structure management and state machine complexity. For example, for each SINR evaluation, the state machine requires performing a state transition and access to the signal information data structure, which are costly operations.

As an alternative approach, we use the *messaging* mechanism provided by the underlying simulation framework to represent packet exchange between nodes.<sup>2</sup> For example, each message is capable of containing information such as packet length, signal strength, packet fields, and the duration of each field. Since each message corresponds to a packet, which in turn represents a signal, we use the terms “message,” “signal,” and “packet” interchangeably. As a variation of the messaging mechanism, *self-messaging* allows a module to schedule a message to be delivered to itself at a desired time. Using these mechanisms, CAMA avoids the overhead of maintaining a data structure through self-scheduling a message for the next time that message's information is required. This also enables CAMA to support partial packet delivery to the higher layers without requiring too many states. Figure 5 is a simple scenario to show how CAMA employs the messaging concept for radio synchronization and packet reception. A message arrives at the physical-layer module at time  $t_0$ . CAMA extracts the average received signal power to update the received signal strength (RSSI) variable. Then, it schedules this message as a self-message to be delivered at  $t_1$  for synchronization evaluation.

<sup>2</sup>From the software engineering point of view, a message is an object, which contains various fields. Therefore, for example, having a class that defines the CC1000 packet, instantiating a CC1000 packet object, and initializing its fields are equivalent to generating a new packet. Exchanging messages between nodes is in contrast to the method call mechanism used for informing nodes about a new signal reception.

Upon receiving the scheduled self-message at  $t_1$ , CAMA evaluates the SINR value and schedules this message to be self-delivered at the end of the settling duration. At  $t_2$ , CAMA checks whether the radio has been successfully synchronized. As long as the packet is synchronized with the radio, the packet is scheduled for the end of the next field to evaluate the correct reception of each field and deliver the received bytes to the MAC layer. In order to provide higher accuracy in terms of reflecting SINR variations during packet reception, scheduling a self-message for the end of a packet field can be decomposed into several self-messages during that field. For example, SINR evaluation can be performed at each byte boundary. Although we employed this technique in our simulator development, we avoid its representation for simplicity.

#### 4.2. SINR Characterization

In this section, we prove a theorem that is used in the state machine of CAMA to reduce the computational and simulation engine's event management costs.

Using Equation 1, the SINR of each signal can be computed. With respect to this model, in the following we prove that at a given receiver, at most one signal's SINR value can be higher than a given threshold level  $T$ , where  $T > 1$ .

**THEOREM 1.** *Given that the signal set  $\mathbf{S} = \{S_1, S_2, \dots, S_n\}$  is being received at a receiver, only one of the following two cases holds:*

*Case 1:*

$$\text{SINR}(S_i) < T \quad \forall S_i \in \mathbf{S} \quad \forall T \in \mathbb{R}, T > 1 \quad (7)$$

*Case 2:*

$$\exists S_i \in \mathbf{S} \quad \text{for which} \quad \text{SINR}(S_i) = T, \quad T \in \mathbb{R}, T > 1 \quad (8)$$

and

$$\text{SINR}(S_j) < \frac{1}{T} \quad \forall S_j \in \mathbf{S} \setminus \{S_i\}. \quad (9)$$

**PROOF 1.** We first present the following two lemmas.

**LEMMA 1.** *Given a set of numbers  $\mathbf{K} = \{K_1, K_2, \dots, K_n\}$ , where*

$$K_i > 0 \quad \forall K_i \in \mathbf{K} \quad (10)$$

*if for a  $K_i$*

$$\frac{K_i}{\sum_{K_t \in \mathbf{K} \setminus \{K_i\}} K_t} \geq 1, \quad (11)$$

*then*

$$\frac{K_j}{\sum_{K_t \in \mathbf{K} \setminus \{K_j\}} K_t} < 1 \quad \forall K_j \in \mathbf{K} \setminus \{K_i\}. \quad (12)$$

**LEMMA 2.** *Given a set of numbers  $\mathbf{K} = \{K_1, K_2, \dots, K_n\}$ , where*

$$K_i > 0 \quad \forall K_i \in \mathbf{K} \quad (13)$$

*if for a  $K_i$*

$$\frac{K_i}{\sum_{K_t \in \mathbf{K} \setminus \{K_i\}} K_t} = T, \quad T \in \mathbb{R}, T > 1, \quad (14)$$

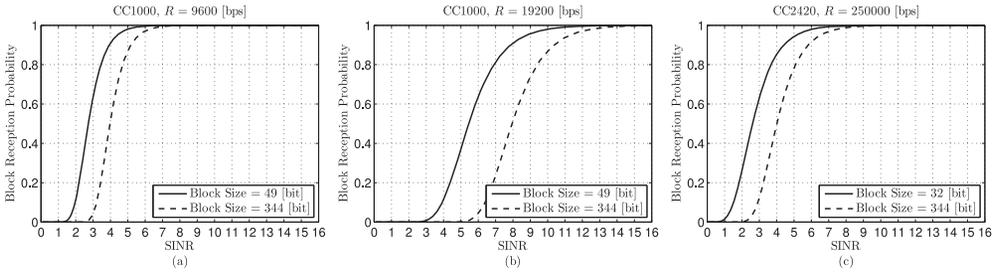


Fig. 6. Probability of radio synchronization and correct packet reception against SINR (not in dB) for CC1000 and CC2420 radios.

then

$$\frac{K_j}{\sum_{K_i \in \mathbf{K} \setminus \{K_j\}} K_i} < \frac{1}{T} \quad \forall K_j \in \mathbf{K} \setminus \{K_i\}. \quad (15)$$

Lemma 2 is a straightforward extension of Lemma 1. Considering the noise power as a signal in  $\mathbf{S}$ , both Lemma 1 and Lemma 2 can be applied to the values of  $\mathbf{S}$  (signal powers are given in watts). Hence, Theorem 1 proved.  $\square$

Assuming  $T = SINR_{th}$ , Theorem 1 states that either one of the signal's SINR value is higher than  $SINR_{th}$  or all of the SINR values are lower than  $SINR_{th}$ . Figure 6 demonstrates the probability of successful radio synchronization and correct packet reception versus SINR. For example, the required SINR values to achieve 10% synchronization probability with CC1000 9,600bps, CC1000 19,200bps, and CC2420 are about 2, 4, and 1.56, respectively. In addition, in order to achieve a 10% packet reception rate, the required SINR values for these radios are about 3.2, 6.4, and 2.9, respectively.

### 4.3. Capture Modeling Algorithm (CAMA)

This section presents the state machine of CAMA (Figure 7) and describes its operation. In order to provide simpler explanations, we assume that field reception correctness is determined by the SINR values computed at the beginning and end of each field (as demonstrated in Section 4.1). Table II presents the notations and operations used in the state diagram of CAMA.

The algorithm is composed of four states. At each state, a newly arrived signal is scheduled for the start of its settling duration, regardless of its SINR value. This feature is particularly required for low-power wireless networks, where the preamble is longer than the settling bits. We will explain this scenario in Figure 8. The description of each state is as follows:

- NO SIGNAL.** The value of the  $RSSI$  variable equals the average noise floor ( $\bar{\Psi}$ ). Therefore, the only signal being received by the radio is the noise signal.
- RECEIVING SIGNAL.** The received signal strength at the radio is higher than the average noise power. In addition, either there is no signal to satisfy condition  $SINR_{th} < SINR(S_i)$  during its settling bits or none of the packets' settling bits have arrived.
- SYNCHRONIZATION.** The settling bits of a packet have been started, and this signal meets condition  $SINR_{th} < SINR(S_i)$ .
- SYNCHRONIZED.** The radio has successfully been synchronized with a signal. The radio is receiving the SFD and MPDU of the packet.

We clarify CAMA through the scenario given in Figure 8. This scenario assumes packet  $S_1$  can be fully received, and packet  $S_2$  can be received after signal  $S_1$  finishes. Also,  $S_2$

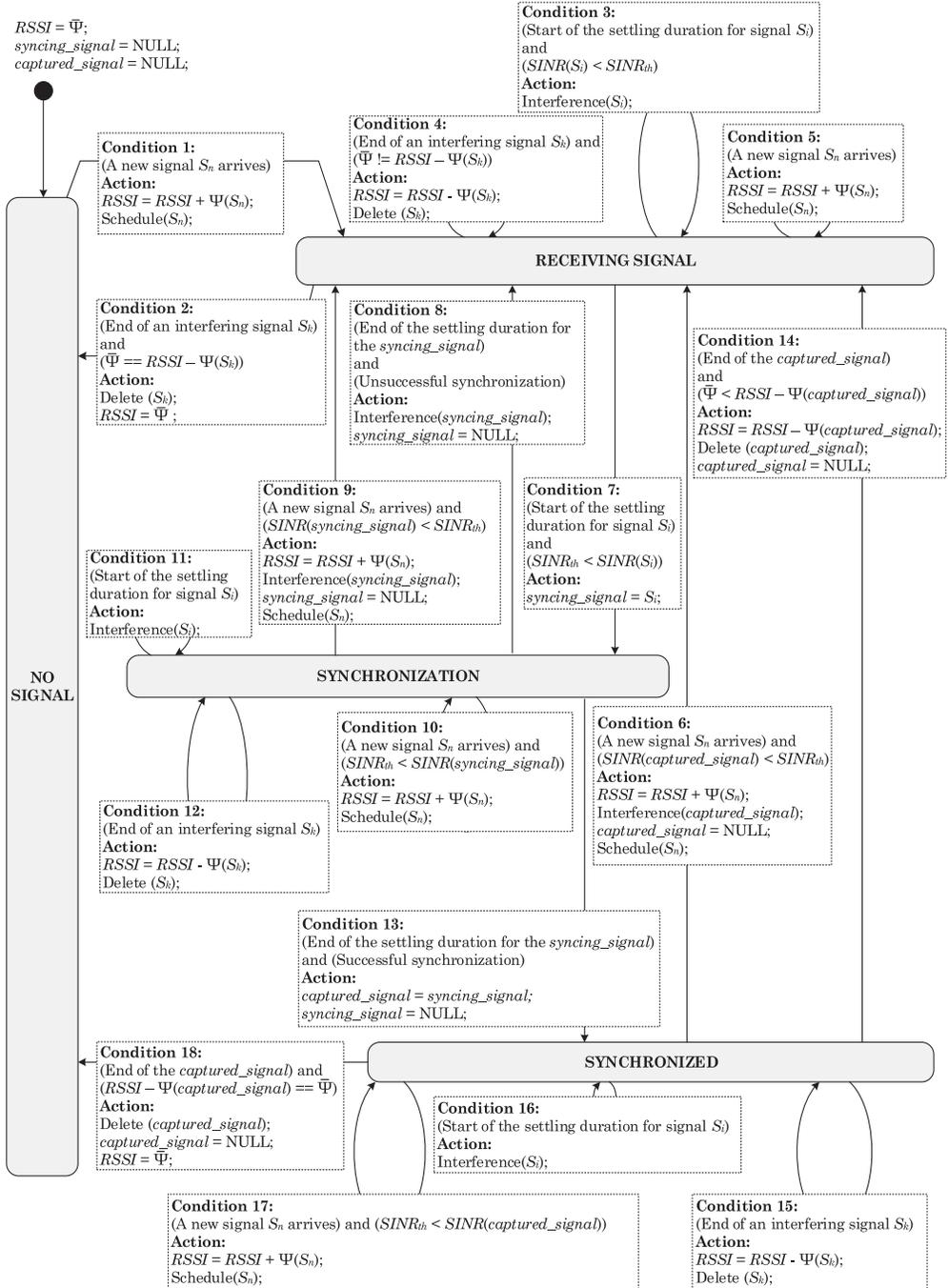


Fig. 7. The state diagram of CAMA. State transitions are demonstrated with arrows, and the conditions and actions of each transition are included in the rectangle positioned on its corresponding arrow.

Table II. Description of the Variables and Operations Used in the State Diagram of CAMA

Description	Variables	Symbol
The total received power (including noise power) at a node.		$RSSI$
The signal with which the radio is currently being synchronized.		$syncing\_signal$
The signal with which the radio has been synchronized.		$captured\_signal$
Operations		
Considers signal $S_i$ as interference for the rest of its duration. This is fulfilled by (1) scheduling signal $S_i$ for the end of its packet duration and (2) subtracting the power of signal $S_i$ from the $RSSI$ variable when the signal finishes.		Interference ( $S_i$ )
Schedules signal $S_i$ for the start of its settling duration.		Schedule ( $S_i$ )
Deletes signal $S_i$ .		Delete ( $S_i$ )

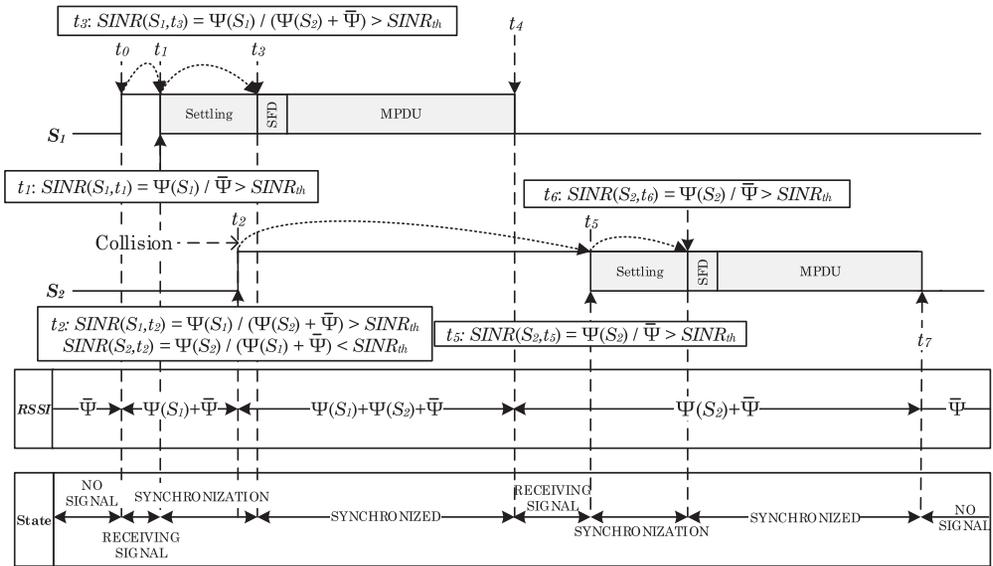


Fig. 8. Signal  $S_1$  arrives at  $t_0$ . CAMA starts synchronizing with this signal at  $t_1$ . Signal  $S_2$  arrives at  $t_2$ ; however, CAMA continues synchronizing with  $S_1$ , since  $SINR_{th} < SINR(S_1, t_2)$ . At  $t_2$ , CAMA finishes its synchronization with  $S_1$ . It receives the SFDP and MPDU of  $S_1$  from  $t_3$  to  $t_4$ . As  $SINR_{th} < SINR(S_2, t_5)$ , CAMA starts synchronizing with  $S_2$  at  $t_5$ . This figure also shows the ability of CAMA to receive the long-preamble packets that their SINR is lower than the threshold value upon arrival.

has a longer preamble length. Therefore, this scenario also shows how CAMA provides efficient capture modeling for variable-length preambles. To this aim, we show that a packet may be successfully received even if its SINR is lower than the threshold value upon its arrival. The description of Figure 8 is as follows:

- The radio is in the NO SIGNAL state until  $t_0$ . Therefore, the  $RSSI$  variable is equal to  $\bar{\Psi}$ . Signal  $S_1$  arrives at  $t_0$  (Condition 1). The capture algorithm adds the power of this signal to the  $RSSI$  variable, schedules this packet for the start of its settling duration (i.e., time  $t_1$ ), and goes into the RECEIVING SIGNAL state. Scheduling a packet for the start of its settling duration is performed through the self-message scheduling technique given in Section 4.1. This is demonstrated in Figure 7 by the  $Schedule(S_n)$  method.

- At  $t_1$ , the settling bits of  $S_1$  start and the capture algorithm computes the SINR value of this signal. If  $SINR(S_1, t_1) < SINR_{th}$  (i.e., Condition 3 holds),  $S_1$  is considered as interference for the rest of its duration, because this signal cannot provide enough power for synchronization. Therefore,  $Interference(S_1)$  is invoked and the radio stays in the RECEIVING SIGNAL state. In Figure 8, we assumed  $SINR_{th} < SINR(S_1, t_1)$ . Therefore, Condition 7 is satisfied and the radio goes into the SYNCHRONIZATION state. In addition,  $S_1$  is assigned to the *syncing\_signal* variable, because the radio is being synchronized with this signal. Furthermore, the capture algorithm inserts  $SINR(S_1, t_1)$  into the *syncing\_signal* and schedules this message for the end of its settling duration (i.e., time  $t_3$ ) to verify the success of radio synchronization.
- While the radio is in the SYNCHRONIZATION state, signal  $S_2$  arrives at  $t_2$ . Regardless of the SINR value of  $S_2$ , either Condition 9 or Condition 10 is satisfied. In both cases, the RSSI value is updated and the newly arrived signal is scheduled for the start of its settling duration. However, if  $SINR(syncing\_signal) < SINR_{th}$  (i.e., Condition 9 holds), the state changes to RECEIVING SIGNAL, and the *syncing\_signal* (i.e., signal  $S_1$ ) is treated as interference for the rest of its duration (i.e., until time  $t_4$ ). In Figure 8, we assumed  $SINR_{th} < SINR(syncing\_signal)$  (i.e., Condition 10 holds). Therefore, the radio remains in the SYNCHRONIZATION state and continues synchronization with  $S_1$ .
- At  $t_3$ , the settling bits of the *syncing\_signal* (i.e.,  $S_1$ ) finish. At this time, the capture algorithm computes the SINR value of this signal and calculates the average of the SINR values computed during the synchronization. Using Equation (5), the average value is used to compute bit error probability, which is then utilized to determine radio synchronization correctness. If the synchronization has been unsuccessful (Condition 8), the capture algorithm considers the *syncing\_signal* as interference and goes into the RECEIVING SIGNAL state. We assumed successful synchronization in Figure 8. Therefore, Condition 13 holds and the state changes to SYNCHRONIZED. During  $t_3$  to  $t_4$ , the radio is in the SYNCHRONIZED state and receives the SFD and MPDU of  $S_1$ .
- At  $t_4$ , signal  $S_1$  finishes and the capture algorithm checks whether there is any signal on the channel or not. Accordingly, based on the value of  $RSSI - \Psi(captured\_signal)$ , either Condition 18 or Condition 14 holds. Since signal  $S_2$  still exists, Condition 14 is satisfied and the radio enters the RECEIVING SIGNAL state.
- The settling bits of  $S_2$  start at  $t_5$ . Based on the value of  $SINR(S_2, t_5)$ , either Condition 3 or Condition 7 is satisfied. In Figure 8, we assumed  $SINR_{th} < SINR(S_2, t_5)$ ; hence, Condition 7 holds and the radio goes into the SYNCHRONIZATION state.
- At  $t_6$ , the settling bits of  $S_2$  finish. Assuming successful synchronization, the radio enters the SYNCHRONIZED state and starts receiving the SFD and MPDU of  $S_2$ . Therefore, although the SINR value of  $S_2$  at time  $t_2$  was lower than the threshold value, packet  $S_2$  is successfully received by the radio.
- Signal  $S_2$  finishes at  $t_7$ . Since there is no signal on the radio, Condition 18 is satisfied and the radio enters the NO SIGNAL state.

Based on the discussions presented in Section 3, successful reception of the settling bits is required for radio synchronization and packet reception. Accordingly, although CAMA schedules all the incoming signals for the start of their settling duration, no reception evaluation starts until the start of the settling bits. Therefore, CAMA's evaluation cost does not depend on the preamble duration. We will show (in Section 6.5) that through this mechanism, CAMA achieves similar simulation speed with the capture models of NS2 when the preamble size increases.

The state machine of CAMA also shows the use of Theorem 1 for performance optimization. For example, according to Condition 11, if the settling bits of a packet start when the radio is in the SYNCHRONIZATION state, the state machine immediately considers that signal as interference. Hence, though the signal has been scheduled for potential synchronization, the state machine makes a deterministic decision that avoids additional evaluations and unnecessary references to the simulation engine's event queue. Similar explanations also apply to Condition 16.

The state machine of CAMA employs techniques that provide higher accuracy and efficiency over the capture models of NS2. In order to provide further clarification, we refer to the capture model used by NS2 AIM: when a signal arrives, this packet reception algorithm checks the SINR value of the arrived signal and starts preamble reception if the SINR value is higher than  $RxThresh$ . When a new signal arrives during a preamble reception, the SINR of the new signal is evaluated and one of the following cases occurs:

- If the new signal's SINR value is higher than  $RxThresh$ , the signal currently being received should be found in the event queue and signals linked list, and it should be considered as interference for the rest of its duration. More importantly, even if this new signal can not provide enough SINR during its settling bits or MPDU bits, NS2 AIM evaluates the SINR of this signal and may start packet reception. These operations reduce the efficiency of NS2 AIM. In contrast, CAMA reduces these overheads through avoiding packet reception until the start of the settling bits, Theorem 1 and the messaging technique.
- If the newly arrived signal's SINR is lower than  $RxThresh$  at its arrival, but it can provide high SINR value during the settling duration, NS2 AIM cannot model the reception of this packet. For example, for the scenario given in Figure 8, this capture model can only receive packet  $S_1$ .

It should be noted that although CAMA utilizes a threshold SINR value ( $SINR_{th}$ ) in its transition decisions, this value is not used to decide about packet reception. Especially, CAMA determines packet error status based on the probabilistic model given in Section 3. However,  $SINR_{th}$  is required to apply Theorem 1. In order to avoid the effect of  $SINR_{th}$  on the packet reception decision, the utilized threshold value is much lower than the threshold value used in NS2 (i.e.,  $RxThresh$ ). Particularly, instead of using the required SINR value for successful packet reception (as used in NS2), CAMA employs the minimum SINR value required to achieve 10% synchronization.

Adding the collision detection capability to CAMA is straightforward. In Condition 6, if the SFD bytes have been received, the SINR of the newly arrived signal should be evaluated. If it is higher than  $SINR_{th}$ , the link models of Section 3 can be used to compute the probability of receiving at least one preamble byte.

**Discussion.** CAMA assumes that transmission power does not change significantly for the duration of a single packet, and we argue that this assumption generally holds true in the domain of wireless sensor networks. Most sensor nodes do not move considerably during a single packet time. For example, if moving 100 kilometers per hour, a sensor node with the CC2420 radio would only move about 4cm during a 46-byte packet time. Furthermore, most sensor nodes cannot and/or do not intentionally change the transmission power during a packet transmission. In particular, many network and topology control algorithms automatically adjust transmission power at the packet level [Lin et al. 2006, 2008], but to our knowledge no schemes adjust the transmission power at the symbol or byte level. Indeed, modern low-power radios such as the CC2420 do not even have the capability to adjust transmission power at

subpacket granularity.<sup>3</sup> Additionally, radios cannot easily get new information about their environment while they are transmitting and therefore have little to gain by dynamically adjusting transmission power during a single packet transmission.

#### 4.4. CAMA Correctness

**THEOREM 2.** *Assume that signal  $S_i$  is being synchronized or it is synchronized with the radio. When a new signal  $S_n$  arrives, only one of the following three cases holds:*

*Case 1:  $SINR_i < SINR_{th}$  and  $SINR_n < SINR_{th}$*

*Case 2:  $SINR_{th} < SINR_i$  and  $SINR_n < SINR_{th}$*

*Case 3:  $SINR_{th} < SINR_n$  and  $SINR_i < SINR_{th}$*

**PROOF 2.** Follows from Theorem 1.  $\square$

**THEOREM 3.** *Assume that the radio is in the SYNCHRONIZATION state. When the settling bits of a packet start, that packet should be considered as an interfering signal for the rest of its duration.*

**PROOF 3.** Suppose that the settling bits of signal  $S_k$  start when the radio is in the SYNCHRONIZATION state; that is, the radio is being synchronized with the *syncing\_signal*. Therefore, signal  $S_k$  has arrived either before or during the settling bits of the *syncing\_signal*. First, assume signal  $S_k$  has arrived before the settling bits of the *syncing\_signal*. Therefore, while the radio has been in the RECEIVING SIGNAL state, the *syncing\_signal* has satisfied Condition 7 and the radio has entered the SYNCHRONIZATION state. According to Theorem 1, this indicates that the SINR of  $S_k$  is lower than  $SINR_{th}$  as long as the *syncing\_signal* exists. Second, assume that signal  $S_k$  arrives during the settling interval of the *syncing\_signal*. Since the settling bits of this signal begin while the radio is in the SYNCHRONIZATION state, Condition 10 has been satisfied upon the arrival of signal  $S_k$ . Based on Theorem 1, the SINR of  $S_k$  is lower than  $SINR_{th}$  as long as the *syncing\_signal* exists. Hence, the theorem is proved.  $\square$

**THEOREM 4.** *Assume that the radio is in the SYNCHRONIZED state. When the settling bits of a packet start, that packet should be considered as an interfering signal for the rest of its duration.*

**PROOF 4.** Consider that the settling bits of signal  $S_k$  start when the radio is in the SYNCHRONIZED state; that is, the radio is synchronized with the *captured\_signal*. Therefore, signal  $S_k$  has arrived before, during, or after the settling bits of the *captured\_signal*. First, assume signal  $S_k$  has arrived before the settling bits of the *captured\_signal*. With this assumption, the *captured\_signal* has previously satisfied Condition 7 (and Condition 13). Therefore, according to Theorem 1, the SINR of  $S_k$  is lower than  $SINR_{th}$  as long as the *captured\_signal* exists. Second, assume that signal  $S_k$  has arrived during the settling bits of the *captured\_signal*. This implies that the *captured\_signal* has previously satisfied Condition 10 (and Condition 13) upon the arrival of  $S_k$ . Therefore, according to Theorem 1, the SINR of  $S_k$  is lower than  $SINR_{th}$  as long as the *captured\_signal* exists. Third, assume signal  $S_k$  arrives while the radio is synchronized with

<sup>3</sup>CC2420 uses an SPI interface for configuration and data exchange. In order to send a packet with a new transmission power, (1) the power control register should be modified, (2) data bytes should be sent to the radio, and (3) a send command should be issued to start the transmission. Therefore, the entire packet will be sent by the configured power.

Table III. Experimental Parameters

Parameter	Value
Radio: CC1000	
Average noise power ( $\bar{\Psi}$ ) [dBm]	-106
Noise figure [dB]	13
Noise bandwidth ( $B$ ) [kHz]	30
Modulation	NC-FSK
Encoding	Manchester
Radio speed after encoding ( $R$ ) [kbps]	19.2
Number of settling bits ( $L_{settling}$ )	49
Reference distance ( $d_0$ ) [m]	1
Path loss at reference distance ( $PL(d_0)$ ) [dB]	55
Standard deviation of transmission powers ( $\sigma_{tx}$ )	1.2
Standard deviation of noise floors ( $\sigma_{rx}$ )	0.9
Radio: CC2420	
Average noise power ( $\bar{\Psi}$ ) [dBm]	-98
Noise figure [dB]	15.3
Noise bandwidth ( $B$ ) [kHz]	194
Modulation	DSSS-OQPSK
Radio speed ( $R$ ) [kbps]	250
Number of settling bits ( $L_{settling}$ )	32
Reference distance ( $d_0$ ) [m]	1
Path loss at reference distance ( $PL(d_0)$ ) [dB]	39
Transmission channel number	26
Default Mica2 (CC1000) Packet Format	
Preamble/SFD/MAC Header/Payload/CRC	6/1/5/29/2
Default TelosB (CC2420) Packet Format	
Preamble/SFD/Length/MAC Header/Payload/FCS	4/1/1/9/29/2
Environment	
Ambient temperature [ $^{\circ}\text{C}$ ]	27
Path loss exponent ( $\eta$ ) (indoor/outdoor)	3.3/4.7
Multipath channel variations ( $\sigma_{ch}$ ) (indoor/outdoor)	5.5/3.2
White Gaussian noise ( $\sigma_{WGN}$ ) [dB]	4
Other Parameters	
$SINR_{th}$ for CC1000	4
$SINR_{th}$ for CC2420	1.56
Number of nodes in Network0	$9 \times 4$
Spacing for Network0 [m] (grid topology)	2
Number of nodes in Network1	$10 \times 10$
Spacing for Network1 [m] (grid topology)	1.5
Number of nodes in Network2	$20 \times 20$
Spacing for Network2 [m] (grid topology)	3

the *captured\_signal*. Since the radio is currently in the SYNCHRONIZED state, Condition 17 has been satisfied upon the arrival of  $S_k$ . Therefore, according to Theorem 1, the SINR of  $S_k$  is lower than  $SINR_{th}$  as long as the *captured\_signal* exists. Hence, the theorem is proved.  $\square$

## 5. GENERAL CONFIGURATION OF THE EXPERIMENTS

We developed a sophisticated simulation tool using the OMNeT++ [OMNeT++ 2014] simulation framework. The default parameters used in the experiments of this article are listed in Table III. The preamble size used for the Mica2 experiments was selected to provide the minimum settling bits. In addition, the packet format used for the TelosB experiments is in compliance with the 802.15.4 standard. For the empirical measurements with TelosB nodes, we employed channel 26 to avoid interference from

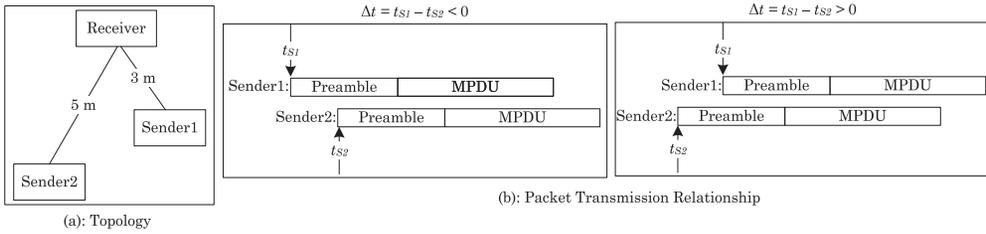


Fig. 9. The three-node experiment conducted with Mica2 nodes.

nearby 802.11 networks. We present the results with median, lower quartile, and higher quartile.

## 6. VALIDATION AND COMPARISON

The main aim of this section is to validate the accuracy of CAMA through comparison with the empirical results of three different experiments. First, we use Mica2 (CC1000) nodes and perform the three-node experiment to carefully confirm the accuracy of CAMA through gradual changes in packet arrival times. Second, we validate CAMA through comparison with the empirical results of the four-node experiment conducted with TelosB (CC2420) nodes. Through this experiment, we carefully validate the accuracy of CAMA against the variations in the capture behavior caused by changes in packet reception times and received signal powers. Finally, we employ a broadcast traffic pattern in a network of 36 Mica2 nodes to prove the network-level credibility of CAMA. Using these experiments, we also reveal the low accuracy of the capture models employed by NS2. Our results show that while CAMA can accurately predict packet reception performance in the presence of interference, the capture models of NS2 demonstrate considerable inaccuracies in terms of packet reception, collision detection, and partial packet recovery. In addition to these evaluations, we also compare the simulation speed achieved by CAMA and the capture models of NS2. Our analyses show that CAMA is more efficient in signal reception and interference handling.

The following definitions are used in the evaluations presented in this section:

- Reception:** When an entire packet has been successfully received.
- Collision detection:** A collision is detected when the following conditions hold: (1) the radio is synchronized with a packet and (2) a new packet arrives and at least one of its preamble bytes is received.
- Partial packet recovery:** A packet is partially recovered when the following conditions hold: (1) the radio is synchronized with the mentioned packet, (2) a collision happens and the radio loses its synchronization, and (3) at least one field of the MPDU of the mentioned packet is received.

### 6.1. The Three-Node Experiment

In this section, we present the results of our three-node experiment, which is aimed for (1) detailed analysis of the capture effect with respect to packet reception timing and (2) confirmation of the accuracy of CAMA with the empirical traces of CC1000 radio. Two nodes, *Sender1* and *Sender2*, send their packets to a receiver with predetermined timing differences. The packet transmission times for *Sender1* and *Sender2* are denoted by  $t_{S1}$  and  $t_{S2}$ , respectively. The timing difference between the transmissions is  $\Delta t = t_{S1} - t_{S2}$ . For every  $\Delta t$  value, each transmitter sends 100 packets as fast as it can (without employing CCA). In order to receive higher signal power from one of the transmitters, *Sender1* is moved so that it is closer to the receiver than *Sender2*. Figure 9 shows the topology and transmission timing relationships of the three-node

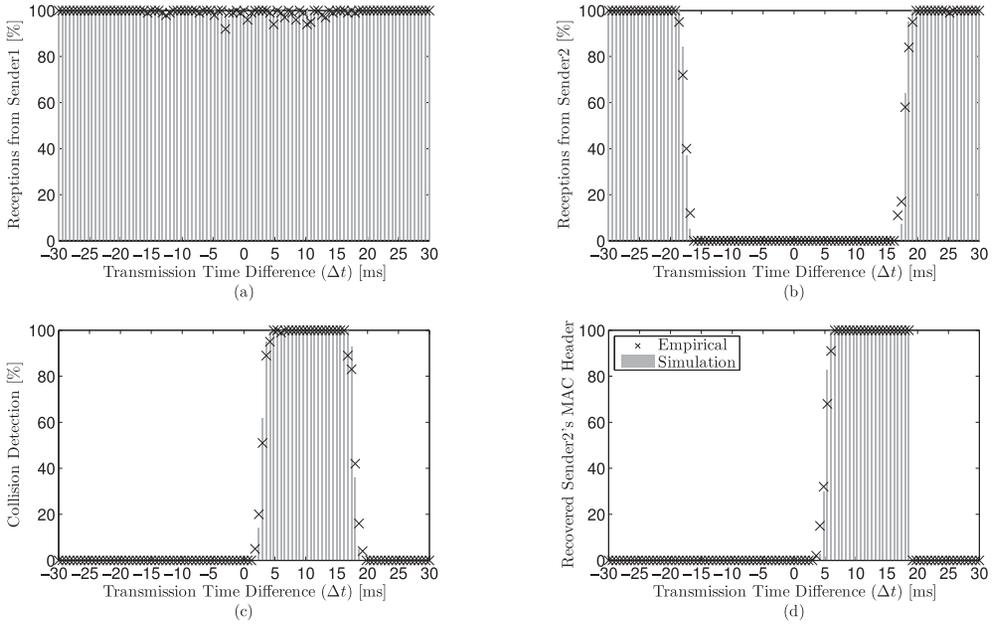


Fig. 10. The results of the empirical and simulated three-node experiment with 6-byte preamble. These results confirm the accuracy of CAMA in reproducing the empirical traces obtained by applying gradual changes in packet arrival times.

experiment. Both the simulation and empirical results consider a 1ms time synchronization accuracy.  $\Delta t$  is varied from  $-30$  to  $30$ ms in  $0.6$ ms intervals. We also employed a synchronizer node to trigger *Sender1* and *Sender2* for packet transmission in the empirical experiments.

We have reported both the empirical and simulation results of this experiment in Figure 10. As it can be observed, CAMA can accurately generate the empirical traces with respect to changes in the transmission times.

*Sender1*'s packets are almost always received due to the higher signal power received from this sender. Since the packet duration is about  $17.92$ ms, *Sender2*'s packets can be successfully received when they are not overpowered by *Sender1*'s transmission, that is,  $|\Delta t| > 17.92$ . In terms of collision detection, two conditions are required: first, the weaker signal should arrive earlier; second, the stronger signal should arrive after the successful synchronization of the weaker signal. Therefore, collision detection requires *Sender1*'s packet to arrive at least  $2.92$ ms later than *Sender2*'s packet. In this situation, after receiving the SFD byte of *Sender2*'s packet, *Sender1*'s packet arrives and synchronizes with the radio. Therefore, the SFD byte of the new signal arrives while the radio was assumed to be synchronized with *Sender1*. This allows the software to detect a collision. Notice that a collision cannot be detected if the preamble bytes of *Sender1* arrive before the SFD bytes of *Sender2*. This is because the MAC layer cannot discriminate between the preamble bytes belonging to different packets. In order to recover *Sender2*'s MAC Header, *Sender1*'s signal should arrive  $5$ ms after *Sender2*'s signal.

## 6.2. The Three-Node Experiment: Long Preamble and Comparison with NS2

In this section, we repeat the three-node experiment to confirm the accuracy of CAMA when a long preamble is used by the MAC layer. Furthermore, we show that none

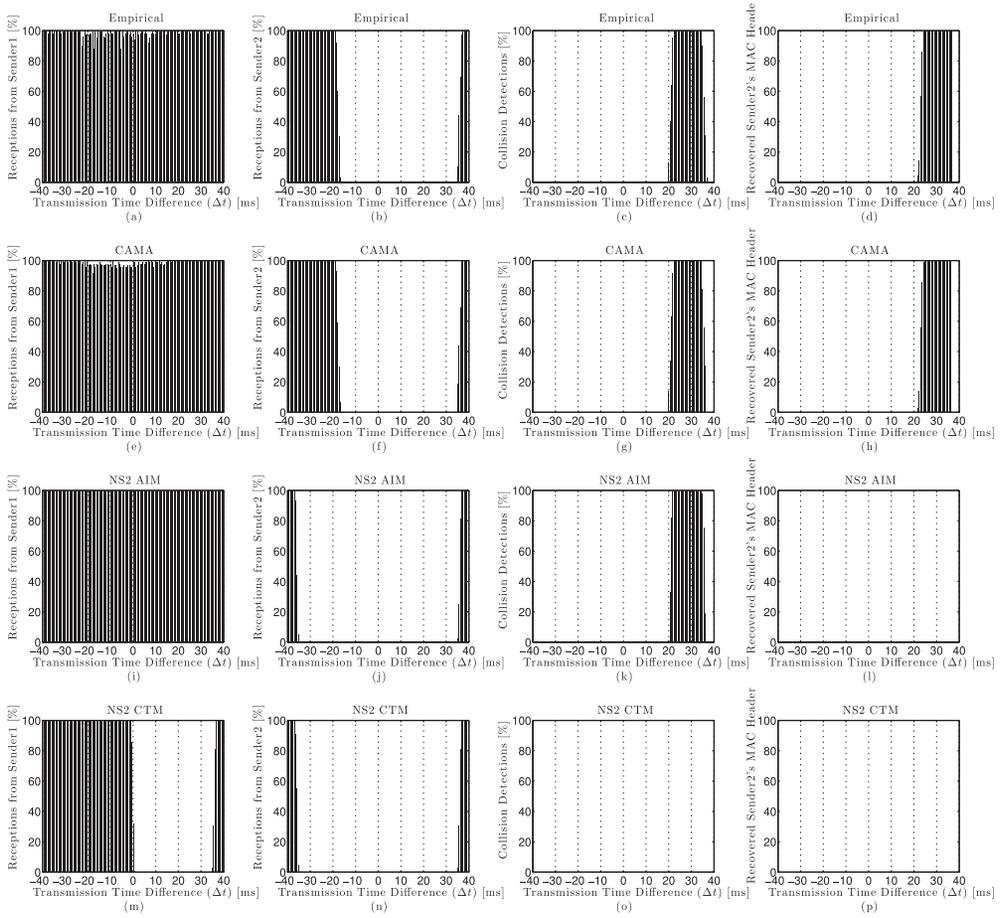


Fig. 11. The results of the empirical and simulated three-node experiment with 49-byte preamble. These results specifically reveal the inaccuracy of the capture models of NS2 when used for evaluating low-duty-cycle networks with long packet preamble.

of the NS2 capture models is suitable for the simulation of low-power networks with long packet preamble. In order to provide a fair comparison, we have improved the capture model of NS2 AIM to adaptively compute  $RxThresh$  based on packet size and other network parameters. In addition, since NS2 AIM supports stronger-last capture, we have enabled this model to indicate collision detection whenever a stronger-last capture occurs after the preamble duration of the first packet.

The empirical and simulation results of this experiment are given in Figure 11. While CAMA agrees with the empirical results, both NS2 models show considerable inaccuracies in terms of packet reception, collision detection, and partial packet recovery.

Similar to CAMA, NS2 AIM can also model the packet receptions from *Sender1*. However, it cannot model packet reception variations, because it employs a threshold value instead of using the probabilistic packet reception model. Using NS2 CTM, *Sender1*'s packets cannot be received if *Sender2*'s packets arrive earlier and overlap with *Sender1*'s packets. In this case, upon the arrival of *Sender1*'s signal, the capture threshold model computes the ratio of the received signal strength from *Sender2* to the received signal strength from *Sender1*. As this value is lower than  $RxThresh$ , both packets are discarded.

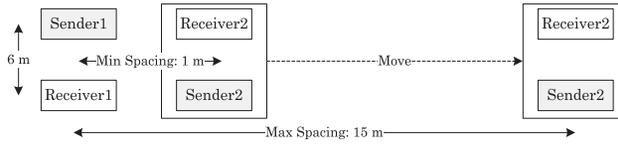


Fig. 12. The topology of the four-node experiment conducted with TelosB nodes.

With respect to the packet receptions from *Sender1*, since the packet duration is about 35.83ms, *Sender2*'s packets can be received when  $|\Delta t| > 35.83$ . However, since the employed preamble is larger than the settling bits, *Sender2*'s packets can also be received when *Sender1*'s packet overlaps with about the first 350 bits of *Sender2*'s preamble (i.e.,  $-35.83 < \Delta t < -17.6$ ). In this situation, after receiving *Sender1*'s preamble, the radio can be synchronized with *Sender2*'s packet, because the remaining number of preamble bits is long enough for synchronization. However, as Figures 11(j) and 11(n) show, both NS2 models fail to capture a packet when a portion of its preamble is overpowered. This behavior specifically highlights the very low accuracy of NS2 for performance evaluation of low-power wireless networks.

In terms of collision detection, a collision can be detected if the preamble bytes of *Sender1* arrive during a packet reception from *Sender2* (i.e.,  $20.83 < \Delta t < 35.83$ ). Figure 11(k) shows that NS2 AIM can achieve the same results as CAMA through the improvements we added to NS2 AIM to support collision detection. However, we will later show that NS2 AIM can perform collision detection only in interference-controlled experiments, where the preamble of the second packet can be completely received (cf. Section 6.4). Figure 11(o) shows no collision detection, since NS2 CTM does not support stronger-last capture.

MAC Header recovery happens when  $22.92 < \Delta t < 35.83$ . As Figure 11(l) shows, NS2 AIM does not provide MAC Header recovery, because when a packet from *Sender1* arrives during a packet reception from *Sender2*, *Sender2*'s packet is discarded due to its low SINR value. A similar inefficiency is observable with NS2 CTM. In contrast, since CAMA handles packet bytes as independent data entities, it can deliver the received data bytes to the MAC layer upon reception.

### 6.3. The Four-Node Experiment

In this section, we conduct a fairly complex experiment with four nodes. This experiment has two main differences with the three-node experiment: First, while the only variable in the three-node experiment was time, both transmission times and node positions are varied here. Therefore, in addition to the changes in signal arrival times, we also vary the received signal powers. Second, instead of using Mica2 (CC1000) nodes, we use TelosB nodes, which employ the CC2420 radio. Therefore, we aim to validate CAMA against a newer radio that utilizes a more complex modulation.

We consider two senders and two receivers in the topology given in Figure 12. We also use a synchronizer node to trigger the senders. We introduce  $\Delta t$ , which indicates the exact timing difference between the transmissions of the senders. The two senders transmit concurrently when  $\Delta t = 0$ , *Sender1* transmits earlier when  $\Delta t = -0.5$ ms, and *Sender2* transmits earlier when  $\Delta t = 0.5$ ms. Since we disabled CCA to avoid random delays, we observed that the senders send almost concurrently when  $\Delta t = 0$ . Therefore, we did not observe any considerable number of collision detections. Accordingly, we introduce the second timing parameter, denoted by  $\delta_t$ , which indicates the random delay chosen by the senders before transmission. In order to observe higher variations in packet reception and collision detection, we set  $\delta_t = 1$ ms so that it is shorter than the packet transmission duration (i.e., 1.47ms). For every  $\Delta t$  value, each sender transmits

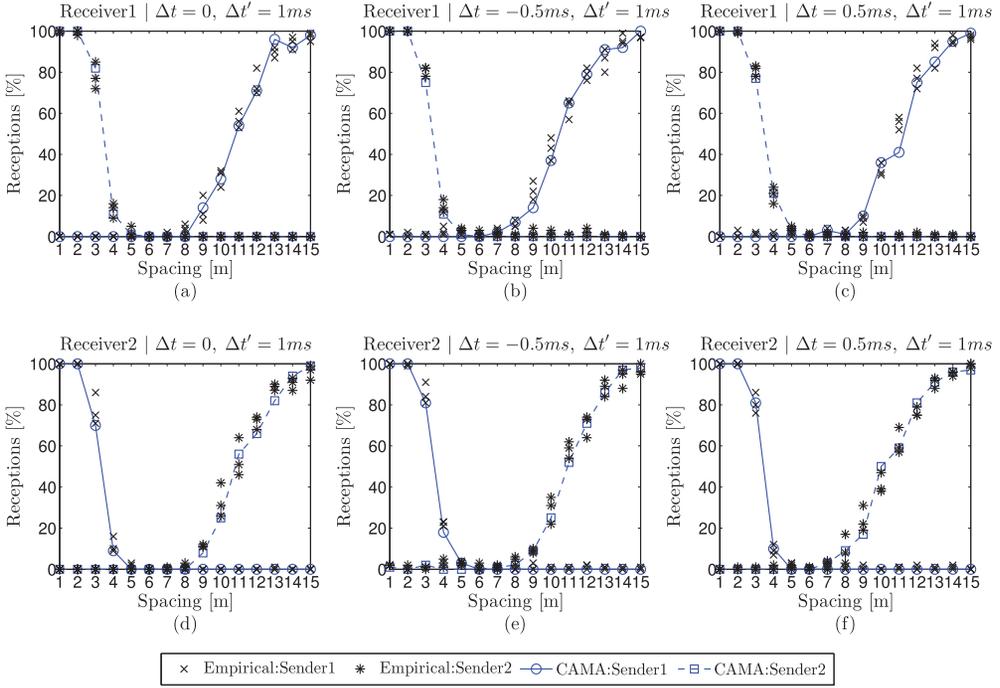


Fig. 13. Comparing the number of packet receptions obtained from the empirical measurements and CAMA. Although the capture effect in the four-node experiment is affected by both transmission time and signal power, CAMA accurately predicts the empirical traces.

100 packets, and each packet’s random delay is uniformly selected from  $[0, \delta_t]$ . We set the transmission power to 0dBm to ensure that all the nodes are in the interference range of each other. Therefore, even when the spacing is 15 meters, each transmitter affects other receivers, because the transmission range of the nodes is up to 37 meters. Due to the variations we observed in the empirical measurements, the experiment was repeated three times at each distance. In particular, after each new node spacing, we slightly changed the direction of *Receiver2* and *Sender2* and repeated the experiment. Therefore, we present three empirical values per spacing.

Figures 13 and 14 compare the empirical results against CAMA and the capture models of NS2, respectively. Each subfigure presents the number of packets received by a specific receiver from the two senders. These results show that each receiver can achieve up to a 100% packet reception rate in the presence of interference. Additionally, when the spacing is between 3 and 14 meters, we observe a transitional region in which the receivers can achieve between 10% and 90% packet capture from at least one of the senders. Although changes in transmission times and reception powers cause considerable variations in packet capture, our results show that CAMA follows the empirical traces.

Neither NS2 CTM nor NC2 AIM can model the empirical traces. For example, when the spacing is increased from 10 to 11 meters, NS2 AIM shows a sharp jump, and no transitional region can be observed. As another example, although the empirical results indicate that packet capture is possible when the spacing is about 9 to 10 meters, NS2 AIM shows no sign of packet reception. NS2 CTM’s results are even worse. For example, in Figure 14(a), when the spacing is higher than 11 meters and the SINR value received from *Sender1* is larger than the packet reception threshold, NS2 CTM shows about 50%

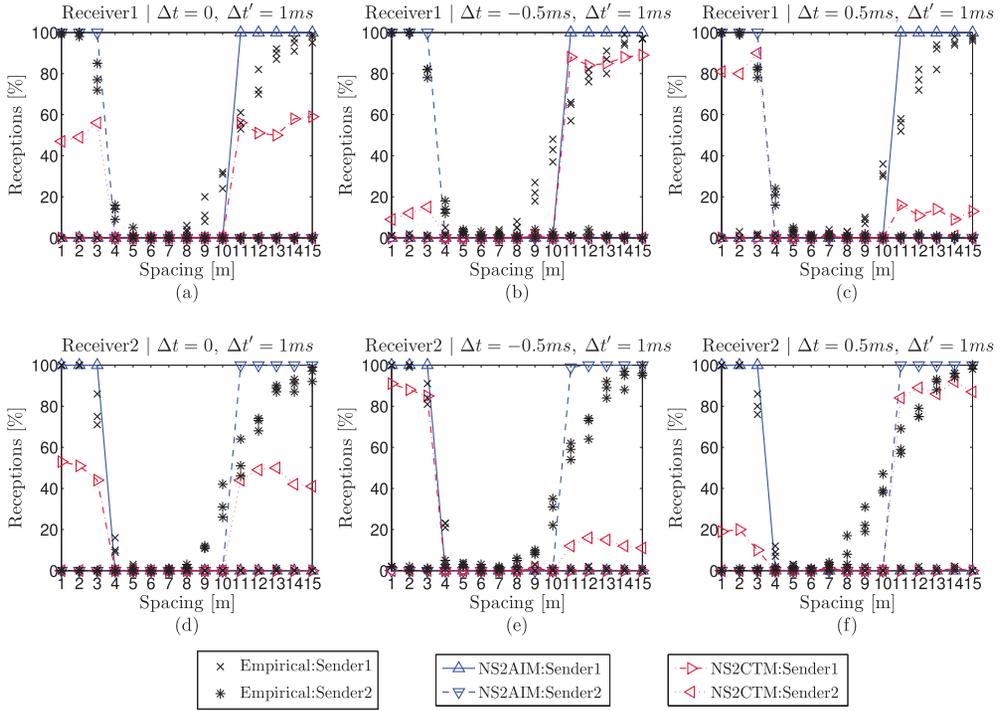


Fig. 14. Comparing the number of packet receptions obtained from the empirical measurements and the capture models of NS2. Both NS2 AIM and NS2 CTM show considerable error compared with the empirical results.

packet reception. With respect to the lack of stronger-last capture support and the 1ms random transmission delay chosen by the transmitters, NS2 CTM can only capture those collisions in which the stronger packet arrives earlier. Therefore, changing the random delay causes higher receptions from the stronger sender.

Figure 15 shows the number of collision detections. We did not show the number of collisions with NS2 AIM, because it was very inaccurate. Our results show that CAMA can regenerate the empirical trace of collision detection with high accuracy. Therefore, through the three-node and four-node experiments, we showed that CAMA can represent the collision detection efficiency achievable by the CC1000 and CC2420 radios.

When  $\Delta t = 0$ , the number of collision detections is always lower than 50. This is because these detections are only achieved due to the 1ms random transmission delay ( $\delta_t$ ) we introduced in the experiment. Specifically, a collision detection only happens when the randomly selected interval between packet transmissions is more than  $160\mu s$ . In this case, the SFD pin of CC2420 goes high after synchronizing with the first packet. However, when the second packet arrives, the SFD pin goes low (due to the loss of synchronization) and goes high again after successful synchronization with the second packet. An interesting observation is the very low collision detection rate when the spacing is around 6 meters. In this case, since the distance between the transmitter-receiver pairs is almost equal, the SINR of the secondly arrived signal is not high enough for radio resynchronization.

When  $\Delta t = -0.5ms$  and the spacing is below 6 meters, *Receiver1* can effectively detect the collisions, because the weaker signal (from *Sender1*) arrives earlier and

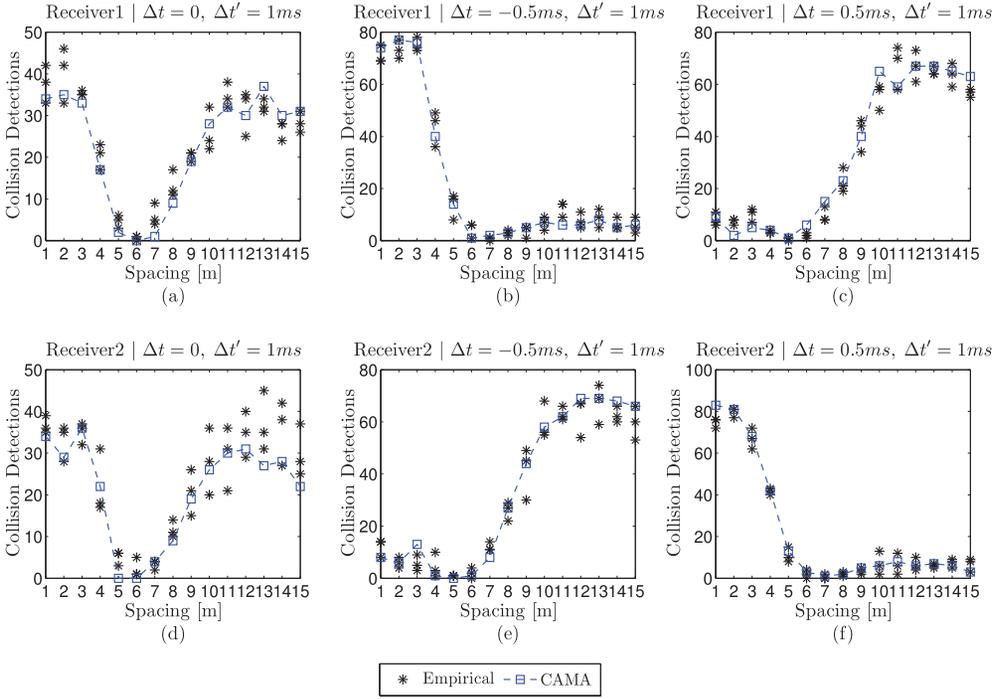


Fig. 15. Comparing the number of collision detections obtained from the empirical measurements and CAMA. These results validate the accuracy of CAMA to generate the collision detection behavior of the CC2420 radio.

the arrival of the stronger signal allows the receiver to be resynchronized. When the spacing is above 6 meters, although  $\Delta t = -0.5ms$  means that the stronger sender transmits earlier, the random delay caused by  $\delta_t$  brings about cases in which *Sender2*'s transmission arrives adequately earlier than that of *Sender1*. This allows *Receiver1* to achieve about 10% collision detection efficiency. Similar analysis can be presented for other scenarios.

#### 6.4. The 36-Node Experiment

In this section, we consider a 36-node network (Network0 in Table III) to validate the network-level accuracy of CAMA. These analyses also allow us to further highlight the considerable inaccuracy of the NS2 capture models. We chose Mica2 nodes for this experiment, because the hardware and signal propagation characteristics of this node are well known and can be modeled in our simulation tool. Therefore, it is easier to eliminate those unwanted effects that may cause a gap between the results of empirical and simulated experiments. Additionally, to avoid the influence of network protocols on performance, we considered a simple scenario in which every node broadcasts a fixed number of packets.<sup>4</sup> As the capture effect efficiency depends on interference intensity, we consider scenarios both with and without MAC to control packet transmissions. The MAC protocol employed is the default TinyOS's CSMA protocol. In addition, to show how employing long preambles reflects in the network performance, we considered 6-byte and 719-byte preambles. Table IV shows the evaluation scenarios. Due to the

<sup>4</sup>This traffic type is similar to that used in the initial phases of sensor networks for performing neighbor discovery and link estimation [Dezfouli et al. 2014b; Radi et al. 2013, 2014].

Table IV. Evaluation Scenarios of the 36-Node Experiment

Scenario0	Scenario1	Scenario2	Scenario3
6-Byte Preamble	6-Byte Preamble	719-Byte Preamble	719-Byte Preamble
CSMA	1 Packet/ 50ms	CSMA	1 Packet/ 1,000ms
$(CW_i, CW_c) = (12, 6)$ ms		$(CW_i, CW_c) = (12, 6)$ ms	

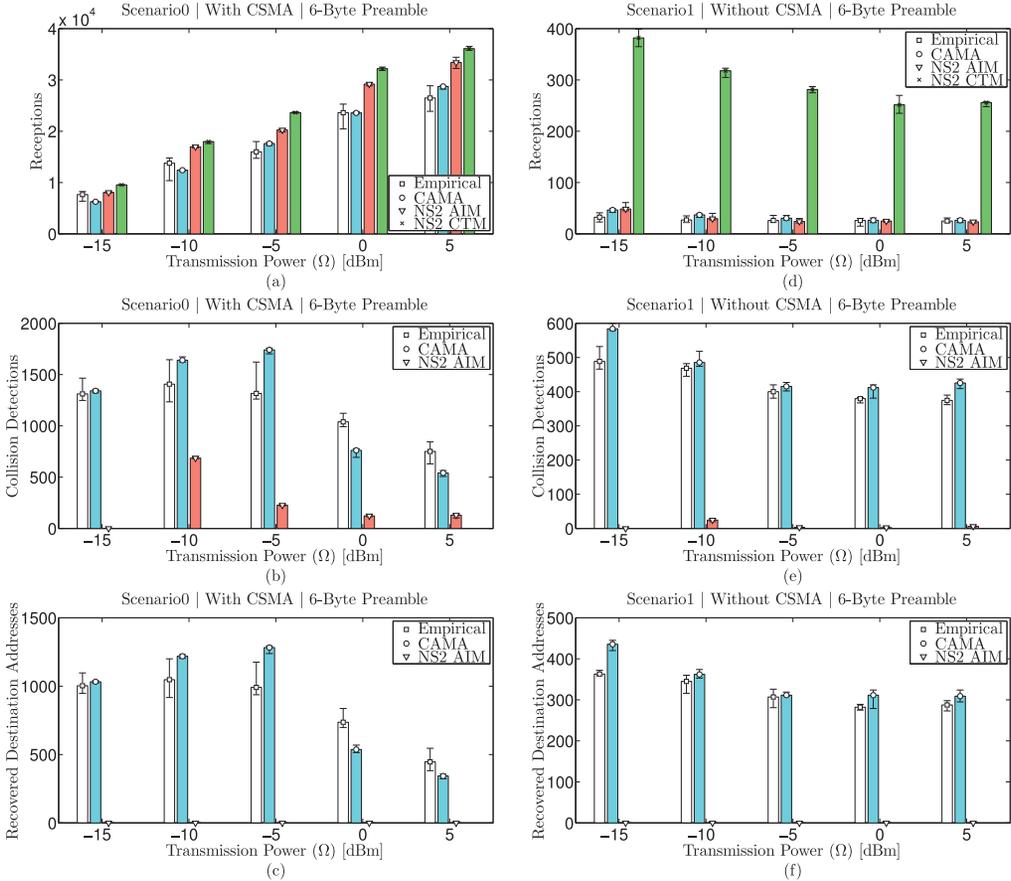


Fig. 16. Empirical and simulation-based evaluation of packet reception, collision detection, and MAC Header recovery using 6-byte preamble in the 36-node experiment.

considerable variations in our empirical measurements, the experiment is repeated 10 times for each scenario.

Figures 16 and 17 show the comparative results. These results confirm the high accuracy of CAMA in all scenarios. While the results of NS2 AIM and NS2 CTM are only reasonable for particular scenarios, CAMA follows the empirical traces in all scenarios. It should be noted that although our simulation results with CAMA do not accurately match with the empirical results, these mismatches are not necessarily due to the inaccuracy of CAMA. In particular, while we have modeled most of the characteristics of low-power wireless communications (such as path loss, multipath effect, and hardware heterogeneity), some characteristics such as radio irregularity are almost impossible to be replicated from a real-world deployment [Dezfouli et al. 2014a]. Therefore, in contrast to the small-scale experiments in which transmission

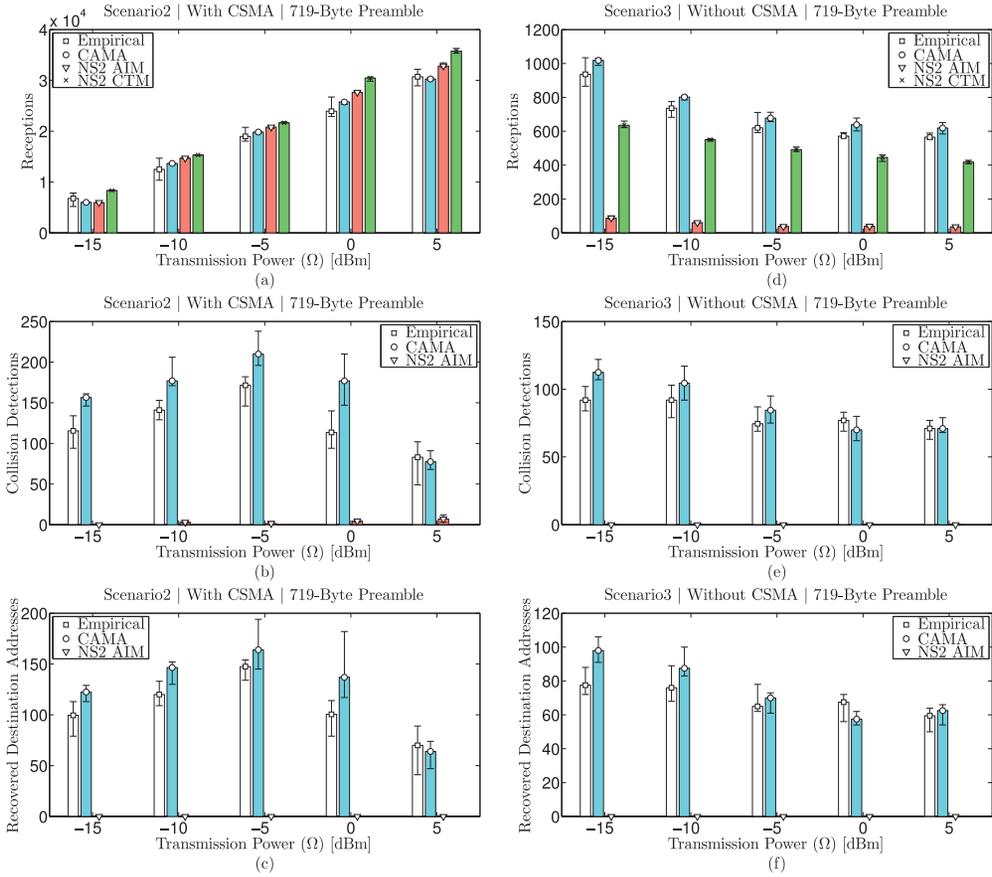


Fig. 17. Empirical and simulation-based evaluation of packet reception, collision detection, and MAC Header recovery using 719-byte preamble in the 36-node experiment.

times and interference are controllable, the random characteristics of larger networks avoid a perfect match between the simulation and empirical traces.

Although NS2 CTM does not support stronger-last packet capture, we can observe that the number of receptions achieved with this model is considerably higher than that of NS2 AIM in all scenarios. This is due to the capture threshold model employed, which underestimates the interference due to neglecting the additive interference. Accordingly, we can observe the higher inaccuracy of NS2 CTM in Figures 16(d) and 17(d) in which the CSMA MAC is not used and packet receptions are more prone to interference. With respect to these results, while not considering that the additive interference causes a higher number of receptions when the preamble size is small, neglecting stronger-last captures reduces the number of packet receptions when the preamble size enlarges.

Our results reveal the inaccuracy of NS2 AIM, especially when the preamble size is large. While NS2 AIM considers additive interference and provides a higher accuracy compared with NS2 CTM, its main deficiency appears in long-preamble scenarios. In contrast to CAMA, which enables packet synchronization with partial preamble reception, NS2 AIM requires all preamble bytes to be received for a successful packet reception. In terms of collision detection, NS2 AIM does not follow the empirical traces for the following reasons: First, its inaccurate packet synchronization model also affects

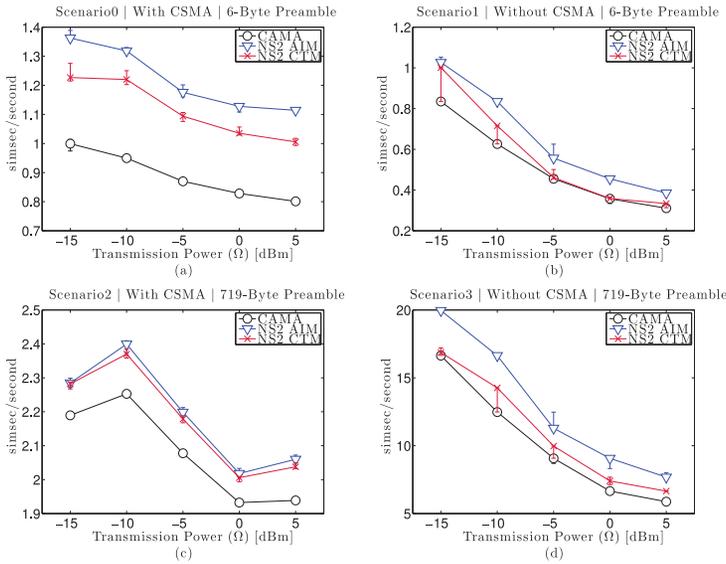


Fig. 18. The simulation speeds of the scenarios of the 36-node experiment. Since CAMA efficiently models the effects of interference on packet receptions, interference intensification reduces the speed gap between CAMA and the capture models of NS2.

collision detection. Second, due to the use of  $RxThresh$  for identifying correct packet reception, a successful collision detection requires the secondly arrived packet's SINR value to be higher than  $RxThresh$ . In contrast, since CAMA provides fine-grained packet reception, it can detect a collision after receiving a preamble byte. Based on these results, although NS2 AIM provides collision detection in interference-controlled environments (such as the three-node experiment shown in Section 6.2), it cannot provide efficient collision detection in realistic high-interference environments.

The observable rise and fall in the number of collision detections is due to the changes in the number and types of collisions. While low transmission power reduces the chance of packet collision, high transmission power is associated with higher coverage and lower collisions. Therefore, usually the highest number of collisions is achievable with medium transmission powers. Further analysis of collision detection is conducted in Section 7.

### 6.5. Simulation Speed: CAMA Versus NS2

In this section, we compare the simulation performance of CAMA versus the capture models of NS2. This study confirms the higher efficiency of interference handling in CAMA. Specifically, since increasing the network size or interference intensity does not compromise the accuracy and speed of CAMA, it provides a scalable solution for modeling the capture effect. Although the simulation speed depends on many parameters, we focus on the simulation speed from the physical-layer point of view. Therefore, in order to present a fair comparison, we provided three identical simulation tools, where the only difference was the physical-layer implementation. Note that higher *simsec/second* indicates faster simulation.

Figure 18 shows the simulation speed corresponding to the scenarios of the 36-node experiment. By comparing Scenario0 and Scenario1, we observe that interference intensification reduces the speed reduction of CAMA. For example, while the speed reduction of CAMA against NS2 AIM is up to 28% in Scenario0, the performance reduction is as low as 13% in Scenario1. Increasing interference causes higher signal

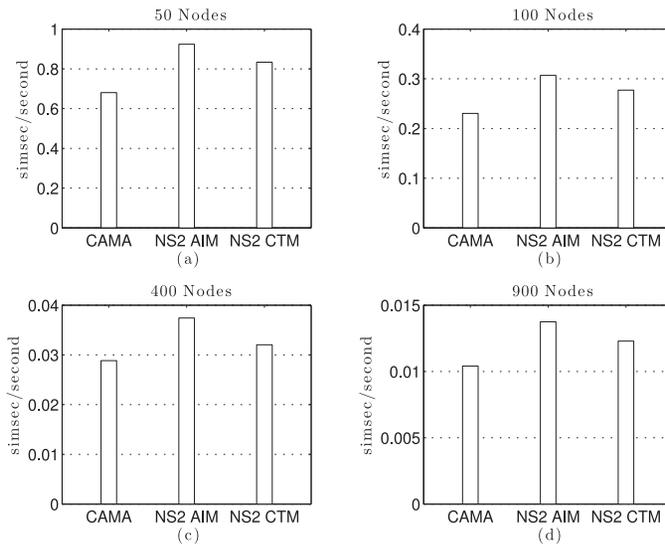


Fig. 19. The simulation speed of CAMA against the capture models of NS2 for four network sizes. These results show that the higher accuracy of CAMA does not cause extra overhead when the network size enlarges.

arrivals per node. With respect to NS2, this causes a reduction in the simulation speed for two reasons: First, it increases the number of SINR computations upon signal arrivals. Second, since a signal arrival during a packet reception may corrupt an ongoing packet reception, NS2 requires finding the signal being received in the event list and considers that signal as interference. Accordingly, interference intensification results in a considerably higher computational and memory management cost for NS2. In contrast, CAMA reduces the aforementioned costs as follows. First, regardless of the number of signal arrivals at a node, CAMA does not start packet reception unless the settling bits have arrived. Therefore, it does not require performing SINR computation or managing interference before packet reception. Second, CAMA reduces the number of SINR computations and lowers the simulation engine's event management overhead through utilizing Theorem 1. In order to further confirm these observations, we analyzed how increasing the number of nodes affects the simulation speed. Since we assumed that each node's transmitted signal is delivered to other nodes (regardless of the reception power), increasing the number of nodes enhances the number of signal arrivals per node. As our results in Figure 19 show, the reduction in the simulation speed of CAMA against NS2 AIM and NS2 CTM is always less than 25% and 15%, respectively, regardless of network size. Therefore, the higher accuracy of CAMA does not restrict its simulation scalability.

Comparing the scenarios with short and long preambles, it is obvious that a larger preamble size reduces the speed gap between CAMA and NS2. Since both NS2 models start packet reception immediately after arrival, their number of SINR computations and event management tasks increases as the packet size enlarges. However, based on our earlier discussions, the packet reception cost of CAMA is completely independent of preamble size. To confirm this behavior, Figure 20 shows the simulation speeds achieved with different preamble sizes. While the speed reduction of CAMA against NS2 AIM is about 25% with 6-byte preambles, it reduces to 8% with 719-byte preambles.

Since providing partially received packet bytes for the higher layers reduces the speed of CAMA, its performance can be improved through disabling this feature when

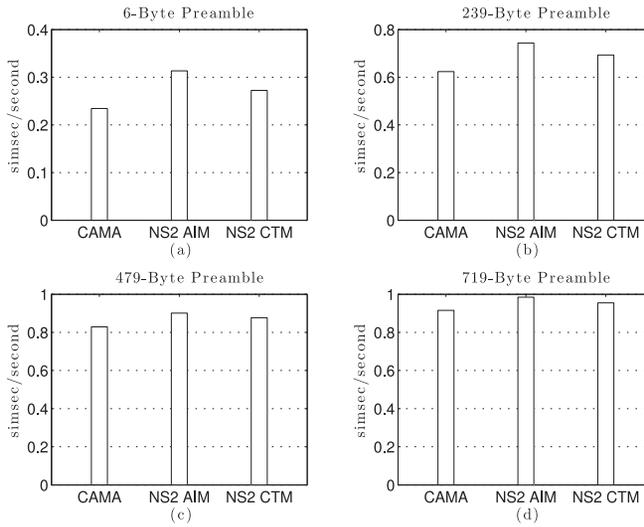


Fig. 20. The simulation speed of CAMA against the capture models of NS2 for four preamble sizes. As the packet reception cost of CAMA is independent of preamble size, these results confirm that preamble enlargement results in lower reduction in the speed of CAMA compared with NS2 models.

the higher layers are only interested in the completely received packets. On the other hand, the speed of CAMA can further be improved through lowering the overhead of the simulation engine's event management. For example, signals with a received power lower than the minimum required power for radio synchronization can immediately be considered as interference. In this case, the maximum interference-free SINR of the incoming signal should be computed and compared with a carefully determined threshold value.

**Discussion.** Computational time is essential for physical-level network simulation and is the main reason most network simulators today do not model the physical layer during simulation and instead use link-level modeling. Conventional physical-level models take thousands of times longer than link-level models because every packet has thousands of chips, each of which is modeled independently. Thus, a simulation that takes 10 minutes with a link-level model would take 7 days with a physical-level model, assuming each packet has only 1,000 chips. Although a large set of simulations can easily be parallelized on a cluster, a single simulation is difficult to parallelize efficiently due to the requirement for synchronizing the event queues and the overhead of remote method calls. In contrast, CAMA provides a technique to model physical-layer effects with no more than two times the simulation time of using a link-level model.

## 7. INVESTIGATING THE CAPTURE EFFECT IN WIRELESS SENSOR NETWORKS

In this section, we study the capture effect in large wireless sensor networks. Our investigations show that regardless of the carrier-sensing threshold and transmission power, a significant number of packet receptions is affected by high-power interference. We also perform sensitivity analysis of the capture effect to reveal the effects of design approaches and network parameters on the efficiency of collision detection and reception recovery.

We use the flooding traffic pattern due to its numerous applications [Lu and Whitehouse 2009; Radi et al. 2011; Maróti et al. 2004]. Each node retransmits the received flood only once. Therefore, a flood finishes when no packet is received at a node that has not sent the flood. Medium access is controlled by the TinyOS's CSMA

MAC protocol with  $(CW_i, CW_c) = (120 \text{ ms}, 12 \text{ ms})$ . We use the two network topologies denoted as Network1 and Network2 in Table III. Each result point is the median of 10 simulation runs. In our analyses, we refer to a specific point in a figure as  $w \setminus x \setminus y \setminus z$ , where  $w$  is the network name,  $x$  is the transmission power level,  $y$  is the carrier-sensing level, and  $z$  determines the environment type. Followings are the definitions and metrics used for the analyses of this section:

- High-quality link:** The synchronization probability of a signal transmitted over a high-quality link is higher than 75%.
- High-quality/power signal:** A signal transmitted over a high-quality link.
- High-power collision:** Having at least one high-quality signal at a receiver, the arrival of each high-quality signal is considered as a high-power collision. Notice that a high-power collision does not necessarily happen during a packet reception. Also notice that when a packet is being received, the arrival of a high-quality signal results in a high-power collision, but it does not necessarily cause packet corruption.
- Reception with high-power collision:** When at least one high-power collision happens during a reception. Therefore, a reception with high-power collision experiences one or both of the capture types (i.e., stronger-first and stronger-last).
- Detection efficiency:** The number of detected collisions divided by the total number of high-power collisions. Notice that detection efficiency depends on the capture timing for successful collision detection.
- Reception efficiency:** The number of receptions in collision divided by the total number of high-power collisions. Notice that reception efficiency is independent of the capture timing.

### 7.1. Packet Reception Analysis

The number of receptions and the number of receptions with high-power collision at each run are demonstrated in Figure 21. In order to characterize the influence of carrier sensing and environmental parameters, the results are given for three different carrier-sensing thresholds in the indoor and outdoor environments. These results signify the importance of modeling the capture effect due to the high number of receptions with collision. For example, with Network2\0\ -100\Indoor and Network2\0\ -100\Outdoor, more than 60% and 30% of the receptions are with high-power collision, respectively.

At a given transmission power, for both networks, the number of receptions and receptions with high-power collision are higher for the indoor environment. This is due to the lower path-loss exponent of the indoor environment that results in larger transmission coverage as shown in Table V. However, as the transmission power goes above 0dBm, the number of receptions with high-power collision reduces in Network1 for the indoor environment. Through increasing the transmission power, the transitional region passes over Network1 and covers the entire network with the connected region. Moreover, since the path-loss exponent of the indoor environment is lower, the transitional region passes over Network1 at lower transmission powers, compared with the outdoor environment. Considering the transmission ranges given in Table V, for transmission powers higher than 0dBm, Network1 located in the indoor environment is completely covered with the connected region; nevertheless, the number of receptions with high-power collision is still remarkable, even with a -100dBm carrier-sensing threshold. In this case, even though the number of hidden-node collisions is about zero, these results show that a large number of receptions are affected by the arrival of high-power interfering signals. Therefore, although the number of collisions (and receptions with high-power collision) reduces when the transmission range covers the entire network, collisions cannot completely be eliminated due to the MAC mechanism. With the utilized CSMA MAC protocol, a collision occurs when at least two nodes

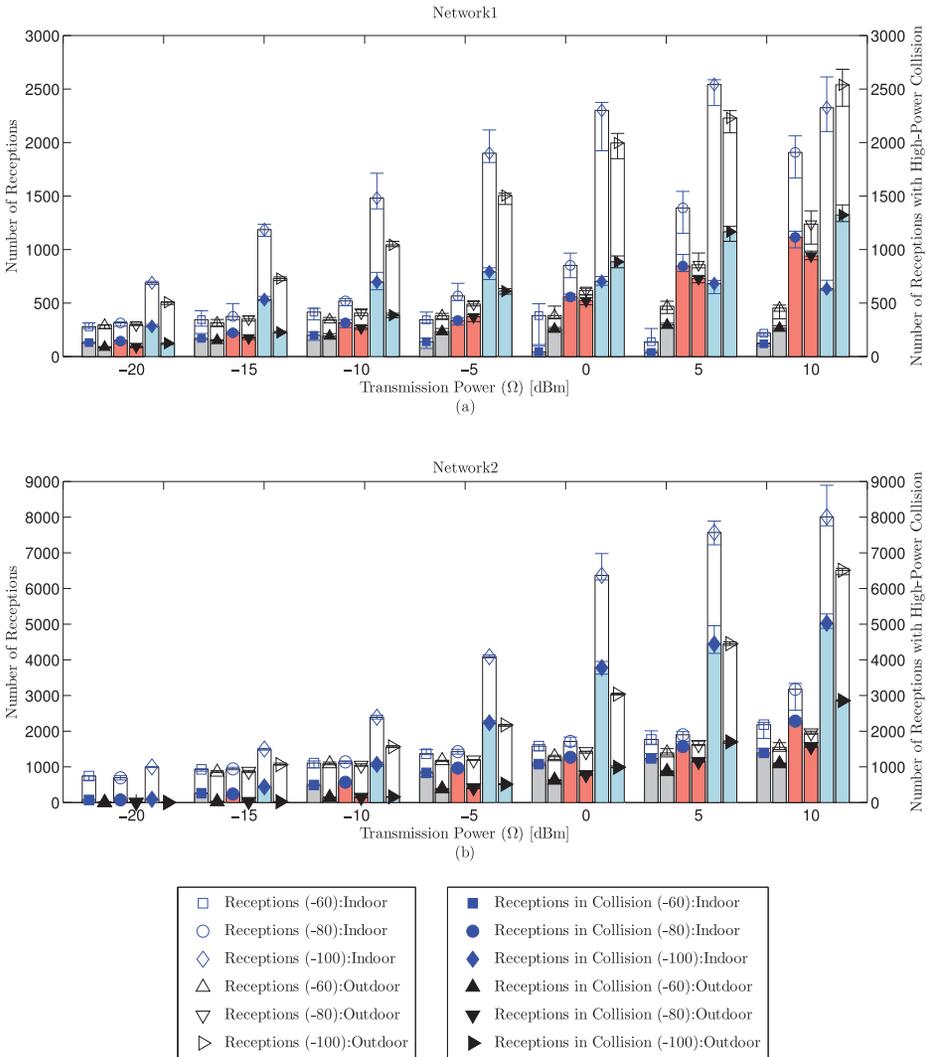


Fig. 21. The number of receptions and the number of receptions with high-power collision in Network1 (a) and Network2 (b). The value in each parenthesis is the carrier-sensing threshold in dBm.

start their transmission at the same back-off slot. In this situation, since the interval between packet receptions is too short, no packet is synchronized with the radio when the other packet arrives. Therefore, the MAC layer will not be able to detect this type of collision. Notice that the concept of back-off slot also exists with non-slotted MAC protocols. In particular, the delays incurred by the propagation delay, carrier sensing, and radio switching can result in concurrent transmissions.

After analyzing the packet reception results, we observed that usually more than 50% of the packet receptions have been involved in a stronger-first capture. However, as only one preamble is received in this type of capture, it cannot be accompanied with collision detection. Using byte-based radios (such as CC1000) through which every transmitted byte can be controlled by the micro-controller [Dezfouli et al. 2014a], inserting specific bytes at the end of the packets may enable the stronger-first collisions to

Table V. The Start (90% Packet Reception Rate) and End (10% Packet Reception Rate) of the Transitional Region for the Indoor and Outdoor Environments. Note that the start of the transitional region corresponds to the connected region size.

Environment	Indoor		Outdoor	
$\eta$	3.3		4.7	
$\sigma_{ch}$	5.5		3.2	
$\Omega_i [dBm]$	Transitional Region		Transitional Region	
	Start [m]	End [m]	Start [m]	End [m]
-20	2.6	9.3	2.2	4.2
-15	3.7	13.1	2.8	5.4
-10	5.2	18.6	3.6	6.9
-5	7.4	26.4	4.6	8.8
0	10.5	37.4	5.9	11.3
5	14.9	53	7.5	14.4
10	21.1	75.1	9.6	18.4

be indicative of collision. Although we have not employed this approach, these results indicate the potential benefits of supporting stronger-first collision detection. To this aim, the following challenges and tradeoffs should be considered: (1) This approach requires equal-size packets to make sure that the secondly arrived packet is not completely covered by the first packet, and the extra synchronization bytes of the secondly arrived packet can be received. (2) Since packet-based radios (such as CC2420) perform packet formatting and preamble generation at the hardware level, packet reformatting is not straightforward. Additionally, as hardware cannot detect these collisions, software should read and analyze the data bytes continuously. (3) Inserting synchronization bytes at the end of packets may introduce new transmission overhead with respect to the small packet size used in sensor networks. Although CAMA does not support packet-end synchronization, it is straightforward to add this feature to its algorithm. Evaluating the potential benefits of utilizing packet-end synchronization is left as a future work.

## 7.2. Sensitivity Analyses

In this section, we investigate the influence of various network parameters on the capture effect. We present the actual number of collision detections and the number of high-power collisions for the investigated parameters. Furthermore, in order to provide comparative metrics for studying the capture effect, we use the “detection efficiency” and “reception efficiency” metrics, as defined earlier. While detection efficiency denotes the probability of collision detection for a given high-power collision, reception efficiency implies the probability of packet reception when a high-power collision happens. Therefore, reception efficiency reflects the benefits of the capture effect in improving network throughput. Our investigations in this section are performed with Network2.

*7.2.1. Environmental Parameters and Carrier-Sensing Threshold.* Environmental parameters depend on the environment in which the network is deployed. The carrier-sensing threshold is a MAC-layer parameter that can be adjusted by the system engineer. As Figure 22 shows, at a given transmission power, the number of collision detections is higher for the indoor environment, which is mainly due to the higher number of high-power collisions. The higher number of high-power collisions in the indoor environment is due to two reasons: First, and more importantly, the lower path-loss exponent of the indoor environment results in a larger transmission coverage area (cf. Table V). Second, the stronger multipath effect of the indoor environment causes a larger transitional region, which affects the carrier-sensing accuracy and may result in signal collision at distant nodes. As described earlier, a collision can be detected when a node

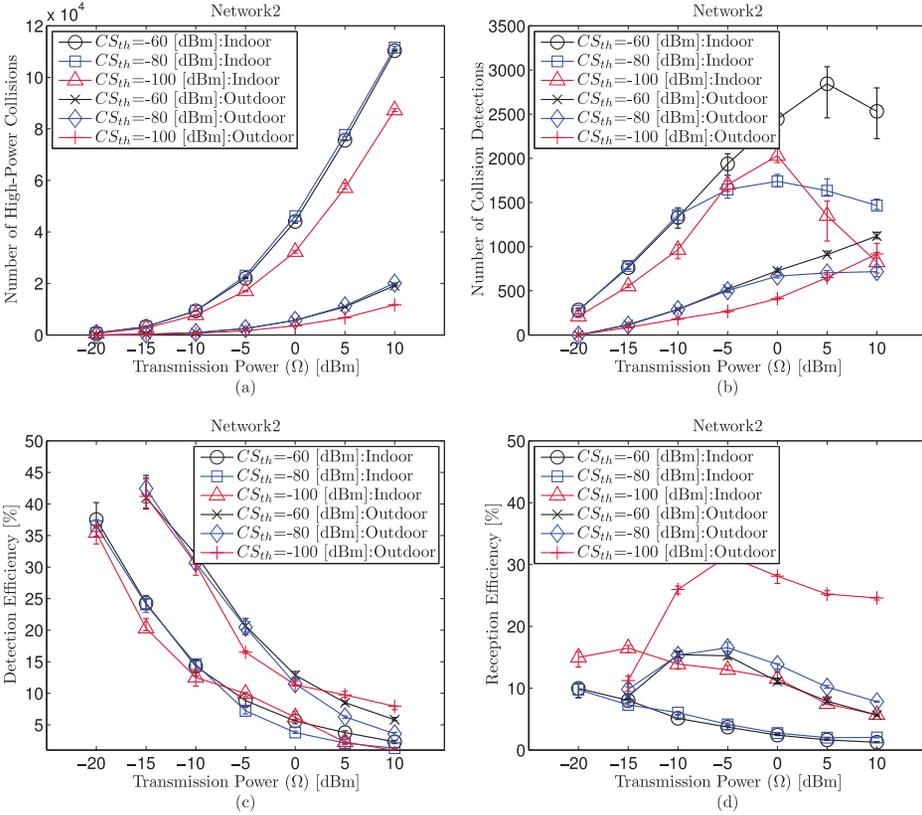


Fig. 22. The effects of environmental parameters and carrier-sensing threshold on collision detection and packet reception efficiency. The lower path-loss exponent of the indoor environment intensifies the number of high-power collisions, which results in lower detection efficiency and lower reception efficiency, compared with the outdoor environment.

receives the preamble bytes of a packet during the MPDU reception of another packet. Therefore, since increasing the transmission power causes a higher number of high-power collisions during MPDU reception, the number of collision detections increases versus transmission power. However, at high transmission power levels, when the high-power collisions are mainly caused by the MAC protocol inefficiency, and not because of the hidden-node collisions, high-power collisions cannot be detected due to the short intervals between packet arrivals; therefore, the number of detections is reduced. The rise and fall of the number of collision detections versus transmission power also complies with the empirical results presented in Figures 16 and 17 of this article and with Figure 3 of Whitehouse et al. [2005]. Due to the considerable increase in the number of high-power collisions versus transmission power, all the curves in Figure 22(c) display a downward trend. Moreover, although Figure 22(b) shows the higher number of collision detections for the indoor environment, Figure 22(c) shows lower detection efficiency, which is due to the significantly higher number of high-power collisions in the indoor environment.

While detection efficiency depends on the number of stronger-last collisions with sufficient interpacket arrival time, reception efficiency depends on the number of receptions in stronger-last and stronger-first collisions, regardless of the interpacket arrival time. Therefore, increasing the number of high-power collisions results in the higher number of receptions with high-power collision. However, as the number of

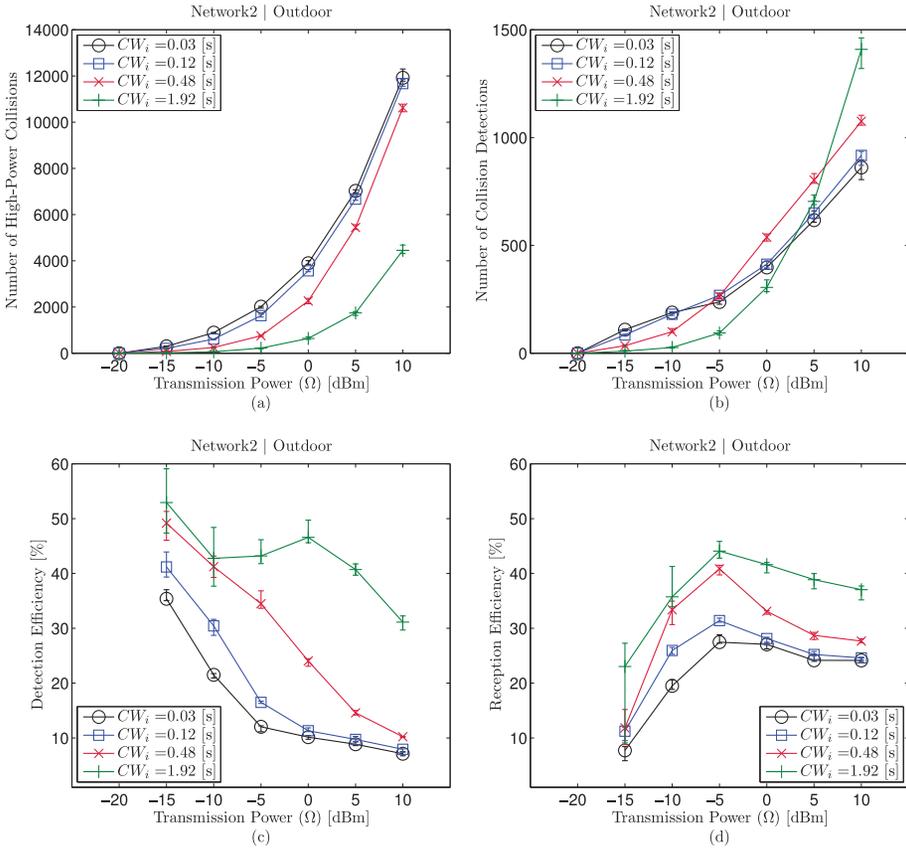


Fig. 23. The effects of contention window duration on collision detection and packet reception efficiency. Although a larger contention window size reduces the number of signal collisions, it improves detection efficiency through reducing the probability of selecting the same back-off slot by multiple senders.

high-power collisions is lower for the outdoor environment, a rise and fall trend can be observed for the reception efficiency. Although Figure 22(c) exhibits approximately similar detection efficiencies for the three carrier-sensing thresholds, Figures 22(a) and (d) show the effects of reducing the number of high-power collisions through carrier-sensing threshold on the reception efficiency. A higher carrier-sensing threshold reduces the carrier-sensing range and intensifies the number of hidden-node collisions during a packet reception. Therefore, although the interval between hidden-node collisions is usually long enough for collision detection, increasing the number of hidden-node collisions during a packet reception enhances the chance of packet corruption.

**7.2.2. Contention Window Size.** Contention window size is a MAC-layer parameter. Figure 23 shows the benefits of increasing the contention window size. Specifically, the higher value of the contention window size results in the lower number of high-power collisions, higher detection efficiency, and improved reception efficiency. It has been stated earlier that a collision caused by multiple transmissions at the same back-off slot cannot eventuate in collision detection. Consequently, increasing the contention window size improves detection efficiency through reducing the probability of selecting the same back-off slot by multiple senders. Moreover, a larger contention window size reduces the probability of hidden-node collisions when the senders cannot hear each other's transmissions [Dezfouli et al. 2014b]. Generally, as increasing the contention

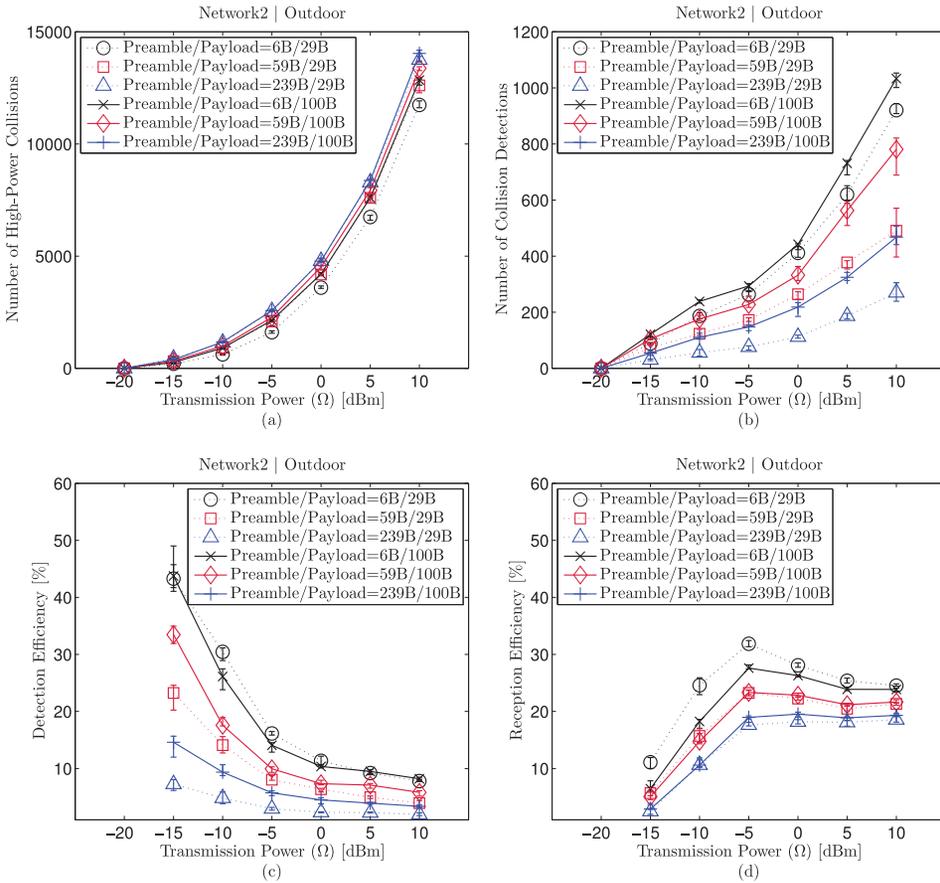


Fig. 24. The effects of packet formatting on collision detection and packet reception efficiency. Detection efficiency reduces as the preamble to payload ratio grows. This implies the lower probability of collision detection when the MAC layer employs long preambles for duty cycle reduction.

window size reduces the number of collisions during packet reception, the probability of packet corruption reduces and causes higher reception efficiency. Figure 23 shows that both 0.48sec and 1.92sec can effectively reduce the number of high-power collisions at low transmission powers; therefore, they represent similar performance. For example, about 50% of the high-power collisions can be detected due to large interpacket intervals. However, as the transmission power increases and internode contention grows, the collision avoidance capability of the 1.92sec becomes more apparent. For example, at 0dBm, the 1.92sec improves the detection efficiency by about 96% over the 0.48sec.

**7.2.3. Preamble and Payload Size.** The next investigated parameters are the preamble and payload size, which are the MAC and network layer parameters, respectively. Figure 24 shows the results. Dotted lines and solid lines have been used for 29-byte and 100-byte payloads, respectively. It is obvious that a larger packet size intensifies network traffic and causes a higher number of collisions. Although reception efficiency mainly depends on the total packet size, increasing the number of collisions through larger packet size does not improve the collision detection capability. For example, while the packet size for 6B/29B is shorter than that of 59B/29B, Figure 24(b) shows a significantly higher number of collision detections for 6B/29B. Moreover, although the

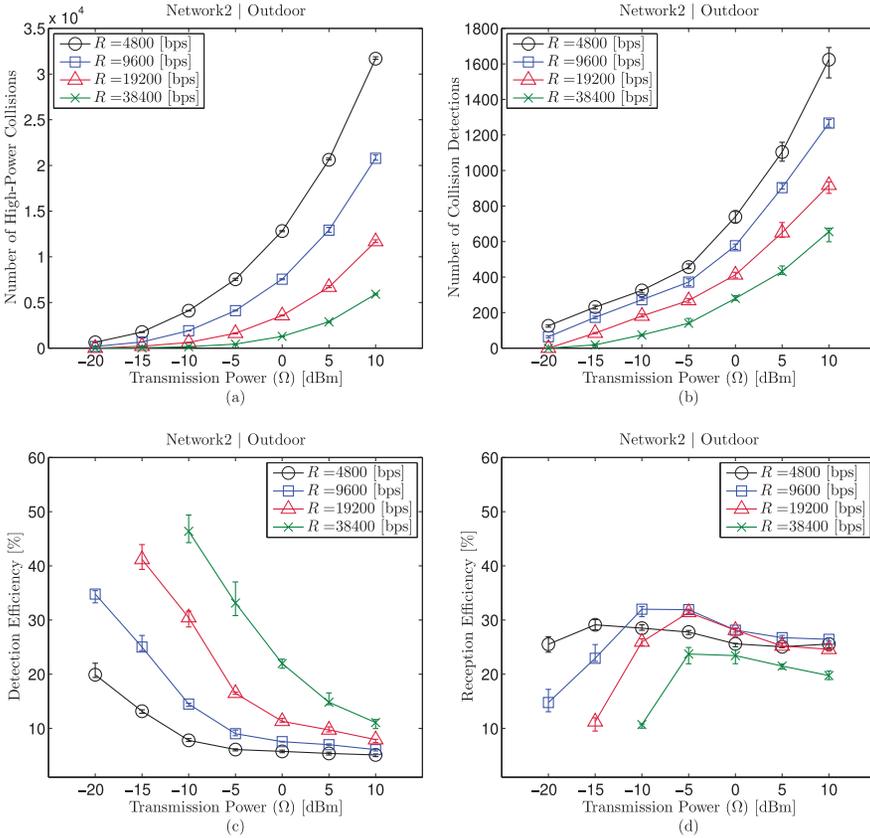


Fig. 25. The effects of radio bit rate on collision detection and packet reception efficiency. Increasing radio bit rate results in higher detection efficiency and lower reception efficiency.

packet sizes of 59B/29B and 6B/100B differ by 18 bytes, Figure 24(b) shows that the number of collision detections is higher for 6B/100B. A similar behavior can also be observed in Figure 24(c). The three-node experiments presented in Section 6 showed that the packet portion during which collision can be detected becomes shorter as the preamble size enlarges. This is because enlarging the preamble size increases the probability of collision during preamble reception. These results reveal the impact of utilizing long preambles when the MAC layer employs a long preamble to reduce duty cycle and improve energy consumption. In this instance, in order to improve detection efficiency, network parameters such as carrier-sensing threshold and contention window size should be carefully adjusted to reduce the number of collisions during preamble reception. Another solution to support collision detection with partially overlapped preambles would be to divide a large preamble into several small subpreambles with distinct patterns. In this case, a collision can be detected when a received subpreamble violates the expected pattern. Figure 24 also shows that for a given preamble size, a longer payload causes more collision detections. This is due to the increase in the number of high-power collisions that occurred during the payload reception.

**7.2.4. Radio Bit Rate.** Figure 25 shows the results of our sensitivity analysis for various radio speeds. According to Equation (5), for a given SINR value, a linear increase in the bit rate causes an exponential increase in BER. Moreover, when the bit rate is doubled,

the transmission power should also be doubled to achieve the same bit error probability. That is why there is no network connectivity when the bit rate is 38,400bps and the transmission power equals  $-20$  or  $-15$ dBm. Therefore, at a given transmission power, increasing the bit rate produces a lower number of high-power collisions. On the other hand, a faster radio reduces the channel busy time through shortening packet transmission duration. Therefore, the transmission speed significantly affects the number of collisions. Figure 25(c) shows the higher detection efficiency for the faster radio, because enhancing the bit rate causes a higher reduction in the number of high-power collisions, compared with the number of collision detections. In contrast, Figure 25(d) shows a lower reception efficiency for the higher bit rate. As increasing the bit rate raises the required SINR value for successful packet reception, this figure shows that the effect of bit rate on SINR threshold is higher than its influence on the number of collisions.

## 8. CONCLUSIONS

In this paper, we introduced CAMA, an algorithm for improving the accuracy and efficiency of modeling the capture effect in low-power wireless networks. In addition to accuracy and scalability, CAMA also provides partial packet reception and enables the higher layers to obtain information about the collision status at the physical layer. These features provide opportunities for protocol improvements that have not been possible with packet-level simulators such as NS2.

Although CAMA has been presented and evaluated with respect to the characteristics of low-power radio transceivers, it is a general algorithm and can be integrated with various link models. Moreover, CAMA is a physical-layer algorithm and it is independent of MAC-layer implementation. In addition to these features, since CAMA supports preamble and packet body capture, it can also be used for simulating 802.11 networks where the reception probabilities of preamble and packet body are different. We leave this study as a future work.

We performed experiments with various radios and under different topologies to validate the operation of CAMA. While CAMA could regenerate the empirical traces, we precisely highlighted the inherent inaccuracies of the capture models employed by NS2. Furthermore, through evaluating CAMA and NS2 in various scenarios, we showed that the capture models of NS2 cannot provide accurate collision detection information. These results also showed that, using small-size preambles, the simulation speed of CAMA is less than 30% lower than that of NS2 capture models. However, as the preamble size enlarges (i.e., duty cycle reduces), the speed improvement of CAMA is higher than that of NS2.

Our investigations on the capture effect in sensor networks show that a significant number of received packets are affected by high-power interfering signals. In other words, a large number of received packets experience at least one high-power collision during their reception duration. Furthermore, even when the transmission power is high enough to cover the entire network, there are a considerable number of collisions caused by the MAC protocol inefficiency. Accordingly, while these results show the influence of the capture effect on packet reception performance, they also confirm the importance of modeling the capture effect for the performance evaluation of wireless networks. Although these results indicate a significant number of collisions in the flooding traffic pattern, the number of collisions would be even higher during the data-gathering phase of sensor networks. This is because of the convergecast traffic pattern in which all the nodes send their packets toward a common destination. Therefore, a large number of collisions is expected near the sink node. Analyzing the capture effect with the convergecast traffic pattern and various MAC protocols is a potential area of future work.

Our studies on the sensitivity of the capture effect to various network parameters revealed the following key findings: (1) The lower path-loss exponent of the indoor environment intensifies the number of collisions, which results in lower detection efficiency and lower reception efficiency, compared with the outdoor environment. (2) Although a larger contention window size reduces the number of signal collisions, it improves detection efficiency through reducing the probability of selecting the same back-off slot by multiple senders. (3) Detection efficiency reduces as the preamble-to-payload ratio grows. This implies the lower probability of collision detection when the low-power listening technique is used at the MAC layer. (4) Increasing the radio bit rate results in higher detection efficiency and lower reception efficiency.

## REFERENCES

- S. Abukharis, R. MacKenzie, and T. O'Farrell. 2011. Throughput and delay analysis for a differentiated p-persistent CSMA protocol with the capture effect. In *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.
- M. Al-Bado, C. Sengul, and R. Merz. 2012. What details are needed for wireless simulations? A study of a site-specific indoor wireless model. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12)*. IEEE, 289–297.
- J. C. Arnbak and W. Van Blitterswijk. 1987. Capacity of slotted ALOHA in rayleigh-fading channels. *IEEE Journal on Selected Areas in Communications* 5, 2 (Feb.), 261–269.
- P. Cardieri. 2010. Modeling interference in wireless ad hoc networks. *IEEE Communications Surveys & Tutorials* 12, 4, 551–572.
- CC1120 2014. High Performance RF Transceiver for Narrowband Systems. Retrieved from [www.ti.com/product/cc1120](http://www.ti.com/product/cc1120).
- Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi. 2006. IEEE 802.11 based vehicular communication simulation design for NS-2. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks (VANET'06)*. ACM, 50.
- Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein. 2007. Overhaul of IEEE 802.11 modeling and simulation in ns-2. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM'07)*. ACM, 159.
- K. Cheun and S. Kim. 1998. Joint delay-power capture in spread-spectrum packet radio networks. *IEEE Transactions on Communications* 46, 4 (April), 450–453.
- Chipcon CC1000. 2014. Single Chip Very Low Power RF Transceiver. Retrieved from [www.ti.com/lit/ds/symlink/cc1000.pdf](http://www.ti.com/lit/ds/symlink/cc1000.pdf).
- Chipcon CC2420. 2014. 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver. Retrieved from <http://www.ti.com/lit/gpn/cc2420>.
- F. Daneshgaran, M. Laddomada, F. Mesiti, M. Mondin, and M. Zanolò. 2008. Saturation throughput analysis of IEEE 802.11 in the presence of non ideal transmission channel and capture effects. *IEEE Transactions on Communications* 56, 7 (July), 1178–1188.
- D. H. Davis and S. A. Gronemeyer. 1980. Performance of slotted ALOHA random access with delay capture and randomized time of arrival. *IEEE Transactions on Communications* 28, 5 (May), 703–710.
- B. Dezfouli, M. Radi, S. A. Razak, T. Hwee-Pink, and K. A. Bakar. 2014a. Modeling low-power wireless communications. *Journal of Network and Computer Applications*. DOI: <http://dx.doi.org/10.1016/j.jnca.2014.02.009>.
- B. Dezfouli, M. Radi, S. A. Razak, K. Whitehouse, K. A. Bakar, and T. Hwee-Pink. 2014b. Improving broadcast reliability for neighbor discovery, link estimation and collection tree construction in wireless sensor networks. *Computer Networks* 62, 101–121.
- P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis. 2010. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*. ACM, 1.
- B. Firner, C. Xu, R. Howard, and Y. Zhang. 2010. Multiple receiver strategies for minimizing packet loss in dense sensor networks. In *Proceedings of the 11th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'10)*. ACM, 211.
- S. Ganu, K. Ramachandran, M. Gruteser, I. Seskar, and J. Deng. 2006. Methods for restoring MAC layer fairness in IEEE 802.11 networks with physical layer capture. In *Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality (REALMAN'06)*. ACM, 7.

- C. Gezer, C. Buratti, and R. Verdone. 2010. Capture effect in IEEE 802.15.4 networks: Modelling and experimentation. In *Proceedings of the IEEE 5th International Symposium on Wireless Pervasive Computing (ISWPC'10)*. IEEE, 204–209.
- P. Gupta and P. R. Kumar. 2000. The capacity of wireless networks. *IEEE Transactions on Information Theory* 46, 2 (March), 388–404.
- Z. Hadzi-Velkov and B. Spasenovski. 2002. On the capacity of IEEE 802.11 DCF with capture in multipath-faded channels. *International Journal of Wireless Information Networks* 9, 3, 191–199.
- E. B. Hamida, G. Chelius, J. M. Gorce, and E. Ben Hamida. 2009. Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. *Simulation* 85, 9 (June), 574–588.
- C. Hu and J. C. Hou. 2005. A reactive channel model for expediting wireless network simulation. *SIGMETRICS Performance Evaluation Review* 33, 1, 410.
- IEEE 802.15.4™-2011. Wireless personal area networks (PANs). Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). DOI: <http://standards.ieee.org/about/get/802/802.15.html>
- A. Iyer, C. Rosenberg, and A. Karnik. 2009. What is the right model for wireless channel interference? *IEEE Transactions on Wireless Communications* 8, 5 (May), 2662–2671.
- J. H. Kim and J. K. Lee. 1999. Capture effects of wireless CSMA/CA protocols in Rayleigh and shadow fading channels. *IEEE Transactions on Vehicular Technology* 48, 4 (July), 1277–1286.
- A. Kochut, A. Vasan, A. Shankar, and A. Agrawala. 2004. Sniffing out the correct physical layer capture model in 802.11b. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)*. IEEE, 252–261.
- J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. 2007. An experimental study on the capture effect in 802.11a networks. In *Proceedings of the the 2nd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH'07)*. ACM, 19.
- J. Lee, J. Ryu, S.-J. Lee, and T. T. Kwon. 2010. Improved modeling of IEEE 802.11a PHY through fine-grained measurements. *Computer Networks* 54, 4 (March), 641–657.
- P. Levis, N. Lee, M. Welsh, and D. Culler. 2003. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*. ACM, 126–137.
- S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. 2006. ATPC: Adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM, 223.
- S. Lin, G. Zhou, Y. Wu, K. Whitehouse, J. A. Stankovic, and T. He. 2008. Achieving stable network performance for wireless sensornetworks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys'08)*. ACM, 453.
- J. Lu and K. Whitehouse. 2009. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *Proceedings of the 28th Conference on Computer Communications (INFOCOM'09)*. IEEE, 2491–2499.
- R. Maheshwari, S. Jain, and S. R. Das. 2008. On estimating joint interference for concurrent packet transmissions in low power wireless networks. In *Proceedings of the 3rd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH'08)*. ACM, 89.
- M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. 2004. The flooding time synchronization protocol. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM, 39.
- H. Nikoogar and H. Hashemi. 1993. Statistical modeling of signal amplitude fading of indoor radio propagation channels. In *Proceedings of 2nd IEEE International Conference on Universal Personal Communications*, Volume 1. IEEE, 84–88.
- NS-2. 2014. Network simulator. <http://www.isi.edu/nsnam/ns/>.
- OMNeT++. 2014. The OMNeT++ Network Simulation Framework. Retrieved from <http://www.omnetpp.org>.
- J. Polastre, J. Hill, and D. Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM, 95.
- M. Radi, B. Dezfouli, K. Abu Bakar, and S. Abd Razak. 2014. Integration and analysis of neighbor discovery and link quality estimation in wireless sensor networks. *Scientific World Journal* 2014, 1–23.
- M. Radi, B. Dezfouli, K. A. Bakar, S. A. Razak, and M. Lee. 2013. Network initialization in low-power wireless networks: A comprehensive study. *Computer Journal*. DOI: <http://dx.doi.org/10.1093/comjnl/bxt074>
- M. Radi, B. Dezfouli, K. A. Bakar, S. A. Razak, and M. A. Nematbakhsh. 2011. Interference-aware multipath routing protocol for QoS improvement in event-driven wireless sensor networks. *Tsinghua Science & Technology* 16, 5 (Oct.), 475–490.

- T. S. Rappaport. 2002. *Wireless Communications: Principles and Practice* (2nd ed.). Prentice Hall.
- C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. 2006. Measurement-based models of delivery and interference in static wireless networks. *ACM SIGCOMM Computer Communication Review* 36, 4 (Aug.), 51.
- D. Son, B. Krishnamachari, and J. Heidemann. 2006. Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the 4th international conference on embedded networked sensor systems - SenSys'06*, Boulder, Colorado, USA, pp. 237. ACM Press.
- TOSSIM. 2014. TinyOS Simulator. Retrieved from <http://docs.tinyos.net/tinywiki/index.php/TOSSIM>.
- C. Ware, J. Chicharo, and T. Wysocki. 2001. Simulation of capture behaviour in IEEE 802.11 radio modems. In *Proceedings of the IEEE 54th Vehicular Technology Conference (VTC Fall'01)*, Vol. 3. IEEE, 1393–1397.
- K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. 2005. Exploiting the capture effect for collision detection and recovery. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*. IEEE, 45–52.
- H. Xu, J. J. Garcia-Luna-Aceves, and R. S. Hamid. 2010. Exploiting the capture effect opportunistically in MANETs. In *Proceedings of the Military Communication Conference (MILCOM'10)*. IEEE, 1490–1495.
- J.-H. Yun and S.-W. Seo. 2007. Novel collision detection scheme and its applications for IEEE 802.11 wireless LANs. *Computer Communications* 30, 6 (March), 1350–1366.
- M. Z. Zamalloa and B. Krishnamachari. 2007. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Transactions on Sensor Networks* 3, 2 (June), 63–81.
- L. Q. Zhang, F. Wang, M. K. Han, and R. Mahajan. 2007. A general model of wireless interference. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'07)*. ACM, 171.
- G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. 2006. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks* 2, 2 (May), 221–262.

Received April 2013; revised January 2014; accepted May 2014