Mobile Information Systems 7 (2011) 69–92
DOI 10.3233/MIS-2011-0111
IOS Press

IEEE 802.15.4 modifications and their impact

M. Goyal, W. Xie and H. Hosseini
Department of Computer Science, University of Wisconsin Milwaukee, Milwaukee WI 53201, USA
E-mail: {mukul,wxie,hosseini}@uwm.edu

Abstract. IEEE 802.15.4 is a popular choice for MAC/PHY protocols in low power and low data rate wireless sensor networks. In this paper, we suggest several modifications to beaconless IEEE 802.15.4 MAC operation and evaluate their impact on the performance via stochastic modeling and simulations. We found that the utility of these modifications is strongly dependent on the traffic load on the network. Accordingly, we make recommendations regarding how these modifications should be used in view of the prevalent traffic load on the network.

1. Introduction

IEEE 802.15.4 [1] provides *physical* (PHY) and *medium access control* (MAC) layer functionality in low power and low data rate wireless sensor networks (WSN). Wireless communication among sensor devices, enabled by IEEE 802.15.4 technology, is increasingly replacing the existing *wired* technology in a wide range of monitoring and control applications in home, urban, building and industrial environments [2–7,10,12,13,15–23]. IEEE 802.15.4 MAC operation is based on *carrier sense multiple access* with *collision avoidance* (CSMA/CA). Thus, an IEEE 802.15.4 node competes with all the nodes in its radio range for access to the transmission channel.

The CSMA/CA based networks are known to suffer from performance deterioration with increase in the number of nodes competing for channel access at any given time, which in turn depends on both the total number of nodes in each other's radio range and their packet generation rates. Future IEEE 802.15.4 networks may consist of several thousand nodes distributed over a large area and an individual node may possibly have hundreds of nodes in its radio range [3]. Also, it is not uncommon for IEEE 802.15.4 nodes to have high packet generation rates over some time intervals. A combination of large number of nodes in each other's radio range and high packet generation rates may have a severe impact on the performance of the wireless sensor network in terms of the packet loss probability and the packet latency.

In [9], we developed a stochastic model for popular *beaconless* operation of IEEE 802.15.4 MAC protocol. Given the number of nodes competing for channel access and their packet generation rates, the model can very accurately predict the packet loss probability as well as the packet latency and thus serve as a useful tool in the design of large scale wireless sensor networks based on IEEE 802.15.4. In this paper, we use this model to investigate the efficacy of various proposals to improve the performance of becaonless IEEE 802.15.4 MAC protocol.

The rest of the paper is organized as follows. Next two sections present background information regarding IEEE 802.15.4. Section 2 presents a brief overview of beaconless IEEE 802.15.4 MAC operation and describes *channel access failure* and *collision failure* that a node may encounter in the process of sending a packet. Section 3 describes different ways a packet collision may occur in

1574-017X/11/$27.50 © 2011 – IOS Press and the authors. All rights reserved

the operation of beaconless IEEE 802.15.4 network. In Section 4, we describe the stochastic model presented in [9]. Section 5 describes several modifications to IEEE 802.15.4 beaconless MAC operation and use the stochastic model to evaluate their impact on performance. Based on this evaluation, we make recommendations regarding the modifications to be used for a given traffic load on the network. We demonstrate that these recommendations, derived using the stochastic model, result in better performance in simulations as well. Finally, Section 6 concludes the paper.

2. About IEEE 802.15.4

As mentioned before, IEEE 802.15.4 protocol provides PHY and MAC layer functionality in low power and low data rate wireless sensor networks. Typically, IEEE 802.15.4 constitutes the PHY/MAC layer of a larger protocol suite (e.g. Zigbee [24]), where the upper layers provide multi-hop routing and other functionality to allow formation of large-scale wireless sensor networks. In the following discussion, we assume the popular 2.4 GHz range operation of IEEE 802.15.4 PHY layer, where the information is sent 4 bits, or a *symbol*, at a time and maximum data rate is 250 kbps (or 62500 symbols per second). Various time durations are expressed in terms of symbols.

IEEE 802.15.4 MAC operation has two modes – *beacon-enabled* and *beaconless*. The beacon-enabled mode allows splitting of time into multiple *active* durations with a *cluster* [1] having exclusive access to the transmission channel during its active duration. The coordinator broadcasts a *beacon* to inform other nodes in the cluster about the beginning of the cluster's active duration. The cluster nodes compete for channel access during their active period using a *slotted* CSMA/CA algorithm. In the beaconless operation, there is no division of time into active durations and a node competes for channel access with other nodes in its radio range using an unslotted CSMA/CA algorithm. In this paper, we have focussed on the beaconless operation of IEEE 802.15.4 MAC layer.

As per the unslotted CSMA/CA algorithm, the source node begins a transmission attempt with a CSMA wait for a random number of *backoff* periods ($=$ 20 symbols each) between 0 and $2^{BE} - 1$. Here, *BE* refers to a variable called the *backoff exponent* that is initially set to the value of *macMinBE* parameter (by default 3). After the CSMA wait, the source node determines if the channel is available for transmission. This *clear channel assessment* (CCA) is performed over a time duration of 8 symbols. If the CCA fails (i.e. the channel is found to be busy), the node increments *BE* (up to the value of *macMaxBE* parameter, which is 5 by default) and repeats the procedure. If the CCA fails even after *macMaxCSMABackoffs* (by default 4) CSMA waits, a *channel access failure* is declared and any further attempt to transmit the packet is abandoned. If the CCA succeeds, the source node performs an *RX-to-TX* turnaround [2] and transmits the packet. The propagation delay for the packet is expected to be negligible. On receiving the packet, the destination node performs an *RX-to-TX* turnaround and sends the acknowledgement (ACK) if required by the source. No CSMA wait is performed for ACK transmission. As discussed in Section 3, the transmitted packet or its ACK may suffer a collision. In this case, the source node waits for the *macAckWaitDuration* (typically 54 symbols for 2.4GHz operation) for the ACK to arrive and then proceeds with next attempt to transmit the packet. The source node can make upto *macMaxFrameRetries* (3 by default) further attempts to transmit the packet and receive the ACK. The failure to receive an ACK

[1] consisting of a *coordinator* and its *associated* nodes

[2] The IEEE 802.15.4 nodes are typically *half-duplex* in nature, i.e. they can not perform both the *transmit* (TX) and *receive* (RX) operations at the same time. The IEEE 802.15.4 specification [1] requires the *RX-to-TX* or *TX-to-RX* turnaround time to be 12 symbols or less.

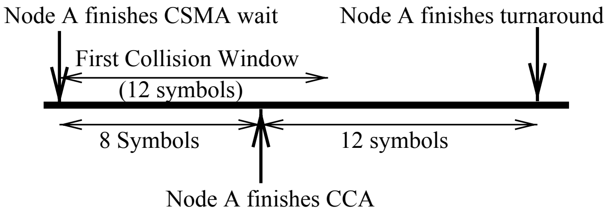


Fig. 1. First Collision Window.

even after *macMaxFrameRetries* +1 attempts causes the IEEE 802.15.4 MAC layer to accept failure in sending the packet. Such a failure is referred to as *collision failure* in the following discussion.

3. Packet collision scenarios in beaconless IEEE 802.15.4 networks

Even though the IEEE 802.15.4 nodes use a *collision avoidance* algorithm to compete for channel access, collisions do occur for reasons described next.

Hidden nodes: Some nodes in a WSN may not be in the hearing range of a node (say node *X*) and hence may transmit a packet at the same time as node *X*. Such nodes are called *hidden* nodes for node *X*. However, if node *Y*, the destination of node X's transmissions, can hear these hidden nodes, any concurrent transmission by a hidden node would cause node *Y* to drop node *X*'s transmission.

Collisions due to turnaround time: As mentioned earlier, an IEEE 802.15.4 node may take upto 12 symbols to turn around from *RX* mode to *TX* mode and vice-versa. This non-negligible turnaround time may cause packet collisions to take place in the following situations:

- Suppose, a number of nodes, all in each other's hearing range, are competing for channel access and all of them are doing the CSMA wait at a certain time, hence the transmission channel is idle. Suppose, node *A* is the first node to wake up at time t. Node *A* performs a CCA till time $t+8$, which is guaranteed to succeed, and then performs an *RX-to-TX* turnaround that finishes at time $t + 20$. The transmission channel would continue to be idle until time $t + 20$ when node *A* begins its packet transmission. Thus, if another node finishes its CSMA wait between times t and $t + 12$, its CCA would succeed and its subsequent packet transmission would collide with that of node *A*. Figure 1 refers to this 12 symbol duration as the *first collision window*. Note that the first collision window is actually equal to the *RX-to-TX* turnaround time.
- A destination node (say *B*) needs to complete an *RX-to-TX* turnaround before it can send the ACK for a packet. If another node finishes its CSMA wait during the first 4 symbols of this turnaround, its CCA would succeed and its packet transmission would collide with node *B*'s ACK. Figure 2 refers to this 4 symbol duration as the *second collision window*. Note that the second collision window is the result of CCA duration being less than the *RX-to-TX* turnaround time. Also, note that the second collision window exists only if no collision takes place in the first collision window.
- A destination node would ignore a packet transmission if it begins before the destination has completed the *TX-to-RX* turnaround after sending the ACK for the previous transmission. Even though this situation does not involve a collision, its impact is same as that of a collision.

Corrruption due to PHY noise/interference: A packet transmission may get incorrigibly corrupted due to PHY level noise or interference from sources like microwave ovens or WiFi transmissions. The consequent discarding of the packet transmission by the destination has the same impact as a collision.

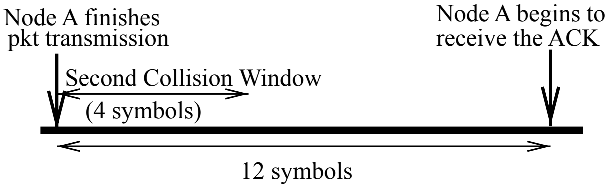

Node A finishes pkt transmission
Node A begins to receive the ACK
Second Collision Window
(4 symbols)
12 symbols

Fig. 2. Second Collision Window.

4. A stochastic model for beaconless IEEE 802.15.4 MAC operation

In [9], we developed a stochastic model to predict the packet loss probability and latency for a group of n source nodes in a network that are in each other's radio range and hence compete with each other for channel access to send packets to one or more destination nodes. The time interval, t, between two consecutive packet send events at each node is assumed to be exponentially distributed with rate $1/T$, i.e. $T = E(t)$. The model further assumes that the *clear channel assessment* (CCA) fails if there is a transmission by any node in the radio range during any part of the CCA duration; otherwise, the CCA succeeds.

The stochastic model takes n and T values as input and generates the expected values for the packet loss probability, $L(n,T)$, and the packet latency, $D(n,T)$, at steady state. Let m be the random variable denoting the number of *active* nodes at any given time. A node is considered *active* only while it has a packet to send and hence is competing for channel access with other active nodes. We denote the corresponding probability of CCA failure as $\alpha(m)$ and the corresponding probability of collision for a transmission as $\beta(m)$. The $\{\alpha, \beta\}$ values for a given m can be used to determine the corresponding packet loss probability $\lambda(m)$ and the corresponding packet latency $\delta(m)$.

The number of active nodes vary with time depending on the total number of source nodes (n) and their packet generation behavior characterized by average inter-packet interval T. The stochastic model is based on the assumption that the probability that m nodes are active at any given time is same as the probability that $m - 1$ nodes get a new packet to send while an active node is sending its current packet, i.e. during a time interval equal to the steady state packet latency $D(n,T)$. Thus, for given values of n and T, m is a function of packet latency $D(n,T)$, which in turn is a function of $\delta(m)$ and hence m. The model exploits this cyclic relationship between m and $D(n,T)$ to determine $D(n,T)$, which can readily be translated to the expected value for packet loss probability $L(n,T)$.

4.1. Modeling the CSMA wait duration

During a packet transmission attempt, a node does a CSMA wait before performing the CCA. The CCA failure causes *BE* value to be incremented (up to *macMaxBE*, by default 5) and the CSMA wait to be repeated (up to *macMaxCSMABackoffs* times, by default 4). The CSMA wait duration is a randomly selected number of *backoff periods* (= 20 symbols each) in the range $[0, 2^{BE} - 1]$, where each number in the range is equally likely to be selected. With α as the probability of CCA failure, a node would perform on average $\boldsymbol{1} + \boldsymbol{\alpha} + \boldsymbol{\alpha}^2 + \boldsymbol{\alpha}^3 + \boldsymbol{\alpha}^4$ CSMA waits during a packet transmission attempt. Let w denote the random variable corresponding to a CSMA wait duration. With *macMinBE* and *macMaxBE* parameters at their default values (3 and 5 respectively), i.e., with 3 and 5 as the initial and maximum values of *BE*, the probability that a CSMA wait is equal to i backoff periods, where $0 \leqslant i \leqslant 31$, is:

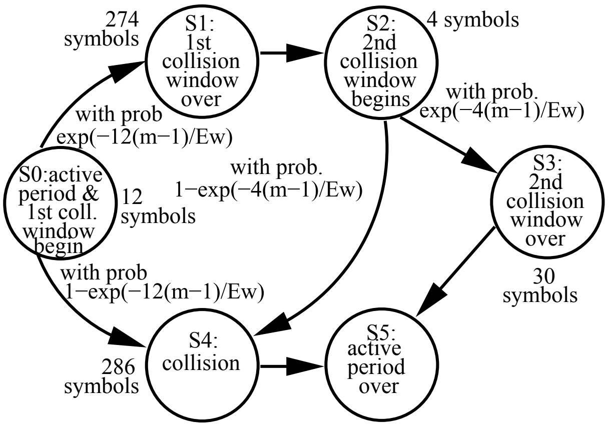

274 symbols
S1: 1st collision window over
S2: 2nd collision window begins
4 symbols
with prob. exp(−4(m−1)/Ew)
with prob exp(−12(m−1)/Ew)
with prob. 1−exp(−4(m−1)/Ew)
S0:active period & 1st coll. window begin
12 symbols
S3: 2nd collision window over
with prob 1−exp(−12(m−1)/Ew)
30 symbols
S4: collision
286 symbols
S5: active period over

Fig. 3. Different states during the life time of an *active period*.

$$p(w=i)=\begin{cases}\frac{\frac{1}{8}+\frac{1}{16}\alpha+\frac{1}{32}\alpha^2+\frac{1}{32}\alpha^3+\frac{1}{32}\alpha^4}{1+\alpha+\alpha^2+\alpha^3+\alpha^4} & 0\leqslant i\leqslant 7,\\ \frac{\frac{1}{16}\alpha+\frac{1}{32}\alpha^2+\frac{1}{32}\alpha^3+\frac{1}{32}\alpha^4}{1+\alpha+\alpha^2+\alpha^3+\alpha^4} & 8\leqslant i\leqslant 15,\\ \frac{\frac{1}{32}\alpha^2+\frac{1}{32}\alpha^3+\frac{1}{32}\alpha^4}{1+\alpha+\alpha^2+\alpha^3+\alpha^4} & 16\leqslant i\leqslant 31,\\ 0 & i>31\end{cases}$$

The expected value of a CSMA wait duration (in terms of backoff periods) can be expressed as the following continuous and monotone increasing function of α, the probability of CCA failure:

$$E_w(\alpha)=\frac{3.5+7.5\alpha+15.5\alpha^2+15.5\alpha^3+15.5\alpha^4}{1+\alpha+\alpha^2+\alpha^3+\alpha^4} \tag{1}$$

Even though the CSMA wait duration is discrete in nature, we model it as a continuous random variable, w', exponentially distributed with rate $1/E_w$, i.e. $w' \sim \mathrm{EXP}(1/E_w)$.

4.2. *The probability of CCA failure*

Consider m active nodes competing for channel access at a certain time. Suppose all the active nodes are performing their CSMA waits initially. Suppose node *A* is the first node to finish its CSMA wait. We mark this event as the beginning of an *active period*. In other words, node *X triggers* an active period. Node *X* would find the channel idle and proceed with its packet transmission. Thus, node *X* would have zero probability of CCA failure. The active period ends when node *X*, and any other node that begins its transmission during the active period, have finished their transmissions.

If another node, say node *Y*, finishes its CSMA wait during the active period, its CCA would fail unless node *Y*'s CSMA wait ends during one of the *collision windows* (as identified in Section 3) associated with node *X*'s transmission. Figure 3 shows different states during the life time of an active period under the following assumptions:

– $w' \sim \mathrm{EXP}(1/E_w)$ represents the individual CSMA wait time for all the nodes;
– the nodes send 133 byte long packets (including the 6 byte IEEE 802.15.4 PHY header);

- in case of a collision, the last colliding node finishes its CSMA wait just before the end of the collision window.

The *sojourn* times in various states could be explained as follows. State *S0* corresponds to the beginning of the active period as well as that of the first collision window and has a sojourn time equal to 12 symbols. These 12 symbols consist of 8 symbols of CCA by node *X* and the first 4 symbols of node *X*'s *RX-to-TX* turnaround. If at least one active node finishes its CSMA wait during the first collision window, which happens with probability $1 - e^{(-12(m-1)/E_w)}$, a collision is guaranteed. In this case, there is a transition to the *collision* state, *S4*. For simplicity, we assume that the last colliding node finishes its CSMA wait just before the end of the collision window and completes its transmission over next 286 symbols (8 symbols of CCA $+$ 12 symbols of turnaround $+$ 266 symbols of packet transmission), which constitute the sojourn time in state *S4*. The completion of sojourn in state *S4* concludes the active period. On the other hand, if no active node finishes its CSMA wait during the first collision window, which happens with probability $e^{-12(m-1)/E_w}$, node *X* would complete its turnaround (additional 8 symbols) and transmit the packet (266 symbols). These 274 symbols correspond to the sojourn in state *S1*.

The second collision window begins as soon as the destination successfully receives the transmitted packet and starts its *RX-to-TX* turnaround to send the ACK. This corresponds to a transition to state *S2*, where the sojourn time is 4 symbols. If at least one active node finishes its CSMA wait during the second collision window, which happens with probability $1 - e^{(-4(m-1)/E_w)}$, there is a transition to the *collision* state, *S4*. Otherwise, following the completion of 4 symbol long collision window, there is a transition to state *S3*. The sojourn time in state *S3* is 30 symbols during which the destination node completes its *RX-to-TX* turnaround that started in state *S2* (8 symbols) and transmits the 11 byte long ACK (22 symbols). The completion of the ACK transmission completes the active period.

Next, we determine different possible durations of an active period as well as the probability of occurance for each duration. We also determine the probability that an active node finishes its CSMA wait during the active period of a particular duration and the probability of CCA failure for such a node:

- If one or more nodes finish their CSMA waits during the first collision window, which happens with probability $p_1 = 1 - e^{-12(m-1)/E_w}$, the active period would sojourn over state sequence *S0-S4* and last for 298 symbols. The probability that an active node finishes its CSMA wait during such an active period is $q_1 = 1 - e^{-298/E_w}$. Since, the CCA would not fail during the collision window, the probability of CCA failure for such a node would be $\alpha_1 = (298 - 12)/298 = 286/298$.
- If no node finishes its CSMA wait during the two collision windows, which happens with probability $p_2 = e^{-12(m-1)/E_w} \times e^{-4(m-1)/E_w}$, the active period would sojourn over state sequence *S0-S1-S2-S3* and last for 320 symbols. Given that no node finishes its CSMA wait during the two collision windows ($= 12 + 4 = 16$ symbols), the probability that an active node finishes its CSMA wait during such an active period is $q_2 = 1 - e^{-304/E_w}$ and the probability of CCA failure for such a node would be $\alpha_2 = 1$.
- If no node finishes its CSMA wait during the first collision window but some nodes do so during the second collision window, which happens with probability $p_3 = e^{-12(m-1)/E_w} \times (1 - e^{-4(m-1)/E_w})$, the active period would sojourn over state sequence *S0-S1-S2-S4* and last for 576 symbols. Given that no node finishes its CSMA wait during the first collision window ($= 12$ symbols), the probability that an active node finishes its CSMA wait during such an active period is $q_3 = 1 - e^{-564/E_w}$. Since, the CCA does not take place during the first collision window but may take place during the second collision window, the probability of CCA failure for such a node would be $\alpha_3 = (564 - 4)/564 = 560/564$.

Clearly, an active node may finish its CSMA wait either by triggering a new active period or inside an ongoing active period. However, it is not necessary that all $m-1$ active nodes (excluding the one that triggered an active period) would finish their CSMA waits during an ongoing active period. Suppose q_i is the probability that a node finishes its CSMA wait during an ongoing active period, i.e. $q_i \in \{q_1, q_2, q_3\}$. Then the expected number of nodes that would finish their CSMA waits during an active period (including the node that triggers the active period) is $1+(m-1)q_i$. Therefore, the probability that an active node triggers an active period is $1/(1+(m-1)q_i$ and the probability that an active node finishes its CSMA wait during an ongoing active period is $(m-1)q_i/(1+(m-1)q_i)$. Since the node triggering an active period has zero probability of CCA failure, the overall probability of CCA failure can be expressed as a function of m, the number of active nodes, and E_w, the average CSMA wait duration as follows:

$$\alpha(m, E_w) = \sum_{i=1}^{3} p_i \frac{(m-1)q_i}{1+(m-1)q_i} \alpha_i \tag{2}$$

Note that, for a given value of m, α is a continuous function of E_w, which in turn is a continuous, monotone increasing function of α Eq. (1). This relationship can be exploited to determine the unique value of α for a given m value.

4.3. The probability of collision for a transmission

Out of $m-1$ active nodes still in the middle of their CSMA waits when an active period starts, the probability that i ($0 \leqslant i \leqslant m-1$) nodes finish their CSMA waits during the first collision window and hence transmit during this active period is $p_{\mathrm{coll1}}(i) = \binom{m-1}{i}(1-e^{-12/E_w})^i\,(e^{-12/E_w})^{m-1-i}$. The second collision window comes into picture if no active node finishes its CSMA wait during the first collision window. The probability that i ($0 \leqslant i \leqslant m-1$) nodes finish their CSMA waits during the second collision window and hence transmit during this active period is $p_{\mathrm{coll2}}(i) = \binom{m-1}{i}(1-e^{-4/E_w})^i(e^{-4/E_w})^{m-1-i}$.

An active period would consist of just one packet transmission, the one by the node that triggers the active period, if no other node finishes its CSMA wait during the two collision windows. Similarly, an active period would consist of i transmissions, including one by the node that triggers the active period, in two cases: 1) if $(i-1)$ active nodes wake up during the first collision window; 2) if no active node wakes up during the first collision window and $(i-1)$ active nodes wake up during the second collision window. Thus, the probability that i transmissions take place during an active period, including one by the node that triggers the active period, is given by:

$$p_{\mathrm{trans}}(i) = \begin{cases} p_{\mathrm{coll1}}(0)p_{\mathrm{coll2}}(0) & i=1 \\ p_{\mathrm{coll1}}(i-1) + p_{\mathrm{coll1}}(0)p_{\mathrm{coll2}}(i-1) & 2 \leqslant i \leqslant m \end{cases}$$

Thus, the expected number of transmissions during an active period is $\sum_{i=1}^{m} i \times p_{\mathrm{trans}}(i)$. Clearly, two or more transmissions during an active period imply collision for all these transmissions. Thus, the expected number of transmissions during an active period that witnesses a collision is $\sum_{i=2}^{m} i \times p_{\mathrm{trans}}(i)$. Thus, the fraction of transmissions that experience a collision, or in other words the probability of collision for a transmission, can be described as follows:

$$\beta(m, E_w) = \left(\sum_{i=2}^{m} i \times p_{\mathrm{trans}}(i)\right) \Big/ \left(\sum_{i=1}^{m} i \times p_{\mathrm{trans}}(i)\right) \tag{3}$$

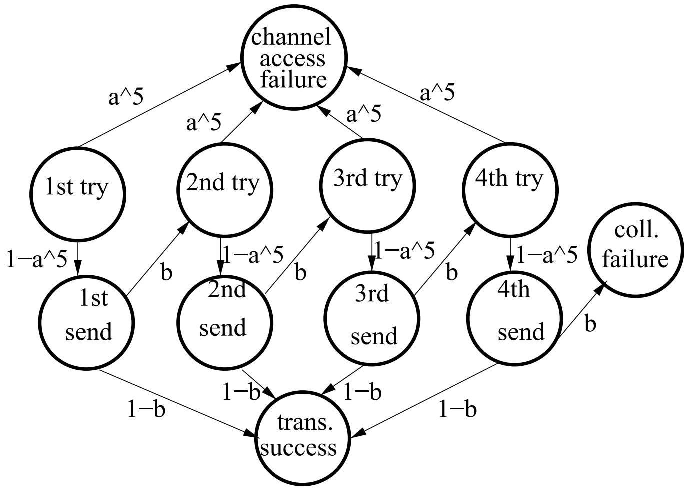


Fig. 4. A state diagram for packet transmission process in IEEE 802.15.4 MAC

As mentioned earlier, the cyclic relationship between α and E_w Eqs (2) and (1) can be used to determine the unique value of α, and hence E_w, for a given m value. Then, we can use Eq. (3) to determine the corresponding β value. Clearly, α and β are functions of m.

4.4. The probability of packet loss in IEEE 802.15.4 MAC

As discussed earlier, the IEEE 802.15.4 MAC layer declares failure in sending a packet if it fails to receive the acknowledgement for the packet even after *1+macMaxFrameRetries* (i.e. 4, by default) transmission attempts. The failure could be declared sooner if, during a transmission attempt, the MAC layer suffers a *channel access failure*, which happens when *1+macMaxCSMABackoffs* (i.e. 5, by default) back-to-back CCA failures take place. Let α be the probability of CCA failure and β be the probability of collision for a packet transmission (or its acknowledgement). Thus, the probability of a *channel access failure* is α^5. Figure 4 shows a state diagram for the packet transmission process followed by IEEE 802.15.4 MAC layer. Clearly, the probability of packet loss, λ, is given by:

$$\begin{aligned}\lambda(\alpha,\beta) = {} & \alpha^5 + (1-\alpha^5)\beta(\\ & \alpha^5 + (1-\alpha^5)\beta(\\ & \alpha^5 + (1-\alpha^5)\beta(\\ & \alpha^5 + (1-\alpha^5)\beta)))\end{aligned} \tag{4}$$

Since α and β are functions of m, the probability of packet loss λ is also a function of m and can be represented as $\lambda(m)$.

4.5. The packet latency

Next, we determine the packet latency as a function of the probabilities $\{\alpha,\beta\}$. The packet latency refers to the time required by IEEE 802.15.4 MAC layer to report back the fate of a packet to the upper layer. As discussed earlier, IEEE 802.15.4 MAC layer performs up to *1+macMaxCSMABackoffs* (5 by default) CCAs to find the channel idle during a transmission attempt. Each CCA is preceded by a CSMA wait. With default values for *macMinBE* and *macMaxBE* (3 and 5 respectively), the average duration of the five CSMA waits are 3.5, 7.5, 15.5, 15.5, 15.5 *backoff periods* ($=$ 20 symbols) respectively. Each CSMA wait is followed by 8 symbols of CCA. Thus, after including the CCA duration, the average durations of five CSMA waits are 78, 158, 318, 318 and 318 symbols respectively. If α is the probability of CCA failure, the probability that a transmission attempt involves exactly i, $1 \leqslant i \leqslant 4$, CSMA waits is $\alpha^{i-1}(1-\alpha)$. The probability that a transmission attempt involves 5 CSMA waits with success in the 5th CCA is $\alpha^4(1-\alpha)$. The probability that a transmission attempt involves 5 CSMA waits, all of which end in CCA failures (i.e. there is a channel access failure), is α^5. Thus, total time spent (in symbols) in CSMA waits and CCAs during a transmission attempt can be described as follows:

$$d_{\text{CSMA}} = \begin{cases} 78 & \text{w.p.} 1-\alpha \\ 78+158 & \text{w.p.} \alpha(1-\alpha) \\ 78+158+318 & \text{w.p.} \alpha^2(1-\alpha) \\ 78+158+2\times 318 & \text{w.p.} \alpha^3(1-\alpha) \\ 78+158+3\times 318 & \text{w.p.} \alpha^4(1-\alpha)+\alpha^5 \end{cases}$$

Clearly, the expected duration of a transmission attempt that ends in a channel access failure is $d_{\text{CAF}} = 78 + 158 + 318 + 318 + 318 = 1190$ symbols (or 19.04ms) and the expected duration in symbols, if the transmission attempt does not end in a channel access failure, is $d_{\text{noCAF}} = \frac{1-\alpha}{1-\alpha^5}(78 + 236\alpha + 554\alpha^2 + 872\alpha^3 + 1190\alpha^4)$.

If a CCA is successful, the node would proceed with *RX-to-TX* turnaround ($d_{\text{TA}} = 12$ symbols) and transmit the packet ($d_T = 266$ symbols for a 133 byte packet) and wait for the acknowledgement. The acknowledgement should be received in $d_A = 34$ symbols (12 symbols *RX-to-TX* turnaround for destination + 22 symbols to transmit 11 byte acknowledgement) unless the packet (or acknowledgement) transmission ends up in collision, in which case the node waits for $d_W = 54$ symbols for acknowledgement and then proceeds with next transmission attempt. Let $d' = d_{\text{noCAF}} + d_{\text{TA}} + d_T + d_A$. Following the state diagram for the packet transmission process (Fig. 4), the packet latency, δ, in symbols can be expressed as a function of α and β as follows:

$$\begin{aligned}\delta(\alpha,\beta) = {} & \alpha^5 d_{\text{CAF}} + (1-\alpha^5)(d' + \beta(d_W - d_A + \\ & \alpha^5 d_{\text{CAF}} + (1-\alpha^5)(d' + \beta(d_W - d_A + \\ & \alpha^5 d_{\text{CAF}} + (1-\alpha^5)(d' + \beta(d_W - d_A + \\ & \alpha^5 d_{\text{CAF}} + (1-\alpha^5)(d' + \beta(d_W - d_A))))))))\end{aligned} \tag{5}$$

Since α and β are functions of m, the packet latency δ is a function of m as well and can be represented as $\delta(m)$.

4.6. *Tying it all together*

Consider n source nodes, in each other's radio range, with the inter-packet generation (or *arrival*) interval at each node being exponentially distributed with rate $1/T$. Thus, the packet arrivals at each source node can be modeled as a Poisson process with rate $1/T$.

Suppose $D(n,T)$ represent the expected packet latency for the source nodes. For a given m value, the corresponding packet latency, $\delta(m)$, can be calculated using Eq. (5). We assume that, for given values of n and T, m is in turn a function of $D(n,T)$ as described below. This relationship between $\delta(m)$ and $D(n,T)$ can be exploited to determine $D(n,T)$.

Suppose node X is in the process of sending a packet. During the service time of this packet, some active nodes would finish sending their packets (and thus become inactive) while some inactive nodes would get a new packet to send (and thus become active). We assume that the probability that m nodes are active is same as the probability that $m-1$ *inactive* nodes get a new packet to send during the time node X is servicing its packet (equal to the expected packet latency $D(n,T)$). Since the number of active nodes at any given time is small and an active node is unlikely to receive another packet while it is servicing one, we calculate the probability that m nodes are active simply as the probability that $m-1$ nodes, out of $n-1$ source nodes, get a new packet to send during time duration $D(n,T)$.

Since the combined packet arrivals across $n-1$ source nodes can be modeled as a Poisson process with rate $(n-1)/T$, the probability that m nodes are active, i.e. $m-1$ nodes get a new packet to send during time interval $D(n,T)$, is given by:

$$p(m) = \frac{(\frac{n-1}{T}D(n,T))^{m-1}e^{-\frac{n-1}{T}D(n,T)}}{(m-1)!} \tag{6}$$

Thus, for given n and T values, the expected packet latency, $D(n,T)$, can be obtained by solving the following recursive equation:

$$D(n,T) = \sum_{m=1}^{n} \delta(m) \times p(m) \tag{7}$$

Once we have determined the expected packet latency $D(n,T)$ for given n and T values, we can determine the probability, $p(m)$, that m nodes are active at any given time using Eq. (6) and hence the expected values of the probability of CCA failure, $A(n,T)$, the probability of collision for a transmission, $B(n,T)$, and the probability of packet loss, $L(n,T)$, as follows:

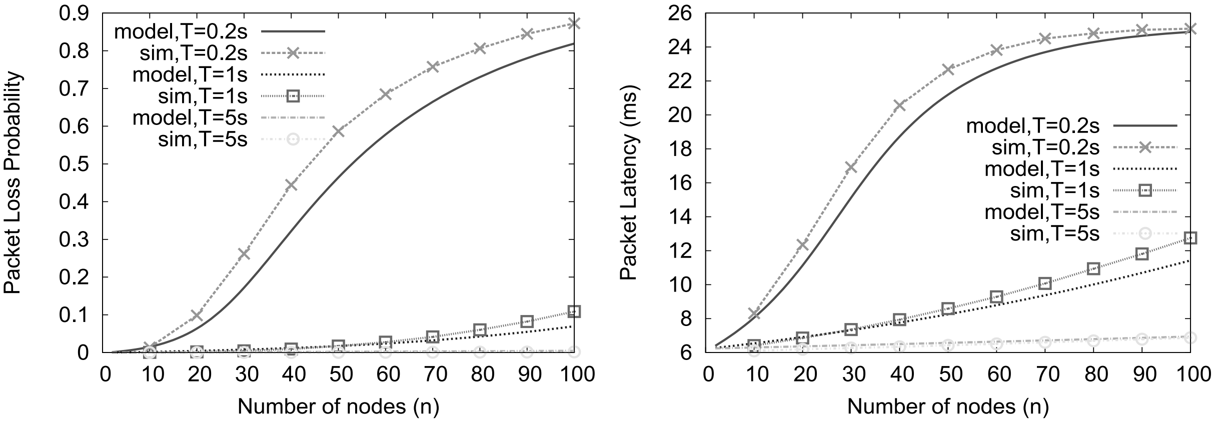
$$A(n,T) = \sum_{m=1}^{n} \alpha(m) \times p(m) \tag{8}$$

$$B(n,T) = \sum_{m=1}^{n} \beta(m) \times p(m) \tag{9}$$

$$L(n,T) = \sum_{m=1}^{n} \lambda(m) \times p(m) \tag{10}$$

4.7. Evaluating the Model's accuracy

To verify the accuracy of the stochastic model developed above, we performed simulations with a significantly improved version [8] of IEEE 802.15.4 protocol implementation in NS2 simulator [11]. IEEE 802.15.4 MAC layer operated in beaconless mode while IEEE 802.15.4 PHY layer operated in 2.4GHz range. All the IEEE 802.15.4 MAC and PHY parameters had their default values as listed in IEEE 802.15.4 specification [1]. Particularly, the *macMinBE, macMaxBE, macMaxFrameRetries* and *macMaxCSMABackoffs* parameters had values 3, 5, 3 and 4 respectively. The CCA duration was 8



(a) (b)

Fig. 5. Comparing the model predictions with NS2 simulation results.

symbols and the packet size used in these simulations was 133 bytes, including 6 bytes of IEEE 802.15.4 PHY header.

Each simulation begins with a certain number of source nodes, $n \in \{10, 20, 30, ..., 100\}$, coming up in a randomly determined sequence within first 100 seconds of the simulation. The source nodes generate packets for a common destination with the time interval between two packet send events at each node being exponentially distributed with average $T \in \{0.2s, 1s, 5s\}$. A successful packet delivery requires the receipt of MAC-level acknowledgement from the destination. All the nodes, including the common destination node, are always in each other's radio range. The simulation time is set such that each node sends close to 10000 packets to the common destination during the simulation. Each node keeps track of the average values of its probability of packet loss (L), i.e. the fraction of packets the MAC layer reports as lost due to channel access or collision failures, and the packet latency (D), i.e. the time difference between the instants when a packet is sent to the MAC layer for transmission and when the MAC layer reports the fate of the packet (successfully delivered or lost) back to the higher layer. The node-level values are then used to calculate the average values of the performance measures, and their confidence intervals, across all the nodes in the network. The 95% confidence intervals were always observed to be within a small range around the average value.

Figure 5 shows the comparison between the average values of the performance measures, calculated across all the nodes, obtained from the simulations and the model predictions. Clearly, there is quite a good match between the model predictions and simulation results for the probability of packet loss (Fig. 5a) and the packet latency (Fig. 5b) for all simulated (n, T) scenarios.

5. Modifications to beaconless IEEE 802.15.4 MAC operation and their evaluation using the stochastic model

It can be observed from Figs 5a and 5b that the packet loss probability and the packet latency increase rapidly as the traffic load in the WSN increases. While the packet latency remains acceptable even for high traffic loads, the probability of packet loss becomes very large even under moderate traffic loads. With 133 byte packet size, a packet transmission takes 300 symbols (266 symbols for packet transmission + 12 symbols for destination's *RX-to-TX* turnaround + 22 symbols for ack transmission) and hence a

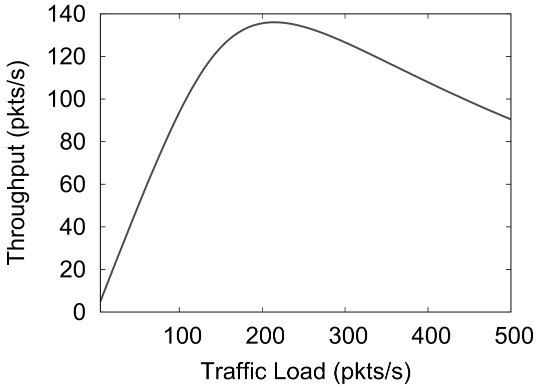


Fig. 6. Throughput versus Traffic Load as predicted by the Model.

maximum packet rate of 208.33 (= 62500/300) packets per second can be supported over a 2.4 GHz channel, assuming all the channel capacity can be utilized. Figure 6 shows how the throughput achieved, as predicted by the model using $n = 100$ and $T = 0.2s$, changes as the traffic load increases. As per Fig. 6, the maximum throughput achieved is approximately 136 packet per second when the offered traffic load is 215 packets per second corresponding to a packet loss probability of 36.72%. Any further increase in traffic load causes almost linear decrease in throughput. Clearly, there is a need to reduce the packet loss probability as the traffic load in the WSN increases. In this section, we investigate the efficacy of various proposals to modify beaconless IEEE 802.15.4 MAC operation in reducing the probability of packet loss while keeping the packet latency under control.

5.1. *Eliminating the second collision window*

In the earlier sections, we observed that if any node finishes its CSMA wait during one of the two possible collision windows during an active period, all the packet transmissions in the active period would end up in a collision. The first collision window is 12 symbols long and is caused by 12 symbol *RX-to-TX* turnaround time. Thus, the first collision window can be reduced by reducing the *RX-to-TX* turnaround time, which is clearly a hardware issue. The second collision window lies across first 4 symbols of 12 symbol *RX-to-TX* turnaround by a destination node as it finishes receiving the packet and prepares to send the acknowledgement back. If a node finishes its CSMA wait during the second collision window, its CCA would succeed because the transmission channel would be idle over the 8 symbol long CCA duration. Any reduction in the *RX-to-TX* turnaround time would also reduce the second collision window as well. Alternatively, we could eliminate the second congestion window by increasing the CCA duration such that it is larger than the *RX-to-TX* turnaround time. In this section, we analyze the impact of increasing the CCA duration from 8 symbols to 16 symbols. For this purpose, we need to modify the calculations of the probability of CCA failure (α), the probability of collision for a transmission (β) and the packet latency (δ) in the stochastic model of IEEE 802.15.4 MAC operation developed earlier.

5.1.1. *The probability of CCA failure*

Figure 7 shows different states in an active period when CCA duration is 16 symbols. State *S0* corresponds to the beginning of the active period as well as that of the first (and only) collision window and has a sojourn time equal to first 12 symbols of 16 symbol long CCA by the node that triggers the active period (say node *A*). If at least one active node finishes its CSMA wait during the collision window,

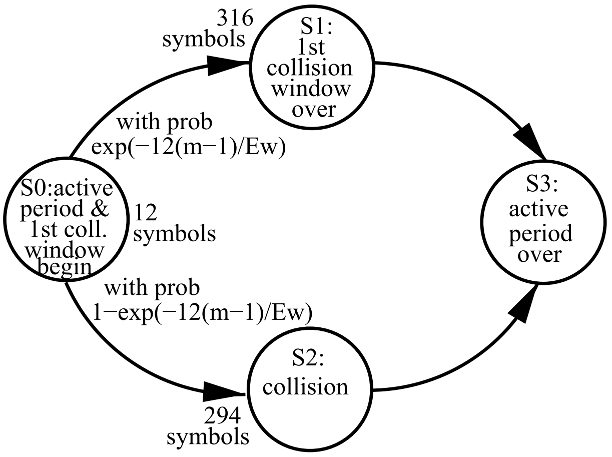


Fig. 7. Different states during the life time of an *active period* when CCA duration is 16 symbols.

which happens with probability $1 - e^{(-12(m-1)/E_w)}$, a collision is guaranteed. In this case, there is a transition to the *collision* state, *S2*. As before, we assume that the last colliding node finishes its CSMA wait just before the end of the collision window. This node would complete its transmission over next 294 symbols (16 symbols of CCA + 12 symbols of turnaround + 266 symbols of packet transmission, assuming 133 byte long packets), which constitute the sojourn time in state *S2*. The completion of sojourn in state *S2* concludes the active period. On the other hand, if no active node finishes its CSMA wait during the collision window, which happens with probability $e^{-12(m-1)/E_w}$, node *A* would complete its CCA (additional 4 symbols), the *RX-to-TX* turnaround (12 symbols), transmit the packet (266 symbols) and receive the acknowledgement (12 symbols of *RX-to-TX* turnaround by destination + 22 symbols for transmission of 11 byte long acknowledgement). These 316 symbols correspond to the sojourn in state *S1*, which also completes the active period.

Thus, different possible durations of an active period as well as other probabilities required to calculate the probability of CCA failure under modified circumstances are as follows:

– If no node finishes its CSMA wait during the collision window, which happens with probability $p_1 = e^{-12(m-1)/E_w}$, the active period would sojourn over states *S0* and *S1* and last for $12 + 316 = 328$ symbols. Given that no node finishes its CSMA wait during the collision window ($= 12$ symbols), the probability that an active node finishes its CSMA wait during such an active period is $q_1 = 1 - e^{-316/E_w}$ and the probability of CCA failure for such a node would be $\alpha_1 = 1$.
– If one or more nodes finish their CSMA waits during the collision window, which happens with probability $p_2 = 1 - e^{-12(m-1)/E_w}$, the active period would sojourn over states *S0* and *S2* and last for $12 + 294 = 306$ symbols. The probability that an active node finishes its CSMA wait during such an active period is $q_2 = 1 - e^{-306/E_w}$. Since, the CCA would not fail during the collision window, the probability of CCA failure for such a node would be $\alpha_2 = 294/306$.

As before, the overall probability of CCA failure can be expressed as a function of m, the number of active nodes, and E_w, the average CSMA wait duration, as follows:

$$\alpha(m, E_w) = \sum_{i=1}^{2} p_i \frac{(m-1)q_i}{1 + (m-1)q_i} \alpha_i \tag{11}$$

As before, for a given value of m, α is a function of E_w, which in turn is a function of α Eq. (1). This cyclic relationship can be exploited to determine the unique value of α for a given m.

5.1.2. The probability of collision for a transmission

Out of $m-1$ active nodes still in the middle of their CSMA waits when an active period starts, the probability that i ($0 \leqslant i \leqslant m-1$) nodes finish their CSMA waits during the collision window is $p_{\text{coll}}(i) = \binom{m-1}{i}(1-e^{-12/E_w})^i(e^{-12/E_w})^{m-1-i}$. Thus, $p_{\text{coll}}(i)$ is the probability that $1+i$ transmissions take place during an active period, including the transmission that triggers the active period. As before, more than one transmissions during an active period imply collisions for all the transmissions and hence the probability of collision β for a transmission is given by Eq. (3) (Section 4.3).

5.1.3. The packet latency

With CCA duration of 16 symbols, the average durations (including the CCA duration) of up to 5 CSMA waits during a transmission attempt would be 86, 166, 326, 326 and 326 symbols respectively. Thus, the expected duration of a transmission attempt that ends in a channel access failure would be $d_{\text{CAF}} = 86+166+3\times 326 = 1230$ symbols and the expected duration in symbols, if the transmission attempt does not end in a channel access failure, would be $d_{\text{noCAF}} = \frac{1-\alpha}{1-\alpha^5}(86 + 252\alpha + 578\alpha^2 + 904\alpha^3 + 1230\alpha^4)$. The rest of the calculations are same as discussed in Section 4.5.

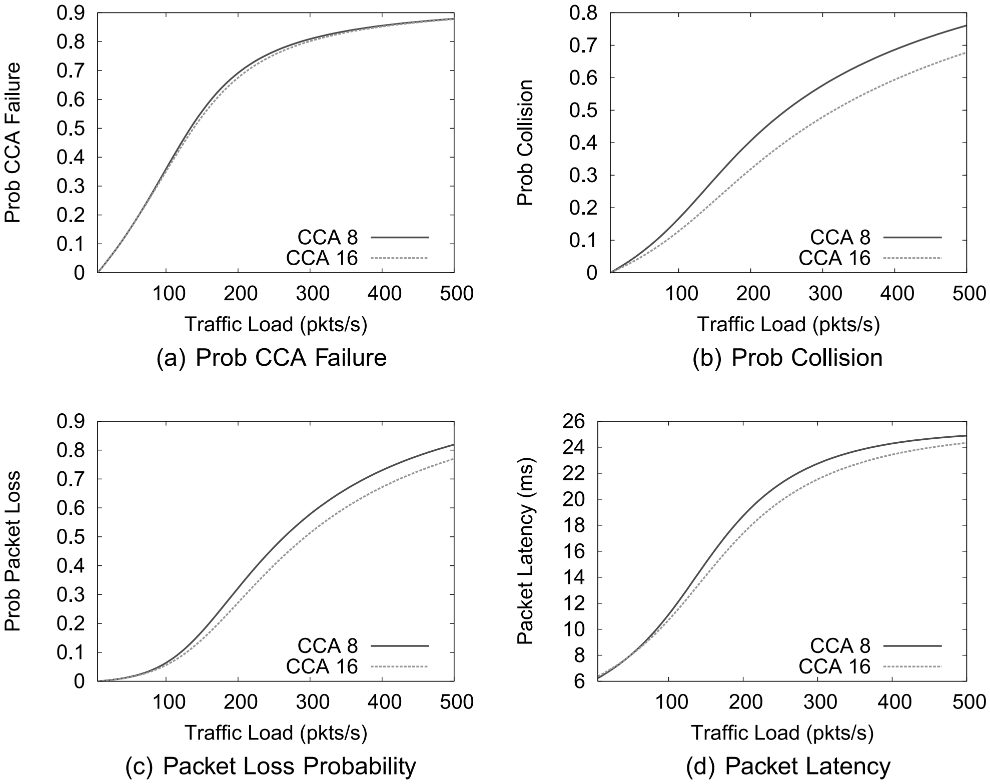
Figure 8 shows the performance under CCA durations 8 symbols and 16 symbols as predicted by the stochastic model (using $n = 100$ and $T = 0.2s$). Clearly, increasing the CCA duration to 16 symbols causes significant decrease in the probability of collision for a transmission while its impact on the probability of CCA failure is negligible. Also, decrease in the probability of collision, resulting from increased CCA duration, translates into slightly lower packet loss probability as well as slightly lower packet latency. A comparison of simulation results with CCA durations 8 symbols and 16 symbols revealed similar trends.

5.2. Understanding the impact of CSMA wait duration on performance

In Section 4.1, we described how the expected CSMA wait duration E_w depends on the probability of CCA failure α and the CSMA algorithm used in beaconless IEEE 802.15.4 MAC operation. Now, we assume that E_w is an independent parameter and analyze its impact on different performance metrics. The objective of this exercise is to see if changing the CSMA wait duration, by either changing the *macMinBE/macMaxBE* values or making the CSMA wait duration independent of α, results in better performance under given traffic load conditions. The following analysis assumes that CCA duration is 16 symbols and α and β are calculated as described in Sections 5.1.1 and 5.1.2 respectively (except that E_w is now an independent parameter and hence Eqs (11) and (3) can directly be used to calculate α and β). The probability of packet loss (λ) still depends on α and β as per Eq. (4) while the packet latency (δ) calculations would change as described next.

5.2.1. The packet latency under independent E_w

IEEE 802.15.4 MAC layer performs up to 5 (by default) CSMA waits to find the channel idle during a transmission attempt. We assume that each CSMA wait duration is exponentially distributed with average E_w symbols. Thus, the expected duration of a transmission attempt that ends in a channel access failure (5 back-to-back CCA failures) is $d_{\text{CAF}} = 5 \times (E_w + 16)$ symbols and the expected duration in symbols, if the transmission attempt does not end in a channel access failure, is $d_{\text{noCAF}} = \frac{(1-\alpha)(E_w+16)}{1-\alpha^5}(1 + 2\alpha + 3\alpha^2 + 4\alpha^3 + 5\alpha^4)$. The rest of the calculations are same as in Section 4.5 and the packet latency (δ) is given by Eq. (5).



(a) Prob CCA Failure

(b) Prob Collision

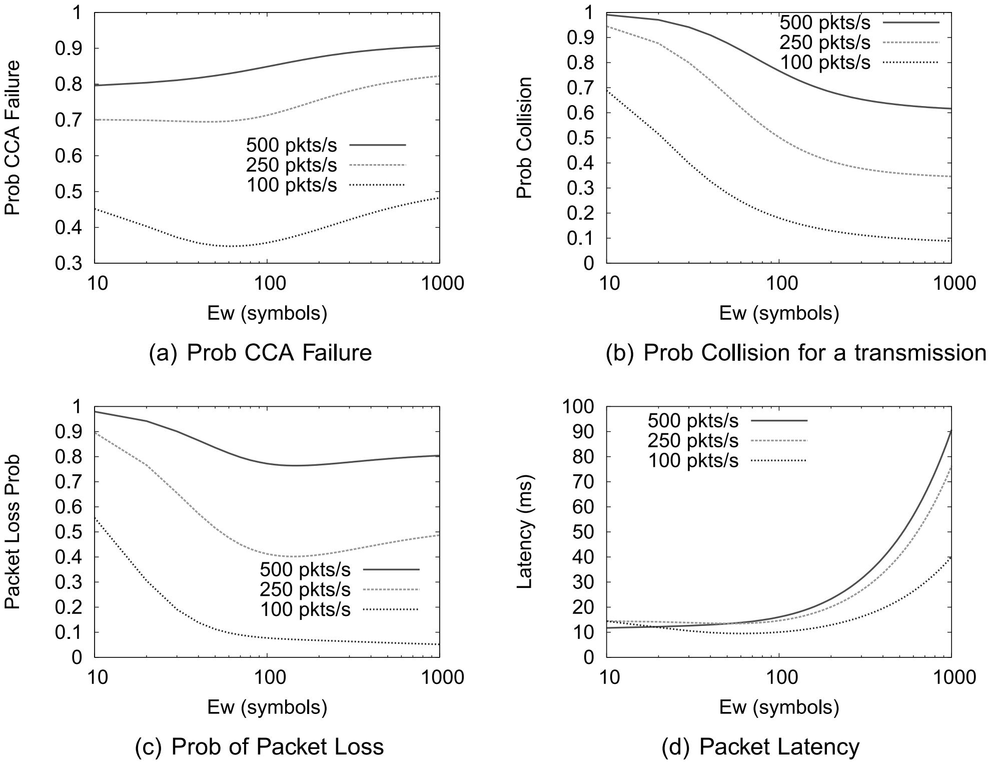
(c) Packet Loss Probability

(d) Packet Latency

Fig. 8. CCA Duration 8 Symbols versus 16 Symbols.

5.2.2. Impact of CSMA wait duration on performance

Figure 9 shows the impact of E_w on different performance metrics calculated using the modified model. Similar trends were observed from the simulation results. The CSMA wait duration has a complex relationship with different performance metrics. The increase in E_w, in general, causes an increase in packet latency. However, as seen in *100 pkts/s* curve in Fig. 9d, the packet latency actually decreases slightly as E_w increases over a certain range. This can be attributed to sharp decrease in the probability of collision, and hence packet retransmissions, as E_w increases over this range. The increase in E_w reduces the probability that a CSMA wait ends during the 12 symbol collision window and hence is expected to reduce the probability of collision for a transmission. However, as seen in Fig. 9b, the probability of collision seems to stabilize as E_w becomes larger and larger. This can be attributed to sharp increase in the number of active nodes as the packet latency increases rapidly with increase in E_w beyond a certain value. The increase in the number of active nodes increases the probability that an active node finishes its CSMA wait inside an active period as well as the probability that an active node finishes its CSMA wait inside the collision window. Thus, increase in the number of active nodes, resulting from increase in E_w, increases both the probability of CCA failure as well as the probability of collision. However, increase in E_w also decreases the probability that an active node finishes its CSMA wait inside an active period and hence has a larger chance to trigger an active period itself, which should decrease the probability of CCA failure. These two factors tend to balance each other, however, as Fig. 9a indicates, the probability of CCA failure increases with increase in E_w beyond a certain value. Finally, Fig. 9c indicates that the impact of increase in E_w on the overall probability of packet loss depends on



(a) Prob CCA Failure

(b) Prob Collision for a transmission

(c) Prob of Packet Loss

(d) Packet Latency

Fig. 9. Impact of E_w on Performance ($n = 100$).

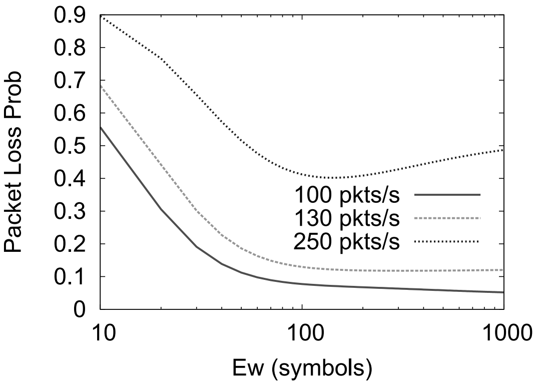


Fig. 10. Increase in E_w causes decrease in packet loss probability for traffic loads less than 130 pkts/s.

the traffic load. Under low and moderate traffic loads (less than 130 packets/s as shown in Fig. 10), increase in E_w decreases the probability of packet loss. Clearly, in this case, the decrease in probability of collision dominates the increase in probability of channel access failure. However, for high traffic loads, the increase in E_w increases the probability of packet loss once E_w crosses a certain value. This implies that the increase in probability of channel access failure dominates the decrease in probability of collision under high traffic loads.

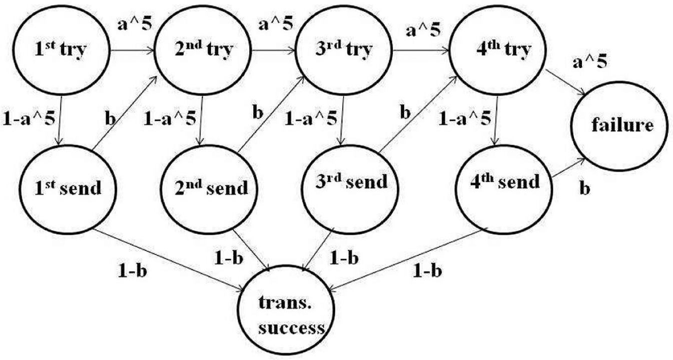


Fig. 11. Modified packet transmission process after eliminating channel access failures.

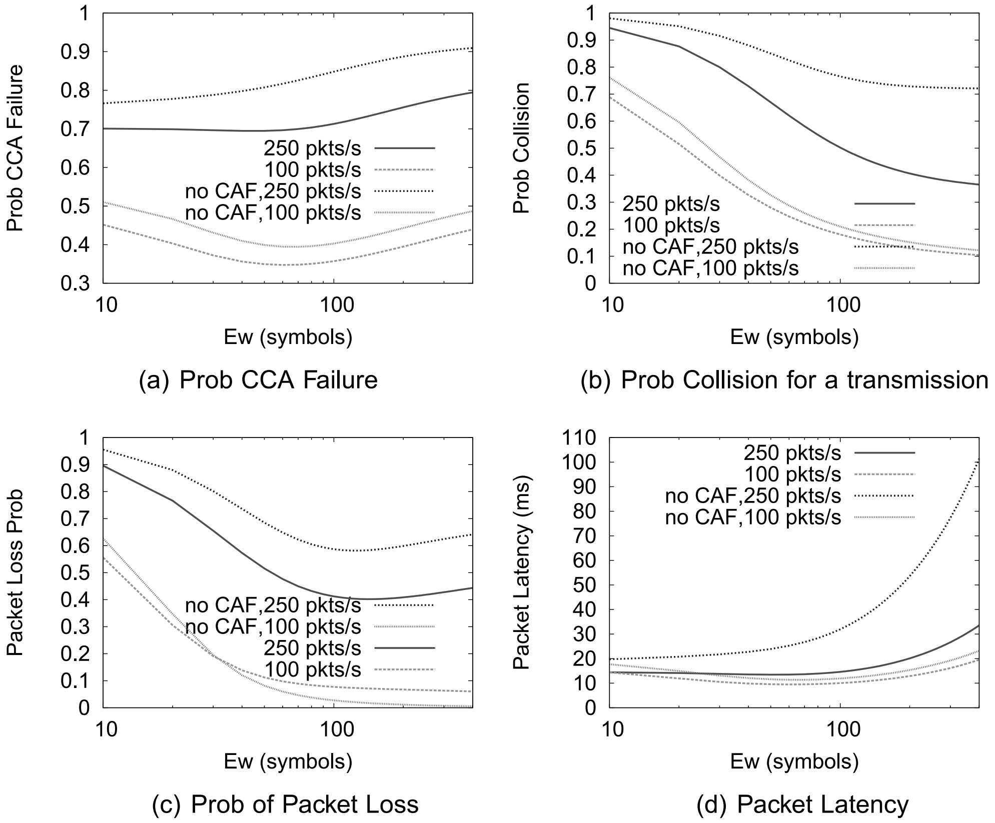
5.3. Eliminating channel access failures

IEEE 802.15.4 MAC layer makes at most *1+macMaxFrameRetries* (by default 4) attempts to transmit the packet and receive the acknowledgement. In a transmission attempt, the MAC layer performs up to *1+macMaxCSMABackoffs* (by default 5) CCAs to find the channel idle. Failure to find the channel idle even after fifth CCA causes a channel access failure and no further attempt is made to transmit the packet. In this section, we analyze the impact of eliminating the channel access failures, i.e., the MAC layer does not abandon the packet after 5 back-to-back CCA failures and simply proceeds with next transmission attempt. Figure 11 shows a state diagram for the modified packet transmission process. Let $p_l = \alpha^5 + (1-\alpha^5)\beta$ be the probability that a transmission attempt fails to send the packet successfully. Clearly, the probability of packet loss, λ, under the modification, is ${p_l}^4$. We still assume that CCA duration is 16 symbols and CSMA waits are exponentially distributed with average E_w, which is a parameter independent of α and CSMA algorithm followed in IEEE 802.15.4 MAC. Thus, α and β are still calculated as described in Sections 5.1.1 and 5.1.2 respectively (except that E_w is an independent parameter).

The expected value of total time spent in CSMA waits and CCAs during a transmission attempt is $d_{\text{CSMA}} = (E_w+16)(1+\alpha+\alpha^2+\alpha^3+\alpha^4)$. A transmission attempt would result in a packet transmission with probability $1-\alpha^5$, which would require $d_{\text{TA}} = 12$ symbols for *RX-to-TX* turnaround and $d_T = 266$ symbols to actually send the packet. The packet transmission would suffer a collision with probability β, in which case the sending node waits for $d_W = 54$ symbols for acknowledgement and then proceeds with next transmission attempt. Otherwise, the sending node would wait for the acknowledgement, which should be received in $d_A = 34$ symbols (12 symbols *RX-to-TX* turnaround for destination + 22 symbols to transmit 11 byte acknowledgement). Thus, the expected value of total time required to complete a transmission attempt is $d_{\text{trans}} = d_{\text{CSMA}} + (1-\alpha^5)(d_{\text{TA}} + d_T + \beta d_W + (1-\beta)d_A)$. Using p_l, the probability of failure in a transmission attempt, the expected packet latency in symbols can be expressed as $\delta(\alpha,\beta) = d_{\text{trans}}(1 + p_l + {p_l}^2 + {p_l}^3)$.

5.3.1. Impact on performance

Figure 12 shows the impact of eliminating channel access failure on different performance metrics as predicted by the model (using $n = 100$). Clearly, eliminating channel access failures causes both α and β to rise (Figs 12a and 12b), although increase in β is minor under moderate traffic load (100 packets/s). Eliminating channel access failures causes packet latency and hence the number of active



(a) Prob CCA Failure

(b) Prob Collision for a transmission

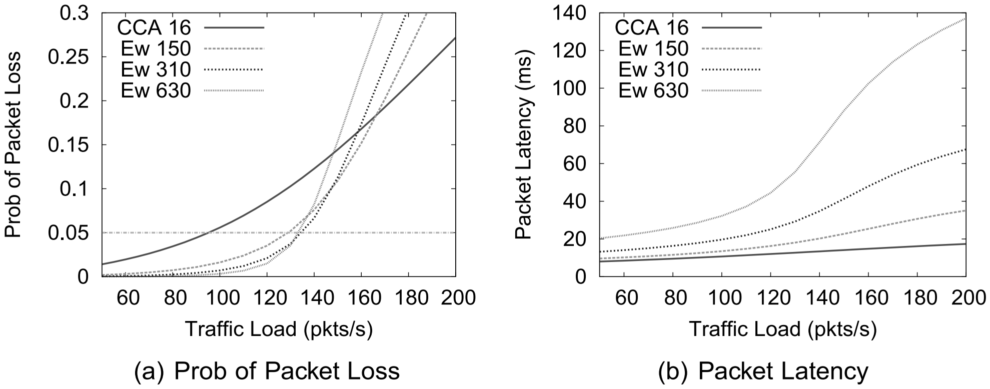
(c) Prob of Packet Loss

(d) Packet Latency

Fig. 12. Performance after eliminating channel access failures.

nodes to increase, which explains the increase in α and β. Minor increase in β value under moderate traffic loads can be explained as resulting from the corresponding minor increase in packet latency. Under high traffic load (250 packets/s), the increase in α and β more than offsets any beneficial impact of eliminating channel access failures and thus probability of packet loss is higher than before. However, under low and moderate traffic loads, eliminating channel access failures results in significant decrease in the packet loss probability for sufficiently high E_w values. Also, notice that eliminating channel access failures does not alter the impact of increasing E_w on the overall packet loss probability. As before, under high traffic loads (more than the observed threshold of *130 packets/s*), the packet loss probability increases with increase in E_w beyond a certain threshold and for lower traffic loads, the increase in E_w causes decrease in the packet loss probability. Note that eliminating channel access failures allows the packet loss rate under *100 packets/s* traffic load to decrease from about 6% to less than 1% (Fig. 12c) as E_w increases while causing only a minor increase in the packet latency (Fig. 12d).

Figure 13 shows the packet loss probability and packet latency values for E_w values 150, 310 and 630 symbols, corresponding to *macMinBE* $=$ *macMaxBE* $=$ *4*, *5* and *6* respectively, after eliminating the channel access failures and compares them against values with standard IEEE 802.15.4 MAC behavior with CCA duration 16 symbols. If we assume 5% to be the maximum acceptable packet loss probability then eliminating channel access failure with $E_w = 310$ symbols (corresponding to *macMinBE* $=$ *macMaxBE* $=$ *5* increases the upper limit on traffic load on WPAN to 134 packets/s (with packet latency about 31.5ms) from 95 packets/s with standard IEEE 802.15.4 MAC behavior with *macMinBE* $=$ *3*,



(a) Prob of Packet Loss

(b) Packet Latency

Fig. 13. Impact of E_w on Performance after eliminating channel access failures.

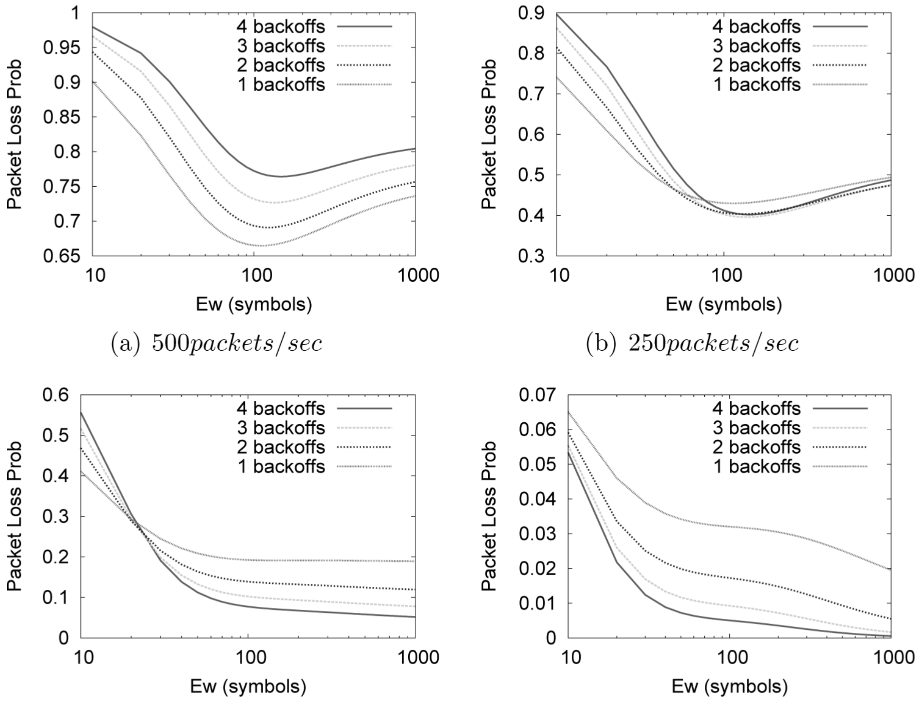
macMaxBE $=$ *5* and CCA duration 16 symbols.

5.4. Reducing macMaxCSMABackoffs

In the previous section, we observed that eliminating channel access failures causes packet latency to increase, which in turn increases the number of active nodes and hence α and β. For high traffic loads, the increase in α and β values offsets beneficial impact of eliminating the channel access failures and the packet loss probability is higher than before. In this section, we examine the impact of reducing the packet latency by reducing the *macMaxCSMABackoffs* value. Reducing *macMaxCSMABackoffs* below its default value (4) would cause a channel access failure to be declared and packet to be abandoned for a smaller (than default 5) number of back-to-back CCA failures, which should tend to increase the packet loss probability. However, reduction in the packet latency, resulting from reduction in *macMaxCSMABackoffs* value, would reduce the number of active nodes and hence α and β values. Reduced α and β values would tend to reduce the packet loss probability. The stochastic model presented in Section 4 can be trivially modified to account for changes in the *macMaxCSMABackoffs* value.

Figures 14 and 15 show the impact of reducing *macMaxCSMABackoffs* on the probability of packet loss and the packet latency for different traffic loads as predicted by the model (using $n = 100$). As expected, reducing the *macMaxCSMABackoffs* value reduces the packet latency, however the impact on the probability of packet loss depends on the traffic load. For low traffic loads (*16.66, 100 packets/s*), the overall impact of reducing the *macMaxCSMABackoffs* is to increase the packet loss probability. However, for very high traffic loads (*500 packets/s*), there is a significant decrease in packet loss proobability as the *macMaxCSMABackoffs* value reduces from 4 to 1. For a traffic load of *250 packets/s*, reducing the *macMaxCSMABackoffs* value seems to help only for small E_w values.

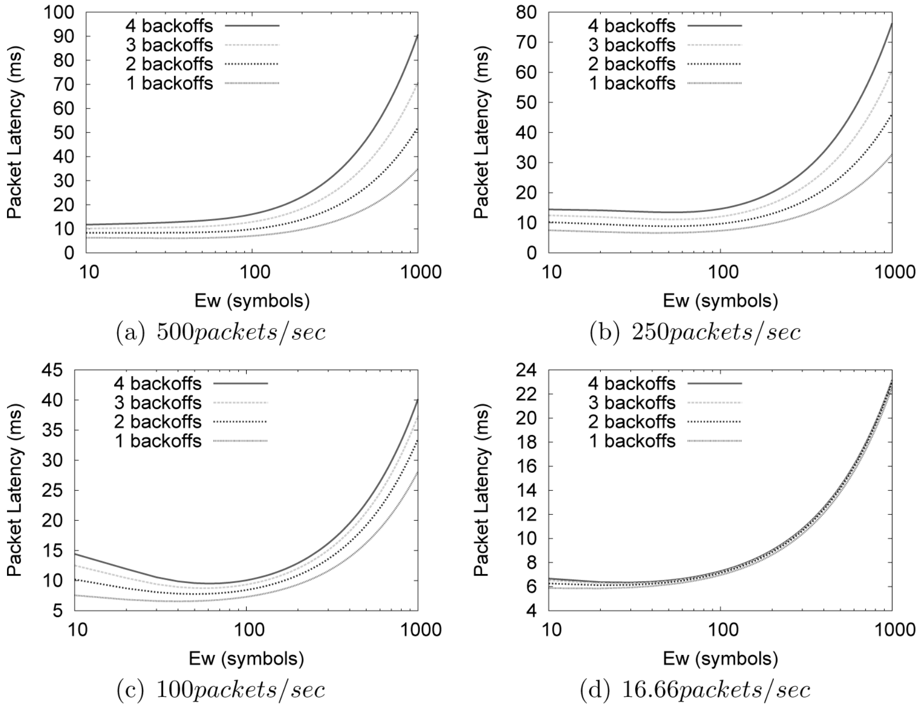
We experimented with reducing packet latency by reducing the *macMaxFrameRetries* value as well. However, resulting increase in collision failures undid any beneficial impact of reduced latency for all traffic loads. We also experimented with dropping a certain percentage of packets at MAC layer without any attempt to send so as to reduce the loss probability for other packets. However, this approach did not prove as effective in reducing the packet loss probability for very high traffic loads as reducing the *macMaxCSMABackoffs* value.



(a) $500 packets/sec$ (b) $250 packets/sec$

(c) $100 packets/sec$ (d) $16.66 packets/sec$

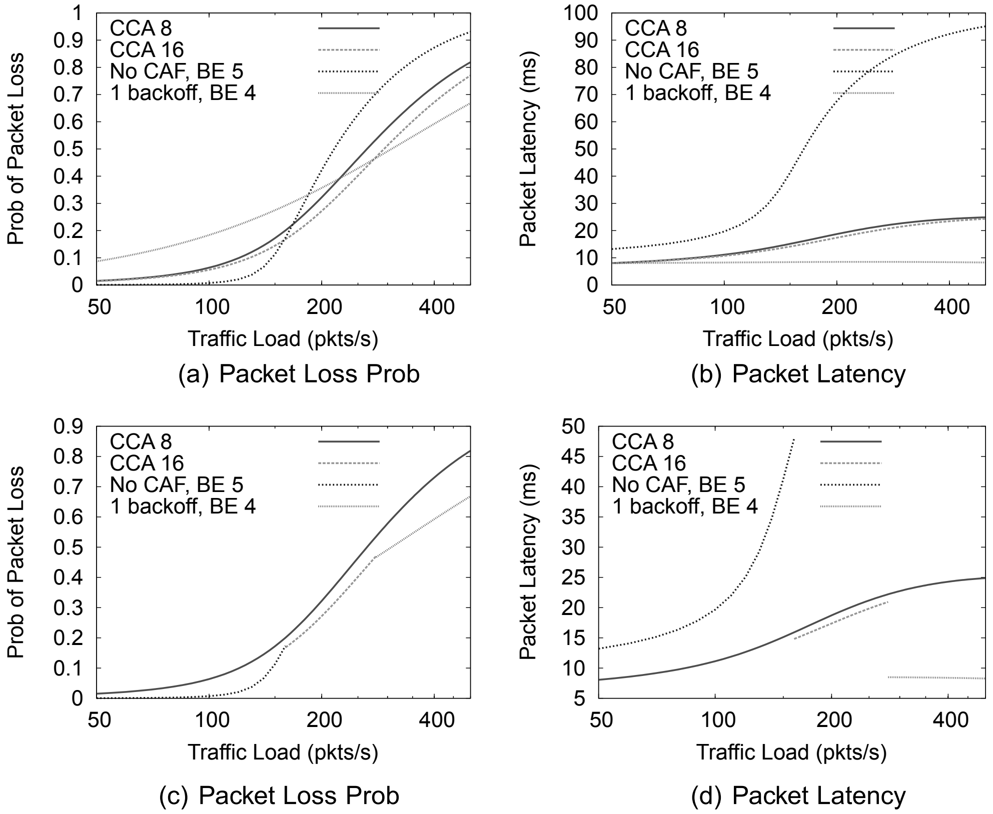
Fig. 14. Change in Packet Loss Probability with change in *macMaxCSMABackoffs*.



(a) $500 packets/sec$ (b) $250 packets/sec$

(c) $100 packets/sec$ (d) $16.66 packets/sec$

Fig. 15. Change in Packet Latency with change in *macMaxCSMABackoffs*.



(a) Packet Loss Prob

(b) Packet Latency

(c) Packet Loss Prob

(d) Packet Latency

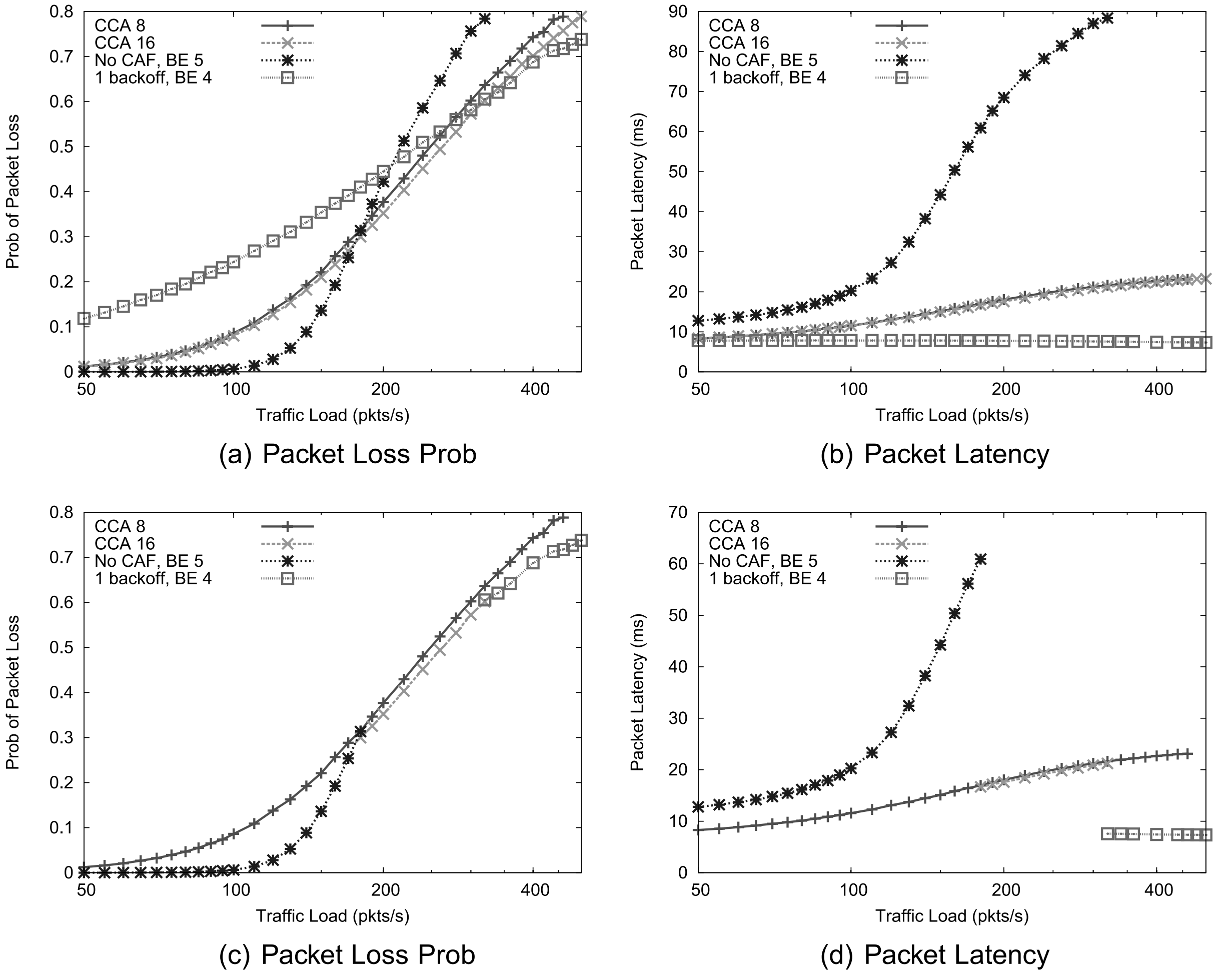
Fig. 16. Performance under Recommended Policies: Model Predictions.

5.5. Recommendations

The observations in Sections 5.1 through 5.4 imply that, in order to reduce the packet loss probability in beaconless IEEE 802.15.4 networks, different policies may have to be used under different traffic loads:

- Under low and moderate traffic loads, we suggest eliminating channel access failures and setting both *macMinBE* and *macMaxBE* to 5 (corresponding to $E_w = 310$ symbols).
- Under very high traffic loads, we suggest reducing *macMaxCSMABackoffs* to 1 and setting both *macMinBE* and *macMaxBE* to 4 (corresponding to $E_w = 150$ symbols).
- Under all traffic loads, we recommend increasing CCA duration from default 8 symbols to 16 symbols.

Figures 16 and 17 compare the performance under the recommended policies with the performance under standard IEEE 802.15.4 MAC operation with default parameter values, as predicted by the stochastic model and as observed in NS2 simulations respectively. Figures 16c and 17c show the traffic load range during which each policy should be used and the packet loss probability values under each policy during its applicable traffic load range. Figures 16d and 17d show the corresponding packet latency values. These figures also compare the performance under desirable policy for a certain traffic load with the performance under standard IEEE 802.15.4 MAC operation with default parameter values. Clearly,



(a) Packet Loss Prob

(b) Packet Latency

(c) Packet Loss Prob

(d) Packet Latency

Fig. 17. Performance under Recommended Policies: Simulation Results.

we can achieve significant drop in the packet loss probability by using the appropriate policy for a given traffic load. For many WSN applications, the traffic load on the WSN may not be predictable and may vary with time over a large range, e.g. a WSN may observe sporadic intervals of very high traffic loads with moderate to low traffic for rest of the time. In such cases, we suggest that a node choose a policy based on the packet loss rate it has observed in recent past, which should be a good indicator of the current traffic load on the WSN [14]. Note that our analysis has assumed that no packet is lost due to signal attenuation/corruption at the physical layer. In situations where significant packet loss takes place due to signal attenuation/corruption at the physical layer, it may be necessary to apply a correction factor to the observed loss rate (to obtain the packet loss rate due to MAC layer effects) before choosing the appropriate MAC policy.

6. Conclusion

IEEE 802.15.4 enabled wireless sensor networks are finding increasing acceptance in various monitoring/control applications. In future, such networks may easily consist of several thousand nodes with each node possibly competing with hundreds of other nodes for access to the transmission channel, a situation that can severly impact the performance. In this paper, we presented a number of modifications to the beaconless IEEE 802.15.4 MAC operation with an objective of improving its performance. We evaluated the impact of these modifications via stochastic modeling and simulations. Our main observation was that the impact of these modifications is strongly dependent on the prevalent traffic load on the network and hence these modifications may be incorporated in IEEE 802.15.4 MAC operation as optional features that can be deployed as per the recommendations made in this paper.

References

[1] Part 15.4: Wireless MAC and PHY layer specifications for low-rate wireless personal area networks. IEEE Std 802.15.4–2006, 2006.

[2] A. Brandt, J. Buron and G. Porcu, Home Automation Routing Requirements in Low-Power and Lossy Networks. Request For Comments 5826, IETF, April 2010.

[3] M. Dohler, T. Watteyne, T. Winter and D. Barthel, Routing Requirements for Urban Low-Power and Lossy Networks. Request For Comments 5548, IETF, May 2009.

[4] A. Durresi and M. Denko. Advances in wireless networks, *Mobile Information Systems* **5**(2) (2009), 101–103.

[5] A. Durresi, P. Zhang, M. Durresi and L. Barolli, Architecture for mobile heterogeneous multi domain networks, *Mobile Information Systems* **6**(1) (2010), 49–63.

[6] J.Y. Goh and D. Taniar, Mobile data mining by location dependencies, In IDEAL, pages 225–231, 2004.

[7] J. Goh and D. Taniar, Mining frequency pattern from mobile users. In KES, pages 795–801, 2004.

[8] M. Goyal, Zigbee/IEEE 802.15.4 module for NS2 simulator, 2008.

[9] M. Goyal, D. Rohm, H. Hosseini, K. Trivedi, A. Divjak and Y. Bashir, A stochastic model for beaconless IEEE 802.15.4 MAC operation. In IEEE SPECTS 2009: Proceedings of the IEEE International Symposium on Performance Evaluation of Computer and Telecommunication Systems, July 2009.

[10] J. Martocci, P. DeMil, N. Riou and W. Vermeylen. Building Automation Routing Requirements in Low-Power and Lossy Networks, *Request For Comments* 5867, IETF, June 2010.

[11] S. McCanne and S. Floyd. ns network simulator.

[12] V. Misic and J. Misic. Improving sensing accuracy in cognitive PANs through modulation of sending probability, *Mobile Information Systems* **5**(2) (2009), 177–193.

[13] K. Pister, P. Thubert, S. Dwars and T. Phinney, Industrial Routing Requirements in Low-Power and Lossy Networks. Request For Comments 5673, IETF, October 2009.

[14] D. Rohm, M. Goyal, H. Hosseini, A. Divjak and Y. Bashir, A simulation based analysis of the impact of IEEE 802.15.4MAC parameters on the performance under different traffic loads, *Mobile Information Systems* **5**(1) (2009), 81–99.

[15] D. Taniar and J. Goh, On mining movement pattern from mobile users, *IJDSN* **3**(1) (2007), 69–86.

[16] A.B. Waluyo, R. Hsieh, D. Taniar, J. Rahayu and B. Srinivasan, Utilising push and pull mechanism in wireless e-health environment. In EEE, pages 271–274, 2004.

[17] A.B. Waluyo, B. Srinivasan and D. Taniar, Optimal broadcast channel for data dissemination in mobile database environment, In APPT, pages 655–664, 2003.

[18] A.B. Waluyo, B. Srinivasan and D. Taniar, A taxonomy of broadcast indexing schemes for multi channel data dissemination in mobile database. In AINA, pages 213–218, 2004.

[19] A.B. Waluyo, B. Srinivasan and D. Taniar, Research in mobile database query optimization and processing, *Mobile Information Systems* **1**(4) (2005), 225–252.

[20] A. Wheeler, Commercial applications of wireless sensor networks using zigbee, *IEEE Communications Magazine* **45**(4) (April 2007), 70–77.

[21] K. Xuan, G. Zhao, D. Taniar and B. Srinivasan. Continuous range search query processing in mobile navigation, In ICPADS, pages 361¤C368, 2008.

[22] Ilsun You and Takahiro Hara. Mobile and wireless networks, *Mobile Information Systems* **6**(1) (2010), 1–3.

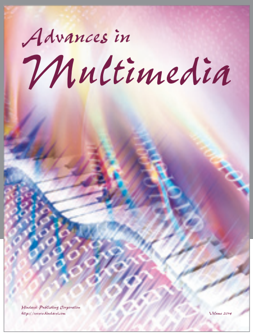
[23] G. Zhao, K. Xuan, D. Taniar and B. Srinivasan, Incremental k-nearest-neighbor search on road networks, *Journal of Interconnection Networks* **9**(4) (2008), 455–470.

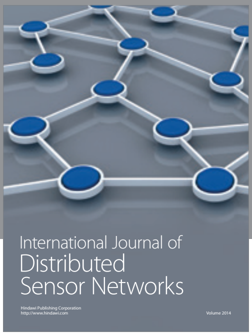
[24] Z. Alliance, Zigbee specification, December 2006.

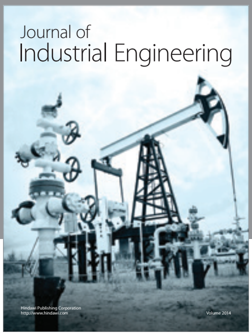
Mukul Goyal is an Assistant Professor in Computer Science department at University of Wisconsin Milwaukee. He received a PhD in Computer and Information Science from Ohio State University in 2003. His research Interests lie in the field of computer networks and performance evaluation.

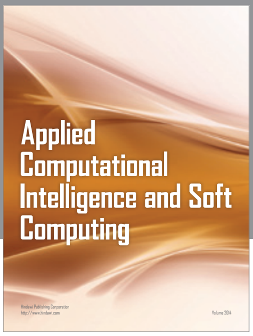
Weigao Xie is a PhD candidate in Computer Science department at University of Wisconsin Milwaukee. Her research interests like in the field of Wireless Sensor Networks and Internet routing protocols such as OSPF.

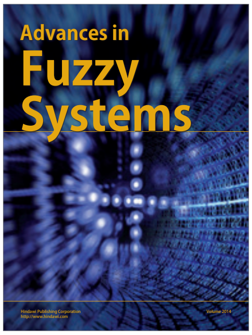
Hossein Hosseini is a Professor in Computer Science department at University of Wisconsin Milwaukee. He received a PhD from University of Iowa in 1982. His research interests lie in the fields of Computer Networks, High Performance Distributed and Parallel Systems, Computer Architecture, Performance Evaluation, Fault Tolerance, Reliability, Operating Systems, Real Time Systems.











Hindawi

Submit your manuscripts at
http://www.hindawi.com