Transmission Early-Stopping Scheme for Anti-Jamming Over Delay-Sensitive IoT Applications

Rami D. Halloush, *Member, IEEE*

***Abstract*—Time-critical, wireless Internet of Things applications have been drawing increasing attention lately. The most characterizing feature of such applications is that a packet has to be delivered within a certain deadline. Not meeting the deadline could result in severe consequences. Jamming attacks utilize the shared nature of the wireless medium to corrupt transmitted packets. Although a corrupted packet could be retransmitted several times until successful delivery, this would add latency and hence could lead to missing the deadline. In this paper, we propose a jamming detection scheme that uses the packet transmission time as a statistic to make detection decisions. The key insight behind our proposed scheme is that, a long transmission/retransmission time for a certain packet indicates an abnormal condition, such as jamming. Therefore, we devise an optimal transmission-time threshold that, when exceeded, a jammer is detected. Unlike most existing detection schemes where, in case of a detection error, retransmission could continue until the deadline is reached, our scheme aims to detect the jammer earlier than the deadline so that the remaining time (until the deadline) could be utilized in retransmitting the packet over a safe channel. The proposed detection scheme is a general framework that can be applied to many situations. After conducting a thorough analysis, we apply the proposed early-stop jamming detection framework to the distributed coordinated function medium access mechanism specified by the 802.11 standard. Our simulation results show significant performance gains achieved by the proposed scheme.**

***Index Terms*—802.11, carrier sense multiple access with collision avoidance (CSMA/CA), distributed coordinated function (DCF), jamming, optimal detection, time-critical Internet of Things (IoT) applications.**

I. Introduction

INTERNET of Things (IoT) [1]–[3] is a networking infrastructure that connects a large number of devices that are equipped with sensors, actuators and microcontrollers. These devices are distributed across heterogeneous networks that are interconnected through the Internet. Due to the valuable benefits they provide to human life, IoT applications are becoming more and more prevalent. In fact, the number of IoT devices is expected to reach 41 billion by 2020 [3]. Common IoT applications include smart grids, smart cities, smart homes, smart manufacturing, smart agriculture [1], [2], [4], [5], etc.

It is quite often that IoT applications are associated with critical human tasks such as health care and vehicle driving. Therefore, having a high security level is a crucial requirement for such IoT applications. Unfortunately, due to certain characteristics in the IoT infrastructure, achieving security is more difficult compared to traditional networks [3], [6]. Such IoT characteristics include limited battery capacity, and limited computational power. Due to these characteristics, complex security schemes would not be affordable.

Security aspects could affect other important IoT requirements, such as timely message delivery. In many IoT applications, it is required to deliver messages between devices in a timely manner. The traffic in such applications is time-critical, and there is always a message delivery deadline beyond which the message is considered to be useless [7]. For example, in smart grid applications, messages in power substations have a message delivery deadline that ranges from 3 to 500 ms [8]. Since message delivery in time-critical IoT applications triggers a chain of responses at the receiver side, failing to meet the delivery-deadline will result in a sequence of failures in the system, and this could have severe consequences, especially in IoT applications that deal with human health, homeland security, money, etc. Consequently, it is crucial to guarantee timely message delivery, and this is considered one of the most challenging aspects in IoT applications and in cyber-physical systems in general.

A main challenge in meeting the delivery deadline is *network unavailability* [9]–[13], where a network resource is temporarily unavailable, which introduces latency and eventually leads to missing the deadline. Network unavailability could be a result of malicious actions [3], [14]. Jamming [15]–[18] is a main security attack that leads to wireless network unavailability. In a jamming attack, an adversary broadcasts radio messages into the wireless network without following the medium access control (MAC) rules, and this causes interference with legitimate packets. Although failed packets could be retransmitted until successful delivery, retransmissions introduce latencies and this could lead to missing the deadline in a time-critical IoT application. Therefore, when transmitting a packet, it is necessary to have a jamming detection technique that is capable of detecting the jammer as early as possible; in order to have enough time to handle the jamming attack before the deadline is reached.

Manuscript received August 18, 2018; revised March 4, 2019 and March 26, 2019; accepted April 13, 2019. Date of publication April 16, 2019; date of current version October 8, 2019.

The author is with the Department of Telecommunication Engineering, Yarmouk University, Irbid 21163, Jordan (e-mail: rami.h@yu.edu.jo).

Digital Object Identifier 10.1109/JIOT.2019.2911683

A. Motivation

Although there have been many research works proposing jamming detection techniques [19]–[25], little attention has been focuses on detecting jamming attacks in time-critical applications [8], [26]. Further, it is remarkable that, in many of the proposed schemes, the detector makes a decision prior transmission. In case the detector decides that there is no active jammer, the transmitter starts transmission. This forces us to ask the following questions.

1) What if the detector incorrectly decides that there is no jammer, while in fact, there is an active jammer?
2) If the detector correctly decides that there is no jamming, it will start transmission. What if a jammer appears during transmission?

In both cases, and due to the presence of a jammer, the transmitter will go through repeated retransmissions until, most likely, the deadline is exceeded. In this case, the time from the beginning of packet transmission until the deadline is reached is wasted; as it eventually led to a failed packet. In this paper, we argue that if the transmitter can, at a point of time, later than the start of transmission and earlier than the deadline, detect that the current packet is doomed to failure, then it could initiate an *early-stop*; i.e., give up transmission over the current channel before the deadline is reached. It could use the remaining time (until the deadline) to handle the jammer, through, for example, switching to a safe channel where transmission could end up in a success. Apparently, this would save time significantly. In this scheme, detection is not only at the beginning of transmission, but also during transmission. *Prior-transmission detection* is to decide whether to start transmission over a certain channel or not. *During-transmission detection* is to decide whether to continue transmitting over the channel or to initiate an early-stop. Unlike most detection schemes in the literature, the proposed early-stop detection scheme follows the during-transmission detection paradigm, which results in significant performance gains.

To build such *early-stop* detector, we argue that a longer transmission time for a single packet indicates an abnormal condition, such as a jamming activity. This is because longer transmission time implies more retransmissions (more failed transmission attempts). Therefore, we propose building a detector based on *packet transmission time*. If the transmission time goes past a certain threshold, a jammer is detected, and transmission shifts to a safe channel. Note that this scheme has a low computational cost, since it merely involves comparing transmission time to a threshold. Hence, it fits IoT applications with stringent nature (e.g., limited battery and computational power). Another cost that is incurred by the proposed early-stop scheme is the channel switch cost (delay). The proposed scheme is carefully designed, taking into consideration this cost. For instance, a switching decision should only be made if the packet transmission time, added to channel switching delay, does not exceed the deadline. As we show later, the extra switching cost is amortized by the attained overall performance gains.

B. Contribution

The main contributions of this paper are as follows.

1) We propose a new anti-jamming algorithm for time-critical wireless IoT applications. Our algorithm employs the packet transmission time as a statistic to make detection decisions at a time point earlier than the deadline. In case a jammer is detected, the transmitter shifts transmission to a safe channel. Unlike other jamming detection schemes in the literature that follow the prior-transmission detection paradigm, where a station decides whether to start transmission over a certain channel or not, our proposed scheme follows the more flexible during-transmission detection paradigm, where a station decides whether to continue transmitting over a channel or to initiate an early-stop.
2) To build the detector, we derive a detailed mathematical model for the transmission time under the existence and absence of a jammer. We use our mathematical model to derive an optimal Bayesian decision rule in a way that minimizes the cost of incorrect decisions.
3) Our algorithm considers both reactive and nonreactive jamming attacks.
4) The early-stop detection scheme is a general framework that can be applied to various situations. We conduct a thorough analysis of the distributed coordinated function (DCF) medium access mechanism specified by the 802.11 standard, and show how it can fit in the proposed detection framework.
5) Through simulation experiments, and comparisons with jamming attack detection based on estimation (JADE), a state-of-the-art jamming detection scheme in the literature, we demonstrate the significant performance gains attained by the early-stop detection scheme.

C. Organization

This paper is organized as follows. In Section II, we survey related work in the literature, and we describe our proposed solution. In Section III, we present a mathematical model for transmission time under the jamming and the no-jamming states. In Section IV, we describe in detail the proposed detector, and we derive the optimal time threshold used for detection. In Section V, we apply the proposed early-stop jamming detection scheme to the DCF medium access mechanism specified by the 802.11a standard. In Section VI, we evaluate the performance of the proposed detector. Finally, in Section VII, we present the conclusion of this paper.

II. Related Work

A variety of detection schemes have been investigated in the literature. Different detection schemes use different statistics for decision making. In [19], the effective channel utilization (ECU) metric is computed and used as a statistic to detect jamming attacks. ECU is a widely used metric that measures the channel utilization in a wireless network. The ECU of a certain channel during a time interval is the fraction of transmission time, or the time during which the channel is sensed to be busy, to the total time interval. The authors proposed a detection algorithm where the ECU is computed and compared to a threshold. In case the ECU is less than the threshold, transmission switches to a new channel. In [20], a monitoring node keeps track of the percentage of transmission collisions in a wireless network. A jamming attack is detected when the percentage exceeds a certain threshold. In that case, a notification message is transmitted out of the jammed region. In [21], a detection scheme is proposed where the cause of a bit error is identified for individual packets by examining the received signal strength during the reception of these packets. The cause could indicate a jamming attack. Xu *et al.* [22] discussed the use of different measurements as a basis for jamming detection. Such measurements include signal strength, packet delivery ratio (PDR), and carrier sensing. The authors argue that using a single measurement is not sufficient to make a reliable detection decision. Therefore, the authors propose an enhanced detection scheme that uses a consistency check to remove decision ambiguity. For instance, they use a PDR measurement to detect jamming attacks, and a signal strength measurement for a consistency check. In [23], the authors use both PDR and signal strength to detect the existence of a jammer. Subsequently, packet send ratio (PSR) and PDR are used to specify the category of the detected jamming attack. Siddhabathula *et al.* [24] focused on increasing the detection speed. They achieved that by using the PDR metric. They argued that many PDR-based detectors use the end-to-end PDR, which requires monitoring the network for a long time. To solve this problem, they suggested a collaborative scheme to evaluate the PDR in a given area instead of evaluating the PDR between two nodes as done by many detectors. Their argument is that the jammer affects a whole area, not just a pair of nodes. Bayraktaroglu *et al.* [25] addressed 802.11 wireless networks. They conducted a theoretical analysis to characterize a network level measure, namely the saturated network throughput under jamming attacks.

Note that the aforementioned jamming detection schemes are designed for conventional wireless applications, not for time-critical applications. Lu *et al.* [8], [26] addressed jamming detection over time-critical wireless applications. They pointed out that the conventional performance metrics used in jamming detection in conventional wireless applications cannot be readily employed in time-critical applications. The reason is that, in conventional wireless applications, the impact of jamming is evaluated at the packet level, such as the case with the PDR, PSR, and percentage of collision metrics [20], [22], [23], or at the network level such as the case with the saturated network throughput metric [25]. However, these packet-level and network-level metrics do not consider time constraints in time-critical applications. For instance, a 100% PDR does not necessarily mean that all packets would be delivered within the time constraints. Therefore, Lu *et al.* [8], [26] proposed a model for jamming attacks over time-critical applications, and they proposed a performance metric that considers a time constraint. They named the metric the message invalidation ratio (MIR). For a certain time constraint, the MIR for a certain packet is the probability that the end-to-end delay of the packet would exceed the time constraint. The authors use the MIR metric to develop JADE. In JADE, before transmitting a packet, the MIR is used to detect a jammer. If the detector decides that there is no jammer, the packet is transmitted. If transmission does not succeed, the packet is transmitted a number of times until success, or until the deadline is reached. As we mentioned earlier, detection prior transmission has many pitfalls; for instance, if due to an incorrect decision, transmission starts while there is a jammer, or, even when the decision is correct, but a jammer appears after the start of transmission. Apparently, transmission will take a long time in these cases, and this indicates the existence of a jammer. Therefore, we propose using the transmission time as a metric for jamming detection. Our detector could be used along with another detector (JADE for instance), or as a stand alone detector.

III. Mathematical Model

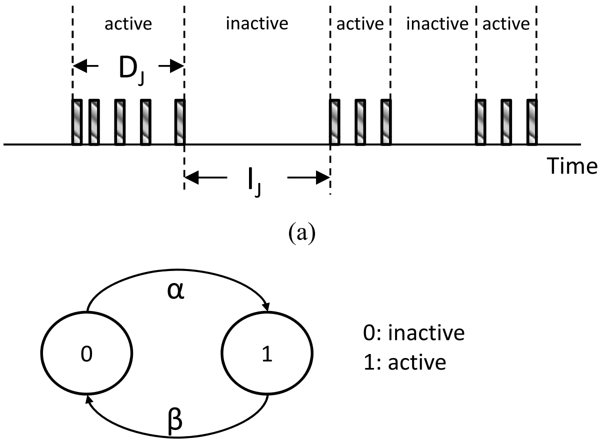
1) Let J denote the event of a jamming attack. Probability of J is π_J. Under jamming conditions, packet errors are due to bad channel conditions, collisions with legitimate packets, and collisions with jamming pulses. We model transmitting a single packet under a jamming attack as a Bernoulli trial with packet success probability p_J, and packet error probability $q_J = 1 - p_J$. Later on, we will derive p_J for both reactive and nonreactive jamming attacks.
2) Let $\overline{J}$ denote the event of no jamming attack. The probability of $\overline{J}$ is $\pi_{\overline{J}} = 1 - \pi_J$. When there is no jamming attack, the only source of packet error is bad channel conditions and collisions with legitimate packets. We model transmitting a single packet under no jamming attack as a Bernoulli trial with packet success probability $p_{\overline{J}}$, and packet error (due to channel conditions and collisions) probability $q_{\overline{J}} = 1 - p_{\overline{J}}$.

Many practical schemes were proposed in the literature to estimate the prior probability of a jamming attack. For example, Wu *et al.* [27] proposed a learning scheme that employs observing the wireless environment and access pattern of stations in the network. It employs history observations to estimate relevant jamming parameters, including $\pi_{\overline{J}}$ and π_J, using maximum likelihood estimation (MLE) techniques. In [8] and [26], the packet failure rate is observed and incorporated in the MIR metric to estimate the likelihood of jamming in a certain wireless environment.

It is worth mentioning that the mathematical model we are developing in this paper does not aim at identifying whether a certain packet error is due to malicious or nonmalicious causes. Instead, it aims at identifying the existence or absence of a jammer through the frequency of packet errors, which is captured by the parameters q_J and $q_{\overline{J}}$.

A. Nonreactive Jammer

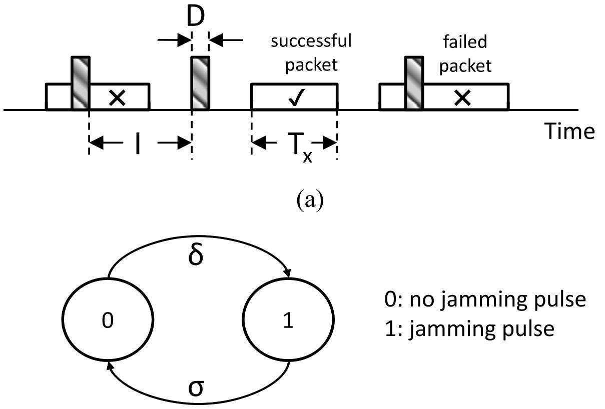
A nonreactive jammer sends interfering pulses without being aware of the activity of the victim IoT device. Fig. 1(a) shows a typical nonreactive jamming activity, where the attacker is active for a random amount of time that we denote by D_J, and then, it is inactive for another random amount of time denoted by I_J. We model the jammer's activity via the two-state Markov chain shown in Fig. 1(b). In state 1, the jammer is active transmitting interfering pulses, whereas in state 0, the jammer is inactive. The jammer leaves state 0



(a)

(b)

Fig. 1. Nonreactive Jamming activity. (a) Activity and inactivity epochs for a nonreactive jammer. (b) Two-state Markov chain model for nonreactive Jammer activity.



(a)

(b)

Fig. 2. Jamming pulses while the nonreactive jammer is active. (a) Packet transmission under a nonreactive jamming attack. (b) Two-state Markov chain model for the interference pulses during a nonreactive jamming attack.

with rate α and leaves state 1 with rate β. This implies that the interarrival times D_J and I_J are exponentially distributed with mean values $(1/\beta)$ and $(1/\alpha)$, respectively. Consequently, it becomes possible to find the probability of the no-jamming event $\overline{J}$, which is the steady-state probability of state 0

$$\pi_{\overline{J}} = \frac{\beta}{\alpha + \beta}. \tag{1}$$

Next, we study packet transmission under an ongoing jamming attack; a situation illustrated in Fig. 2(a). As shown in the figure, we denote the random duration of the jamming pulse by D, and the random pulse-interarrival time by I. The packet transmission time is also random and denoted by T_x. Note that a packet fails when it overlaps with a jamming pulse. We model the sequence of jamming pulses during the jamming attack via the two-state Markov chain shown in Fig. 2(b). In state 1, there is an ongoing jamming pulse, whereas in state 0, there is no jamming pulse and we are in the pulse-interarrival interval (the interval that precedes the next pulse). The jammer leaves state 0 with a rate of δ, and leaves state 1 with a rate of σ. This implies that I and D are exponentially distributed with mean values $(1/\delta)$ and $(1/\sigma)$, respectively.

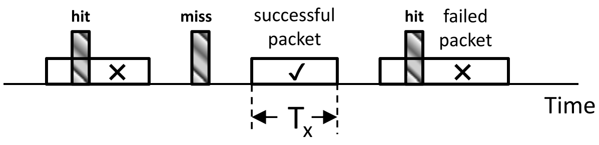


Fig. 3. Packet transmission under a reactive jamming attack. A *hit* occurs with probability p_h, whereas a *miss* occurs with probability $q_h = 1 - p_h$.

Remember that, under a jamming attack, the probability of successful packet transmission is p_J. To derive an expression for p_J for a nonreactive jamming attack, we note the following: when a packet is transmitted during a nonreactive jamming attack, the transmission will be successful if three independent events occur jointly.

1) Transmission starts somewhere in the pulse-interarrival interval [I in Fig. 2(a)]. This guarantees that there is no jamming pulse to interfere with.
2) Transmission completes before the arrival of the next pulse.
3) The packet survives channel-related failures during the whole period of transmission.

We denote the probability of the first event by p_{nj}, and it is simply the steady-state probability of state 0 ("no jamming pulse") in Fig. 2(b), which is given by

$$p_{nj} = \frac{\sigma}{\sigma + \delta}. \tag{2}$$

Probability of the second event, denoted p_{T_x}, is given by

$$\begin{aligned} p_{T_x} &= P[T_x < I] \\ &= P[T_x = \min(T_x, I)] \\ &= \int_0^\infty (1 - F_I(u))\ \ f_{T_x}(u) du \end{aligned} \tag{3}$$

where $F_I(t)$ is the cumulative distribution function of I, and $f_{T_x}(t)$ is the probability density function (pdf) of T_x. Under the exponential assumption of I (rate δ), (3) becomes

$$p_{T_x} = \int_0^\infty e^{-\delta u}\ \ f_{T_x}(u) du. \tag{4}$$

The probability of the third event is $p_{\overline{J}}$; since in the absence of a jamming pulse, the only source of packet failure is the condition of the channel and collisions with legitimate packets. Finally, we compute p_J as

$$p_J = p_{nj} \times p_{T_x} \times p_{\overline{J}}. \tag{5}$$

B. Reactive Jammer

A reactive jammer is aware of the activity of the victim device. It transmits a jamming pulse once it senses activity. Fig. 3 shows a typical reactive jamming attack. Let a *hit* denote the event of detecting and corrupting (via a jamming pulse) a packet sent by a device. A *miss* denotes that a hit failed. There are different strategies for reactive jamming. Each strategy results in a different probability of hit (p_h), and probability of miss ($q_h = 1 - p_h$). Many schemes have been proposed for estimating these parameters based on observations [27]. For

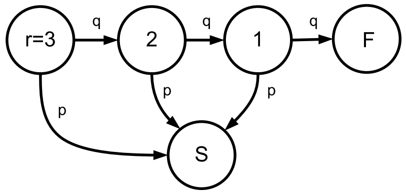


Fig. 4. State-transition diagram of the embedded discrete-time Markov chain model of the transmission/retransmission dynamics when $r = 3$.

our purpose, we assume that an estimation scheme is employed to obtain the values of p_h and π_J.

Now, we derive an expression for p_J (the probability of successful packet transmission) under a reactive jamming attack. We note that, under this kind of attack, a transmission will be successful if two independent events occur jointly.

1) Transmission survives the reactive interference pulse (a miss event occurs).
2) The packet survives channel-related failures during the whole period of transmission.

Therefore, we compute p_J as

$$p_J = q_h \times p_{\bar{J}}. \tag{6}$$

C. IoT Transmitter

In this paper, we consider wireless protocols that handle a packet error through retransmitting the unsuccessful packet M times until a successful transmission is acknowledged, or a predefined retransmission limit r is reached. Note that M is random and r is fixed. Each packet transmission attempt is modeled as a Bernoulli trial with probability of error q and probability of success $p = 1 - q$. Note that $p = p_J$ when there is an attack, and $p = p_{\bar{J}}$ when there is no attack. We model the transmission/retransmission dynamics via a semi-Markov process [9], [10]. The discrete-time Markov chain embedded in our semi-Markov process model is depicted in Fig. 4 (where $r = 3$). Note that there is a total of $r + 2$ states. The states numbered r to 1 represent transmission attempts. The remaining two states are absorbing (i.e., the process ends up eventually in one of them). State F represents the event of packet transmission failure occurring when the r attempts are exhausted with no success. On the other hand, state S represents the event of successful packet transmission. As the figure shows, the process starts at state r where a transmission attempt is made. With probability p, the attempt succeeds and state S is reached, and with probability $q = 1 - p$, the attempt fails and a transition to state $r - 1$ occurs, where a new retransmission is attempted. Note that transmission occurs only in states numbered r to 1. No transmission happens in states F and S. These states are used to represent the termination of a packet transmission either successfully or unsuccessfully.

Let X_i^s and X_i^f be two random variables (RVs) that represent the transmission time of a single transmission attempt succeeding or failing at state i, respectively. Let T be an RV that represents the total transmission time of a packet (aggregate time of all attempts until success or failure). In other words, T is the amount of time until getting absorbed either in state S or in state F in Fig. 4. It can be shown that the Laplace–Stieltjes (LS) transform of T is [28]

$$L_T(s) = \left\{ \sum_{i=1}^{r} q^{i-1} p \left(\prod_{j=1}^{i-1} L_{X_j^f}(s) \right) L_{X_i^s}(s) \right\} + q^r \prod_{j=1}^{r} L_{X_j^f}(s) \tag{7}$$

where $L_{X_i^s}(s)$ and $L_{X_i^f}(s)$ are the LS transform of X_i^s and X_i^f, respectively. The mean value of T is given by

$$\begin{aligned} E[T] &= -\frac{d}{ds} L_T(s)|_{s=0} \\ &= \left\{ \sum_{i=1}^{r} q^{i-1} p \left(m_{X_i^s} + \sum_{j=1}^{i-1} m_{X_j^f} \right) \right\} + q^r \sum_{j=1}^{r} m_{X_j^f} \end{aligned} \tag{8}$$

where $m_{X_i^s}$ and $m_{X_j^f}$ are the mean values of X_i^s and X_i^f, respectively. The second moment is given by

$$\begin{aligned} E[T^2] &= \frac{d^2}{ds^2} L_T(s)|_{s=0} \\ &= \left\{ \sum_{i=1}^{r} q^{i-1} p \left(\sum_{\substack{j=1 \\ j \neq k}}^{i-1} \sum_{k=1}^{i-1} m_{X_j^f} m_{X_k^f} + 2 m_{X_i^s} \sum_{j=1}^{i-1} m_{X_j^f} \right. \right. \\ &\qquad \left. \left. + \; E[X_i^{s^2}] + \sum_{j=1}^{i-1} E[X_j^{f^2}] \right) \right\} \\ &\quad + q^r \left(\sum_{\substack{j=1 \\ j \neq k}}^{r} \sum_{k=1}^{r} m_{X_j^f} m_{X_k^f} + \sum_{j=1}^{r} E[X_j^{f^2}] \right). \end{aligned} \tag{9}$$

Hence, the variance is

$$\sigma_T^2 = E[T^2] - E[T]^2. \tag{10}$$

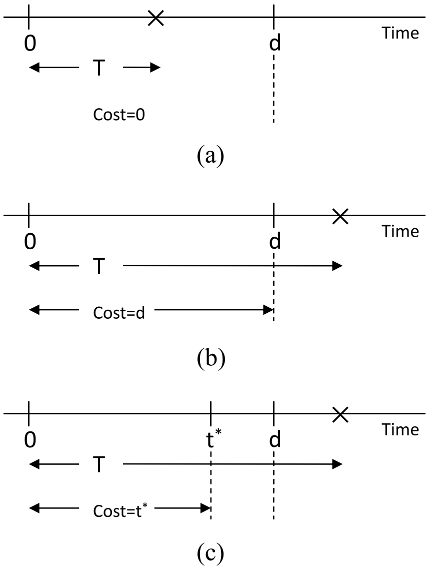
We note that T has a conditional pdf.

1) When there is no attack, $p_{\bar{J}}$, $q_{\bar{J}}$, X_i^s, and X_i^f are substituted in (7)–(9) to find the following.
   a) $f_T(t|\bar{J})$ by finding the inverse of $L_T(s)$.
   b) Mean of T: $m_{\bar{J}} = E[T]$.
   c) Variance of T: $\sigma_{\bar{J}}^2 = \sigma_T^2$.
2) When there is an attack, p_J, q_J, X_i^s, and X_i^f are substituted in (7)–(9) to find the following.
   a) $f_T(t|J)$ by finding the inverse of $L_T(s)$.
   b) Mean of T: $m_J = E[T]$.
   c) Variance of T: $\sigma_J^2 = \sigma_T^2$.

   In many cases, finding the inverse of $L_T(s)$ is not trivial. In such cases, we make Gaussian assumptions[1]

$$\begin{aligned} f_T(t|J) &\sim \mathcal{N}(m_J, \sigma_J^2) \\ f_T(t|\bar{J}) &\sim \mathcal{N}(m_{\bar{J}}, \sigma_{\bar{J}}^2). \end{aligned} \tag{11}$$

[1] This assumption can be justified by invoking the central limit theorem; since T involves a summation of random quantities.



(a)

(b)

(c)

Fig. 5. Different possible scenarios for total transmission time T relative to the time threshold t^* and the deadline d. (a) $T < d$. (b) $T \geq d$. (c) $T \geq d$.

Algorithm 1 Transmission Early-Stopping Algorithm

```
Input:
d: The deadline value.
t*: The transmission-time threshold.
BEGIN
 1: Start timer
 2: Transmit the packet
 3: if transmission attempt failed then
 4:     $t$ = current timer value
 5:     if $t < d$ then
 6:         if $t > t^*$ then
 7:             A jammer is detected, switch channel
 8:         else
 9:             No jammer detected, go to 2
10:         end if
11:     else
12:         $t \geq d$. Deadline reached. Packet failed. STOP.
13:     end if
14: else
15:     Transmission success. STOP.
16: end if
END
```

IV. Proposed Early-Stopping Algorithm

A. Problem Statement and Formulation

We consider a situation where a packet error is handled via repeated retransmissions until a successful reception is acknowledged, or a deadline of d seconds is exceeded. As mentioned earlier, for a certain packet, the total transmission/retransmission time until successful delivery is modeled via the RV T characterized in (7)–(9). Fig. 5(a) shows the case where $T < d$. This represents the case when the transmission/retransmission process ends before exceeding the deadline. Here, the T seconds of transmission lead eventually to a successful packet delivery, and therefore, the wasted time (cost) is zero. On the other hand, Fig. 5(b) shows the case where $T > d$. In this case, the transmission/retransmission processes takes a long time that goes past the deadline. Here, we consider that a time cost of d seconds is incurred; since d seconds are wasted on unsuccessful transmissions. As we mentioned earlier, in this paper, we propose to equip the transmitter with the capability to realize, at a time point earlier than d, that the packet at hand is doomed to failure, and it is necessary to switch to a safe channel. In the proposed scheme, the transmitter does not wait until the deadline is reached. Instead, when the transmission time exceeds a certain threshold $t^* < d$, the transmitter gives up transmission over the current channel and switches to a safe channel. The proposed scheme is shown in Fig. 5(c). Note that when the transmitter switches to a new channel, the amount of wasted time is only t^*, which is less that d. Further, the transmitter will still have some time remaining before the deadline is reached. This time is used to do more retransmission attempts over the new channel, which could end up in a success.

Our main objective is to determine an optimal time threshold $t^* < d$. This requires characterizing the transmission time under both cases: attack and no attack. Let H_1 and H_0 represent the hypotheses of an attack and no attack, respectively, then:

1) under hypothesis H_0, there is no jammer, and packet errors are due to channel conditions and collisions with legitimate packets. Therefore, transmission time T is governed by the pdf $f_T(t|\bar{J})$ characterized in Section III

$$f_T(t|H_0) = f_0(t) = f_T(t|\bar{J}) \tag{12}$$

2) under hypothesis H_1, a jammer exists, and packet errors are due to channel conditions, and collisions with legitimate packets and with jamming pulses. Therefore, transmission time T is governed by the pdf $f_T(t|J)$

$$f_T(t|H_1) = f_1(t) = f_T(t|J). \tag{13}$$

Now that we have a pdf for the packet transmission time T under both hypothesis, we could use a measurement of T as a statistic to make a decision between the two hypothesis. To achieve that, we propose keeping track of the packet transmission time. Whenever a packet is transmitted, a timer is started. Upon a failure and before retransmission, the current value of the timer is read. If the timer value exceeds a certain threshold t^*, it is more likely that the transmission time T is governed by the pdf in (13), and a jammer is assumed to exist, hence, transmission switches to a safe channel. On the other hand, if the timer value is less than t^*, then it is more likely that T is governed by the pdf in (12), and hence, there is no jamming, and the next retransmission attempt takes place on the same channel. Algorithm 1 presents a formal description of the proposed early-stop algorithm (applied to each packet). Next, we show in detail how to derive the threshold t^*.

B. Optimal Detector

To derive t^*, we cast the problem into a binary hypothesis testing formulation. In particular, we seek a Bayesian optimal detector [29]–[31]. As mentioned earlier, H_0 represents the hypothesis that a jammer does not exist, and H_1 represents the hypothesis that a jammer exists. The decision space is defined as $\theta = \{0, 1\}$. Given a measurement t from the RV T whose density function is $f_\theta(t)$, we aim to decide whether $\theta = 0$ (accept H_0), or $\theta = 1$ (accept H_1). Let $\delta(t)$ be a decision rule defined as follows:

$$\delta(t) = \begin{cases} 1 \text{ (accept } H_1), & t \in \mathcal{B}_1 \\ 0 \text{ (accept } H_0), & t \in \mathcal{B}_0 \end{cases} \tag{14}$$

where $\mathcal{B}_1$ and $\mathcal{B}_0$ are the time measurement intervals that would be mapped to decision 1 and decision 0, respectively. Our objective is to determine a threshold t^* that divides the time interval $[0, d]$ into $\mathcal{B}_0 = [0, t^*]$ and $\mathcal{B}_1 = [t^*, d]$.

Let's define L_{ij} for $i, j \in \{0, 1\}$ as the loss that results when we decide to accept the hypothesis H_i while the truth is H_j [29]. Different decisions result in different losses as we next show.

1) L_{00} and L_{11} are losses that result when making correct decisions. Therefore, these losses are assumed to be zero.
2) Decide to accept H_0 while the truth is H_1. Consequently, transmission continues on the same channel (H_0 is accepted; i.e., decide that a jammer does not exist and there is no need to switch channels), and the total transmission time T is governed by $f_1(t)$ (the truth is H_1; a jammer does exist). The loss in this case is
$$L_{01} = 0 \times P_1[T < d] + d \times P_1[T \geq d] \tag{15}$$
where $P_1[.]$ is a probability computed via $f_1(t)$ (13).
3) Decide to accept H_1 while the truth is H_0. Consequently, transmission *needlessly* switches to a new channel, where the total transmission time T is governed by $f_0(t)$. Channel switching takes s seconds (switching cost). In this case, since transmission until success in addition to channel switching should be done before the deadline, the probability of successful packet delivery becomes
$$P_0[T + s < d] \tag{16}$$
which is equivalent to
$$P_0[T < d - s] \tag{17}$$
where $P_0[.]$ is a probability computed via $f_0(t)$ (12). The loss is
$$L_{10} = 0 \times P_0[T < d - s] + d \times P_0[T \geq d - s] + s. \tag{18}$$
Note that a switching cost term s is included in the cost expression in (18); since here, the channel switching is needless, and it only leads to wasting s seconds that could have been otherwise utilized in transmission over the old channel.

The risk $R(\theta, \delta)$ [29] associated with a hypothesis is the average of the loss under that hypothesis

$$R(\theta, \delta) = \begin{cases} L_{01}P_1[\delta(t) = 0] + L_{11}P_1[\delta(t) = 1], & \theta = \theta_1 \\ L_{10}P_0[\delta(t) = 1] + L_{00}P_0[\delta(t) = 0], & \theta = \theta_0. \end{cases} \tag{19}$$

Bayes risk $R(\delta)$ [29] is the average of $R(\theta, \delta)$ over the distribution of θ

$$\begin{aligned} R(\delta) &= E_\theta R(\theta, \delta) \\ &= \pi_J(L_{01}P_1[\delta(t) = 0] + L_{11}P_1[\delta(t) = 1]) \\ &\quad + \pi_{\bar{J}}(L_{10}P_0[\delta(t) = 1] + L_{00}P_0[\delta(t) = 0]). \end{aligned} \tag{20}$$

The Bayes decision rule minimizes the Bayes risk. It is given by the following likelihood ratio test (LRT) [29]:

$$\delta(t) = \begin{cases} 1, & \frac{f_1(t)}{f_0(t)} \geq \eta \\ 0, & \frac{f_1(t)}{f_0(t)} < \eta \end{cases} \tag{21}$$

where

$$\eta = \frac{\pi_{\bar{J}}(L_{10} - L_{00})}{\pi_J(L_{01} - L_{11})}. \tag{22}$$

Plugging the loss (15) and (18) into (22), we get

$$\begin{aligned} \eta &= \frac{\pi_{\bar{J}}}{\pi_J} \frac{0 \times P_0[T < d - s] + d \times P_0[T > d - s] + s}{0 \times P_1[T < d] + d \times P_1[T > d]} \\ &= \frac{\pi_{\bar{J}}}{\pi_J} \frac{d \times P_0[T > d - s] + s}{d \times P_1[T > d]}. \end{aligned} \tag{23}$$

After determining η using (23), the time domain threshold t^* is obtained by solving (for t^*)

$$\eta = \frac{f_1(t^*)}{f_0(t^*)}. \tag{24}$$

V. 802.11 Distributed Coordinated Function

The DCF is the fundamental mechanism to access the medium specified by the 802.11 protocol [32]. This mechanism is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol where a packet collision is handled by a random binary exponential backoff according to certain rules that are briefly discussed in the sequel. In this paper, we focus on the RTS/CTS access mechanism of the DCF.

A. RTS/CTS Handshaking Process

1) A source station with a packet to transmit senses the channel. If the channel is idle for a period of time, denoted the distributed interframe space (DIFS), the station waits for a random backoff time before sending a short request-to-send (RTS) message. Otherwise (i.e., the channel is busy), the station keeps on sensing the channel until it is sensed idle for DIFS.
2) Upon successfully receiving the RTS, the receiver station waits for a period of time, denoted the short interframe space (SIFS), before responding with a short clear-to-send (CTS) message.

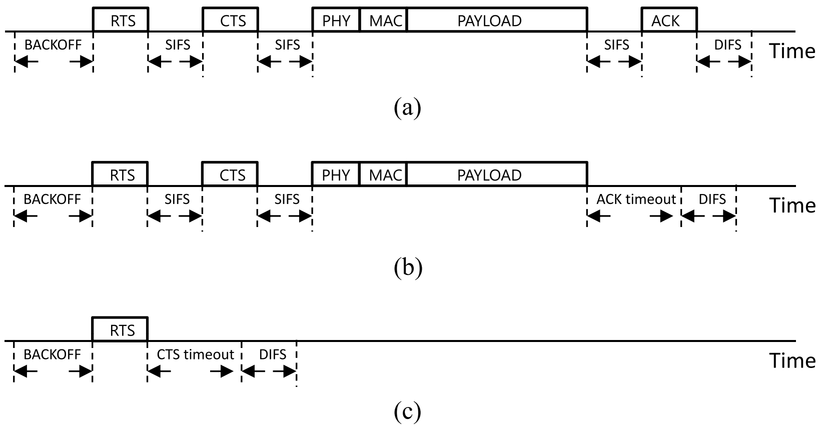


Fig. 6. Time sequence of the steps of transmitting a single packet. (a) Successfully transmitted packet. (b) Collision during the data part of the packet or the ACK. (c) Collision during RTS or CTS.

3) When the CTS is successfully received by the source station, packet transmission starts after waiting SIFS. In the meanwhile, all nodes that had heard the RTS or CTS messages refrain from transmission.
4) After completely receiving the packet, the destination station waits for SIFS before sending an acknowledgment (ACK) message to the source station, signaling successful reception. Upon receiving the ACK, the source waits for DIFS before following the aforementioned steps to send a new packet. Also, all nodes that were refraining from transmission, due to hearing the RTS or CTS messages, can transmit again once they hear the ACK.

Fig. 6(a) shows the time sequence of the RTS/CTS handshaking process. Collision avoidance is accomplished by prohibiting intermediate stations, upon hearing the RTS or CTS, from transmission, until they hear the ACK. In addition to waiting a random backoff time before transmission, which reduces the probability of transmitting at the same time.

In this section, we aim to apply the transmission-time model developed in Section III to characterize the transmission time of a single packet in 802.11 under both jamming and no-jamming conditions. In our analysis, we follow [33] in assuming perfect channel conditions, and thus, collisions are the only source of failure. By examining (7)–(9), we realize that we need to specify the following.

1) The probability of packet success under the jamming and no-jamming states (i.e., p_J and $p_{\bar{J}}$, respectively).
2) Transmission time of a packet succeeding or failing at stage i (i.e., X_i^s and X_i^f, respectively).

The aforementioned quantities highly depend on the backoff process that we briefly describe.

B. Binary Exponential Backoff Process

After the channel is idle for DIFS, the source station enters the backoff phase, where it waits for a random number of time slots, each of which of length t_{slot}. The source station is allowed to transmit only at the beginning of each slot. The details of the backoff phase is as follows.

1) Let stage be a variable that denotes the current backoff stage. The process starts at stage 0 (i.e., stage = 0). Right after the channel is sensed idle for DIFS, a backoff counter is set to a value chosen uniformly at random in the range $(0, w-1)$, where w, called the contention window (CW), is set according to the following equation:
$$w = 2^{\text{stage}}\text{CW}_{\min} \tag{25}$$
where $\text{CW}_{\min}$ is the minimum CW. Since stage is initially set to zero, w is initialized to $\text{CW}_{\min}$.
2) Each time the channel is sensed idle, the backoff timer is decremented by one time slot. Whenever the timer reaches zero, a transmission attempt is made.
3) With probability q_c, the transmission attempt fails (i.e., a collision occurs). In this case, stage is incremented by one, and w is doubled according to (25). Subsequently, in the new stage, the backoff timer is reset to a value chosen randomly from the (updated) interval $(0, w-1)$. The algorithm goes to step 2.
   a) The maximum value w can reach (due to repeated transmission attempt failures) is $\text{CW}_{\max} = 2^{n-1}$ $\text{CW}_{\min}$, where n is the maximum number of stages. When w reaches $\text{CW}_{\max}$, and an unsuccessful transmission occurs, w remains fixed at $\text{CW}_{\max}$, and the algorithm goes to step 2.
   b) If the attempt is successful (collision-free), which happens with probability $p_c = 1 - q_c$, w is reset to $\text{CW}_{\min}$, and used in transmitting a new packet.

In [33], the backoff process is modeled via a discrete-time Markov chain whose state transition diagram is shown in Fig. 7. The figure shows an example with $\text{CW}_{\min} = 2$, $\text{CW}_{\max} = 8$, and $n = 3$ stages. The state number at each stage represents the value of the backoff counter. It is clear in the figure that, whenever the counter reaches zero (state 0 is reached), a transmission attempt is made, and the chain either moves down to a state in the next stage (if the attempt fails, with probability q_c), or moves up to a state in stage 0 (if the attempt succeeds, with probability p_c).

Let τ be the probability that a station in the network transmits at an arbitrarily chosen time slot. It is shown in [33] that

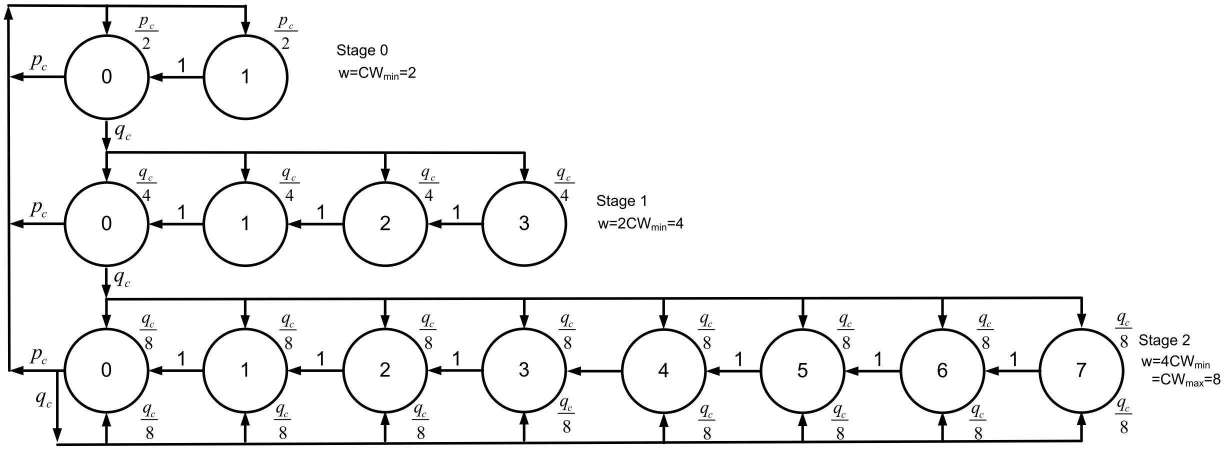


Fig. 7. State transition diagram of the random binary exponential backoff process with $\mathrm{CW}_{\min} = 2$ and $\mathrm{CW}_{\max} = 8$, and the maximum number of backoff stages $n = 3$.

τ is the summation of the steady-state probabilities of state zero at all stages in Fig. 7, which is given by

$$\tau = \frac{2(1 - 2q_c)}{(1 - 2q_c)(W + 1) + q_c W((1 - (2q_c)^n)} \tag{26}$$

where $W = \mathrm{CW}_{\min}$.

Since q_c is the probability of collision whenever a station makes a transmission attempt in a time slot, q_c represents the probability that at least one other station in the network is making a simultaneous transmission attempt. For a network with m stations, this probability is given by

$$q_c = 1 - (1 - \tau)^{m-1}. \tag{27}$$

Note that (26) and (27) form a system of two nonlinear equations with two unknowns through which τ and q_c are found by numerical solution methods [33].

Since the ith transmission attempt takes place in the ith stage ($i = 0, 1, \ldots, n - 1$), in which the backoff timer counts down starting from a value chosen uniformly at random from the interval $(0, w - 1)$, where w equals 2^i $\mathrm{CW}_{\min}$, the backoff time (the average time the backoff timer takes to reach zero) in the ith transmission attempt (stage) is given by

$$t_{\mathrm{bo}}(i) = \frac{1}{2^i W} \sum_{k=1}^{2^i W} k \; t_{\mathrm{slot}} \tag{28}$$

where $W = \mathrm{CW}_{\min}$.

C. Transmission Time of Single Packet

In this section, we find an expression for the transmission time of a packet succeeding or failing at stage i (i.e., X_i^s and X_i^f, respectively). There are many cases we need to address.

1) *Successful Transmission Attempt:* No collision occurs during RTS, CTS, data (which includes header and payload), and ACK. The total duration of a successfully transmitted packet is shown in Fig. 6(a), and is given by (success occurs in the ith attempt) [33]

$$t^s(i) = t_{\mathrm{bo}}(i) + t_{\mathrm{RTS}} + t_{\mathrm{SIFS}} + t_{\mathrm{prop}} + t_{\mathrm{CTS}} + t_{\mathrm{SIFS}} + t_{\mathrm{prop}} + t_{\mathrm{data}} + t_{\mathrm{SIFS}} + t_{\mathrm{prop}} + t_{\mathrm{ACK}} + t_{\mathrm{DIFS}} + t_{\mathrm{prop}} \tag{29}$$

where t_{prop} is propagation delay and t_{data} is the time of a packet payload plus PHY and MAC headers. The other time variables are indicated by their names. It is worth mentioning that t_{data} is random (payload length is random). Nevertheless, for simplicity, we assume a deterministic t_{data} in Section VI.

2) *Failed Transmission:* Collision with either a jamming or a legitimate packet occurs during transmission. There are two cases.

a) The collision occurs during transmitting the data or the ACK. The transmission duration in this case is shown in Fig. 6(b), and is given by

$$t^f_{\mathrm{pl/ACK}}(i) = t_{\mathrm{bo}}(i) + t_{\mathrm{RTS}} + t_{\mathrm{SIFS}} + t_{\mathrm{prop}} + t_{\mathrm{CTS}} + t_{\mathrm{SIFS}} + t_{\mathrm{prop}} + t_{\mathrm{data}} + t_{\text{timeout-ACK}} + t_{\mathrm{DIFS}}. \tag{30}$$

b) The collision occurs during transmitting the RTS or CTS. The transmission duration in this case is shown in Fig. 6(c), and is given by

$$t^f_{\mathrm{RTS/CTS}}(i) = t_{\mathrm{bo}}(i) + t_{\mathrm{RTS}} + t_{\text{timeout-CTS}} + t_{\mathrm{DIFS}}. \tag{31}$$

Now, we combine the results of the analysis in the previous sections to find an expression for the parameters of the transmission time model in (7), namely X_i^s, X_i^f, p, and q under the jamming, and the no-jamming cases.

1) *No-Jamming Case:* The only cause of a packet failure is a collision with a legitimate packet transmitted by another station. We assume that collisions occur during the source's RTS or the receiver's CTS. No collision could happen during the data or the ACK once the RTS and the CTS are exchanged successfully, as the surrounding stations will refrain from transmission once they hear these massages. The probability of a collision during RTS equals the probability of a collision during CTS, and they are equal to q_c, which is computed by solving (26) and (27). Hence, in the no-jamming case, probability of packet success is given by

$$p_{\bar{J}} = p_c^2 \tag{32}$$

and $q_{\bar{J}} = 1 - p_{\bar{J}}$. Further, we have

$$X_i^f = t^f_{\mathrm{RTS/CTS}}(i) \tag{33}$$

and

$$X_i^s = t^s(i). \tag{34}$$

2) *Jamming Case:* We seek to find p_J. There are four transmissions that are prone to jamming, namely RTS, CTS, payload, and ACK. We apply (5) to these transmissions as follows.

   a) *RTS:* The value of p_{T_x}, denoted here as $p_{T_x-\mathrm{RTS}}$, is computed using (3) with $T_x = t_{\mathrm{RTS}}$, which gives

$$p_{T_x-\mathrm{RTS}} = e^{-\delta t_{\mathrm{RTS}}}. \tag{35}$$

   Let $p_{\bar{J}-\mathrm{RTS}}$ be the probability of surviving a collision with a legitimate packet in RTS. Thus, the probability of surviving a jamming pulse in RTS is given by

$$\begin{aligned} p_{J-\mathrm{RTS}} &= p_{nj} \times p_{T_x-\mathrm{RTS}} \times p_{\bar{J}-\mathrm{RTS}} \\ &= \frac{\sigma}{\sigma+\delta} \times e^{-\delta t_{\mathrm{RTS}}} \times p_c. \end{aligned} \tag{36}$$

   b) *CTS:* Similarly, given that RTS was received successfully, the probability of surviving a jamming pulse in CTS is given by

$$\begin{aligned} p_{J-\mathrm{CTS}} &= p_{nj} \times p_{T_x-\mathrm{CTS}} \times p_{\bar{J}-\mathrm{CTS}} \\ &= \frac{\sigma}{\sigma+\delta} \times e^{-\delta t_{\mathrm{CTS}}} \times p_c \end{aligned} \tag{37}$$

   where $p_{\bar{J}-\mathrm{CTS}}$ is the probability of surviving a collision with a legitimate packet in CTS.

   c) *Payload:* Let $p_{\bar{J}-\mathrm{data}}$ be the probability of surviving a collision with a legitimate packet during the data transmission. Given that RTS and CTS were successful, it is assumed that $p_{\bar{J}-\mathrm{data}} = 1$. Hence, the probability of surviving a jamming pulse during the data transmission (given that RTS and CTS were successful) is given by

$$\begin{aligned} p_{J-\mathrm{data}} &= p_{nj} \times p_{T_x-\mathrm{data}} \times p_{\bar{J}-\mathrm{data}} \\ &= \frac{\sigma}{\sigma+\delta} \times e^{-\delta t_{\mathrm{data}}}. \end{aligned} \tag{38}$$

   d) *ACK:* The probability of surviving a jamming pulse during ACK (given that RTS, CTS, and data were successful) is given by

$$\begin{aligned} p_{J-\mathrm{ACK}} &= p_{nj} \times p_{T_x-\mathrm{ACK}} \times p_{\bar{J}-\mathrm{ACK}} \\ &= \frac{\sigma}{\sigma+\delta} \times e^{-\delta t_{\mathrm{ACK}}} \end{aligned} \tag{39}$$

   where $p_{\bar{J}-\mathrm{ACK}}$ is the probability of surviving a collision with a legitimate packet during ACK (equals one, given that RTS and CTS were successful).

Finally, the probability of packet success (the whole packet is successful) is given by

$$p_J = p_{J-\mathrm{RTS}} \times p_{J-\mathrm{CTS}} \times p_{J-\mathrm{data}} \times p_{J-\mathrm{ACK}}. \tag{40}$$

Let $q_{J-\mathrm{RTS/CTS}}$ be the probability of packet failure during RTS or CTS

$$q_{J-\mathrm{RTS/CTS}} = q_{J-\mathrm{RTS}} + p_{J-\mathrm{RTS}} \times q_{J-\mathrm{CTS}}. \tag{41}$$

Also, let $q_{J-\mathrm{pl/ACK}}$ be the probability of packet failure in during data transmission or ACK (given that RTS and CTS were successful)

$$\begin{aligned} q_{J-\mathrm{data/ACK}} &= p_{J-\mathrm{RTS}} \times p_{J-\mathrm{CTS}} \times q_{J-\mathrm{data}} \\ &\quad + p_{J-\mathrm{RTS}} \times p_{J-\mathrm{CTS}} \times p_{J-\mathrm{data}} \times q_{J-\mathrm{ACK}}. \end{aligned} \tag{42}$$

Then, under the jamming conditions, we have

$$X_i^f = \frac{q_{J-\mathrm{RTS/CTS}}}{z} t^f_{\mathrm{RTS/CTS}}(i) + \frac{q_{J-\mathrm{data/ACK}}}{z} t^f_{\mathrm{data/ACK}}(i) \tag{43}$$

where $z = q_{J-\mathrm{RTS/CTS}} + q_{J-\mathrm{data/ACK}}$. Further, the value of X_i^s is given in (34).

Now we are ready to determine the conditional pdf of T.

1) *No Jamming:* Use the values of $p_{\bar{J}}$, X_i^f, and X_i^s evaluated using (32)–(34), respectively, to evaluate $m_{\bar{J}}$ and $\sigma^2_{\bar{J}}$ using (8) and (10), respectively. Then,

$$f_T\left(t|\bar{J}\right) \sim \mathcal{N}\left(m_{\bar{J}}, \sigma^2_{\bar{J}}\right).$$

2) *Jamming:* Use the values of p_J, X_i^f, and X_i^s evaluated using (40), (43), and (34), respectively, to evaluate m_J and σ^2_J using (8) and (10), respectively. We have

$$f_T(t|J) \sim \mathcal{N}\left(m_J, \sigma^2_J\right).$$

By substituting in (23)

$$\eta = \frac{\pi_{\bar{J}}}{\pi_J} \frac{Q\left(\frac{d-s-m_{\bar{J}}}{\sigma_{\bar{J}}}\right) + s/d}{Q\left(\frac{d-m_J}{\sigma_J}\right)} \tag{44}$$

where $Q(x)$ is the Q-function of the standard normal distribution. Using (24), we solve for t^* the following inequality:

$$\eta \leq \frac{\sigma_{\bar{J}}}{\sigma_J} \exp\left(\frac{\left(t^* - m_{\bar{J}}\right)^2}{2\sigma^2_{\bar{J}}} - \frac{(t^* - m_J)^2}{2\sigma^2_J}\right) \tag{45}$$

which results in

$$t^* = \pm\sqrt{\frac{1}{a}(e-c) + \left(\frac{b}{2a}\right)^2} - \left(\frac{b}{2a}\right) \tag{46}$$

where a, b, c, and e are defined as follows:

$$\begin{aligned} a &= \sigma^2_J - \sigma^2_{\bar{J}} \\ b &= 2\left(\sigma^2_{\bar{J}} m_J - \sigma^2_J m_{\bar{J}}\right) \\ c &= \sigma^2_J m^2_{\bar{J}} - \sigma^2_{\bar{J}} m^2_J \\ e &= 2\sigma^2_{\bar{J}} \sigma^2_J \; ln\left(\frac{\sigma_J}{\sigma_{\bar{J}}}\eta\right). \end{aligned}$$

We pick the smaller solution (earlier t^*).

It is worth mentioning that a false alarm (i.e., initiating a channel switch while there is no jammer) has the consequence of wasting s seconds needlessly, and this increases the chances of missing the deadline. Also, a miss-detection (deciding not to switch while there is a jammer) has the consequence of making many retransmission attempts that fail due to jamming, and this increases the likelihood of exceeding the deadline. Theoretically, the probability of detection

TABLE I
PARAMETERS RELATED TO 802.11

Parameter	Value
CW_{min}	16
CW_{max}	1024
PHY header	128 bits
MAC header	272 bits
packet payload	8184 bits
ACK	112 bits + PHY header
RTS	160 bits + PHY header
CTS	112 bits + PHY header
Channel bit rate	100 MBPS
t_{prop}	1 μs
t_{slot}	50 μs
$t_{\text{timeout-ACK}}$	300 μs
$t_{\text{timeout-CTS}}$	300 μs
t_{DIFS}	128 μs
t_{SIFS}	28 μs
m	3 stations
r	7 attempts

(P_d) and probability of false alarm (P_{fa}) can be computed as follows:

$$P_d = P_1\left[t > t^*\right] = Q\left(\frac{t^* - m_J}{\sigma_J}\right) \tag{47}$$

$$P_{fa} = P_0\left[t > t^*\right] = Q\left(\frac{t^* - m_{\bar{J}}}{\sigma_{\bar{J}}}\right). \tag{48}$$

VI. PERFORMANCE EVALUATION

In this section, we conduct simulation experiments to investigate the performance of the proposed early-stop detection algorithm. In particular, we focus on the nonreactive jamming case (reactive jamming differs only in the way of obtaining p_J). We develop our simulations using MATLAB [34].

A. Simulation Setup

We simulate an IoT device that employs 802.11's RTS/CTS DCF mechanism to access and transmit over a channel. The values of the system parameters that we employ are those specified by the 802.11 standard [32], [33]. Parameter values related to 802.11 are listed in Table I. We assume that packet failures could occur due to the following.

1) Normal collisions caused by DCF failures, resulting in more than one node transmitting simultaneously.
2) Malicious collisions caused by jamming.

We set the retransmission limit parameter r to 7. The traffic is time-critical where a packet needs to be successfully delivered within a deadline value $d = 80$ ms. Two values of channel switching cost s are simulated: 20 and 40 ms. Transmission time of a packet in the success and failure cases are computed as described in Section V.

TABLE II
PARAMETERS RELATED TO THE JAMMING ATTACK AND THE TIME-CRITICAL APPLICATION

Parameter	Value
d	80 ms
s	20 ms, 40 ms
π_J Low Hostility π_J Medium Hostility π_J High Hostility	0.1 0.5 0.9
$p_{\bar{J}}$	0.6874
p_J Aggressive p_J Highly Aggressive p_J Extremely Aggressive	0.1217 0.0620 0.0133

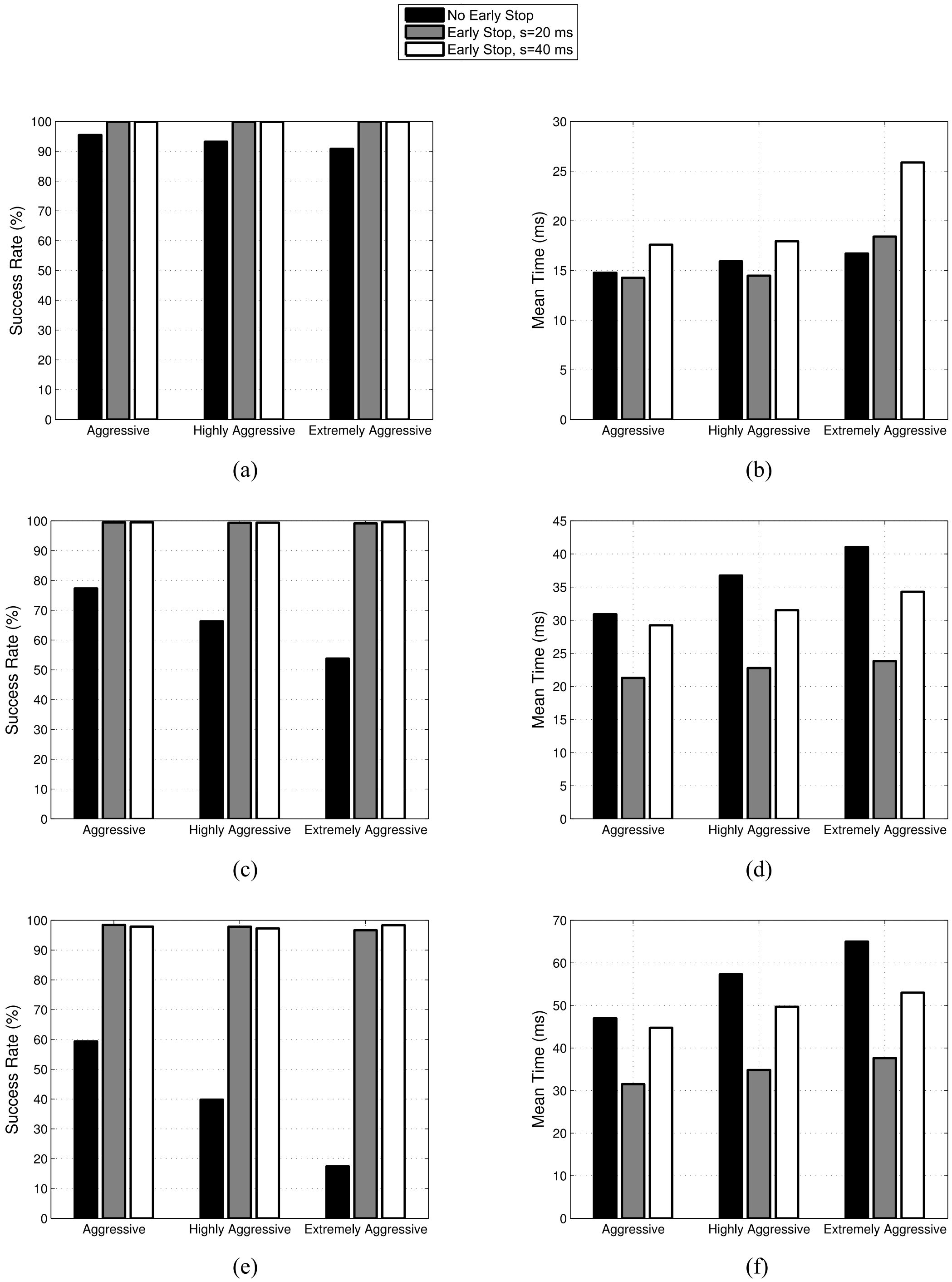
1) We simulate an environment with three hostility levels.
   a) Low hostility: $\pi_J = 0.1$.
   b) Moderate hostility: $\pi_J = 0.5$.
   c) High hostility: $\pi_J = 0.9$.
2) *Probability of Success for a Single Packet Transmission Under No Jamming Attack ($p_{\bar{J}}$):* Packet errors are due to normal collisions. The value of p_c is computed by solving a system consisting of (26) and (27). The value of $p_{\bar{J}}$ is computed using (32). In our set up, $p_{\bar{J}} = 0.6874$.
3) *Probability of Success for a Single Packet Transmission Under a Jamming Attack (p_J):*
   a) The mean width of the jamming pulse $E[D] = 1.0684$ ms. Hence $\sigma = 935.9561$.
   b) The mean interarrival time between jamming pulses $E[I]$ is used to control the aggressiveness level of the jamming attack. Three aggressiveness levels are simulated.
      i) *Aggressive Attack:* $E[I] = 7.4790$ ms. Hence $\delta = 133.7080$, and $p_J = 0.1217$.
      ii) *Highly Aggressive Attack:* $E[I] = 5.3421$ ms. Hence $\delta = 187.1912$, and $p_J = 0.0620$.
      iii) *Extremely Aggressive Attack:* $E[I] = 2.7$ ms. Hence $\delta = 311.9854$, and $p_J = 0.0133$.

   The values of p_J are computed using (40) as described in Section V. Parameters related to the jammer, and the time-critical application are listed in Table II.

In each simulation experiment, we simulate transmitting 100 000 packets in a network that consists of $m = 3$ stations. The main performance metrics that we report are the probability of detection (P_d), probability of false alarm (P_{fa}), success rate (i.e., the percentage of packets that were successfully delivered), and the mean time of packet delivery.

B. Simulation Results

Fig. 8 demonstrates the performance gain attained by employing the early-stop algorithm. Fig. 8(a) shows the percentage of packets that are successfully delivered to the destination after sending 100 000 packets in a mildly hostile environment ($\pi_J = 0.1$), under aggressive, highly aggressive, and extremely aggressive jamming attacks. The figure



(a) (b)

(c) (d)

(e) (f)

Fig. 8. Performance gain, in terms of packet success rate and packet delivery mean time, attained by the early-stop detection scheme under aggressive, highly aggressive, and extremely aggressive jamming attacks with s equals 20 and 40 ms. (a) and (b) Low hostility $\pi_J = 0.1$. (c) and (d) Medium hostility $\pi_J = 0.5$. (e) and (f) High hostility $\pi_J = 0.9$.

presents the success rates with and without employing the early-stop algorithm, and for $s = 20$ and 40 ms. The figure shows that, when the early-stop scheme is not used, the success rate degrades when the aggressiveness level increases. This is expected, as more attempts become needed to successfully deliver a certain packet, and hence, it is more likely to miss the deadline. On the other hand, the figure shows that employing the early-stop scheme with $s = 20$ or $s = 40$ ms, results in a success rate close to 100% over all levels of aggressiveness. This is achieved by switching to a new channel as early as possible, which increases the chances of successfully delivering the packet before the deadline is reached. For the $s = 40$ ms case, the switch will be even earlier (t^* smaller) to ensure meeting the deadline.

Fig. 8(b) shows the mean time of sending (successfully or unsuccessfully) a single packet. The figure shows that, as the

TABLE III
THRESHOLD t^* AND THE RESULTING PROBABILITY OF DETECTION (P_d), PROBABILITY OF FALSE ALARM (P_{fa}), AND SUCCESS RATE

			Hostility											
			Low ($\pi_J = 0.1$)				Medium ($\pi_J = 0.5$)				High ($\pi_J = 0.9$)			
			t^* (ms)	P_d	P_{fa}	Succ. Rate (%)	t^* (ms)	P_d	P_{fa}	Succ. Rate (%)	t^* (ms)	P_d	P_{fa}	Succ. Rate (%)
Aggressiveness	Aggressive	$s = 20$ ms	1.5779	0.8220	0.0970	99.8720	3.5499	0.7812	0.0307	99.4680	6.3742	0.7660	0.0102	98.4570
		$s = 40$ ms	1.0407	0.8189	0.0976	99.8590	2.8726	0.7847	0.0312	99.4920	5.3147	0.7690	0.0095	97.8600
	Highly aggressive	$s = 20$ ms	1.5874	0.8985	0.0978	99.8470	3.5791	0.8827	0.0299	99.3360	6.4661	0.8691	0.0083	97.8530
		$s = 40$ ms	1.0463	0.8942	0.0979	99.8440	2.8939	0.8841	0.0295	99.3690	5.3748	0.8714	0.0084	94.3340
	Extremely aggressive	$s = 20$ ms	0.9903	0.9860	0.3092	99.9070	2.8746	0.9698	0.0306	99.1260	5.4465	0.9686	0.0063	96.6440
		$s = 40$ ms	0.4708	0.9848	0.3128	99.8900	2.2323	0.9778	0.0964	99.5770	4.5097	0.9715	0.0308	98.3190

aggressiveness level goes up, the mean time increases, since more trials are needed to deliver a packet. Also, the highest mean time is attained when the switching cost is 40 ms. Moving to Fig. 8(c) and (e), we can see that increasing the hostility level results in a dramatic decrease in the success rate when no detection scheme is used. Nevertheless, the early-stop scheme manages to maintain a success rate close to 100%, as a result of adopting the early channel switching policy. Further, taking a look at Fig. 8(d) and (f), it is remarkable to see that the mean time of packet delivery, when no detection scheme is used, eventually dominates the mean time of the early-stop scheme, despite the cost of channel switching in the latter scheme. The high success rate achieved by the early-stop scheme can be explained by taking a closer look at the detection accuracy. Table III lists the threshold t^* used in each experiment and the resulting probability of detection (P_d), probability of false alarm (P_{fa}) and packet success rate. We note that in general, the early-stop scheme achieves a high P_d and a low P_{fa} and thus a high success rate.

To further investigate the accuracy of the early-stop detector, the receiver operating characteristic (ROC) curves are presented in Fig. 9(a), (c), and (e) for the aggressive, highly aggressive, and extremely aggressive attacks, respectively. Each ROC curve is plotted by setting the threshold t^* to a value in the interval $[0, d]$, and then recording P_d versus P_{fa} after sending 100 000 packets. This is repeated for all $t^* \in [0, d]$ with $\pi_J = 0.5$. We can see that, as the aggressiveness level goes up, it becomes more possible to achieve a P_d that is close to 1 at a P_{fa} that is close to zero. The reasoning is that, although the mean of T under no attack remains fixed when the aggressiveness increases, the mean value of T under an attack increases. Hence, $f_T(t|J)$ and $f_T(t|\bar{J})$ become further apart, and hence the decision accuracy increases.

Fig. 9(b), (d), and (f) depicts the P_d and P_{fa} separately as a function of t^*. Note that the P_{fa} remains almost the same across the three figures, since P_{fa} depends on $f_T(t|\bar{J})$ as (48) indicates, and the latter is independent of the jamming attack. The figures also show that a larger P_d could be achieved at a certain t^* value as the aggressiveness level increases, since $f_T(t|J)$ moves further apart from $f_T(t|\bar{J})$. It could also be noticed that, at each aggressiveness level, the P_d and P_{fa} go down as t^* increases. This is because the algorithm becomes more reluctant towards switching to a new channel, and hence the number of incidents of erroneously switching (while there is no jamming) decreases (i.e., P_{fa} goes down). Unfortunately, this also causes the number of correct switching incidents (when there is indeed a jammer) to decrease (i.e., P_d goes down).

We next compare the performance of the proposed early-stop detection scheme with the JADE presented in [8] and [26]. JADE is a state of the art algorithm that employs the MIR metric to detect the existence of a jammer. The MIR is the probability that a certain packet will exceed the deadline. Let v denote the MIR. The value of v satisfy the following inequality [8]:

$$v \le \frac{q_J^r c}{\left(1 - q_J^r\right)(d - c) + q_J^r c} \tag{49}$$

where r is the retransmission limit, d is the deadline, and c is the expected packet size (under jamming) given by

$$\begin{aligned} c = \sum_{i=1}^{r} q_J^{i-1} p_J \Big\{ & q_{J-\text{RTS/CTS}} t^f_{\text{RTS/CTS}}(i) \\ & + q_{J-\text{pl/ACK}} t^f_{\text{pl/ACK}}(i) + p_J t^s(i) \Big\} \\ & + q_J^r \left(\frac{q_{J-\text{RTS/CTS}}}{z} t^f_{\text{RTS/CTS}}(r) + \frac{q_{J-\text{pl/ACK}}}{z} t^f_{\text{pl/ACK}}(r) \right) \end{aligned} \tag{50}$$

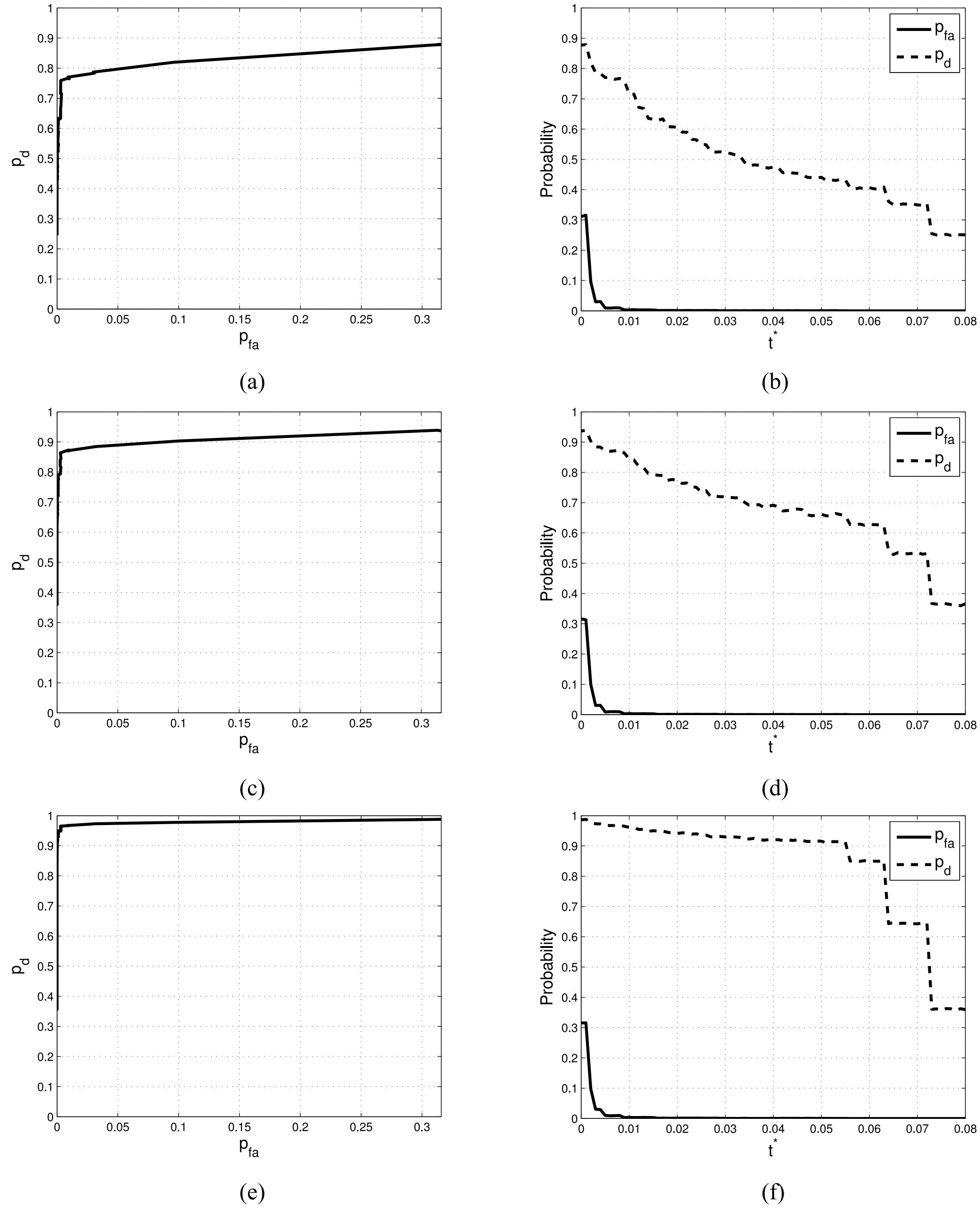
where $z = q_{J-\text{RTS/CTS}} + q_{J-\text{pl/ACK}}$. In [8], it was observed that v, when plotted as a function of q_J, remains initially close to zero, and then it abruptly goes to one at a certain value of q_J denoted as q_J^*. This indicates that the probability of missing the deadline will be significant for $q_J \ge q_J^*$. JADE is based on estimating the probability of failure (the estimate is denoted $\hat{q}_J$), and comparing $\hat{q}_J$ to q_J^*. When $\hat{q}_J \ge q_J^*$, JADE decides that a jammer exists. The estimation of q_J is accomplished by sending N packets, and computing

$$\hat{q}_J = \frac{n_f}{N} \tag{51}$$

where n_f is the number of packets for which no ACK is received.

We now conduct an experiment to compare the performance of JADE and the early-stop scheme. We use the high hostility and extremely aggressive set up. The first step is to determine the threshold q_J^*. To do that, we use the values listed in Tables I and II to compute v as a function of q_J according to (49). The result is plotted in Fig. 10. Note that the threshold value (i.e., the abrupt transition point) $q_J^* \approx 0.47$.

Next, we decide the value of N at which the performance of JADE is maximized. Since we seek jamming detection on a packet-by-packet basis, N can not be greater than $r = 7$. While a larger value of N results in a more accurate estimate



(a) (b)

(c) (d)

(e) (f)

Fig. 9. (a), (c), and (e) ROC curves for the early-stop detection scheme under aggressive, highly aggressive, and extremely aggressive jamming attacks for threshold values $t^* \in [0, d]$, where $d = 0.08$ s. (b), (d), and (f) P_d and P_{fa} as a function of t^*.

of q_J, it takes a longer time, which could lead to missing the deadline. On the other hand, a smaller N can result in a less accurate estimate, but nonetheless, in less time. To decide the value of N that yields the best performance, we simulate JADE using the same setup described earlier (Tables I and II). We transmit 100 000 packets, and we record the P_d, P_{fa}, and the success rate for N in the range from 1 to 7. In each experiment, and for each packet, when a transmission attempt is made, a counter is incremented by one. If the transmission attempt fails, n_f is incremented by 1. When the counter reaches N, $\hat{q}_J$ is computed according to (51). If $\hat{q}_J \geq q_J^* = 0.47$, a channel switch occurs and attempts continue on the new channel. The counter is reset when it reaches N or when a channel switch occurs. The results are listed in Table IV. For $N = 1$, a low success rate is scored, this is expected, as the estimate $\hat{q}_J$ is extremely inaccurate. The success rate peaks at $N = 3$, and then it dramatically decreases for $N > 4$. The reason is that, for a larger N, the estimate is generated too late, in a way that the time remaining until the deadline is tight, which leads to a packet failure.

Finally, we compare the performance of JADE ($N = 3$ and $q_J^* = 0.47$) and the early-stop algorithm. The scenario we simulate is the highly hostile environment with extremely aggressive jamming. We report the P_d, P_{fa}, and success rate after sending 100 000 packets. The experiment is repeated for switching cost $s \in \{40, 45, 50, 55, 60\}$ ms. The results are

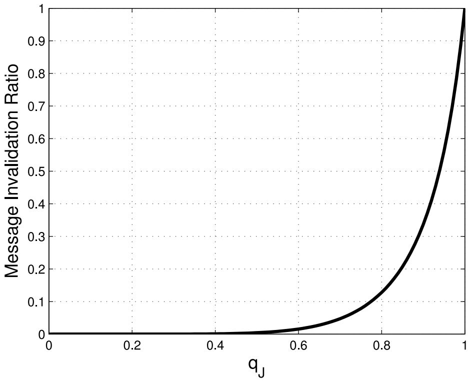


Fig. 10. MIR as a function of q_J with the highly hostile and extremely aggressive scenario.

TABLE IV
PERFORMANCE OF JADE FOR DIFFERENT VALUES OF N

N	p_d	p_{fa}	Succ. Rate (%)
1	0.9852	0.293	71.34
2	0.9892	0.4353	90.68
3	0.9761	0.1888	93.86
4	0.9674	0.1125	89.47
5	0.952	0.0435	64.94
6	0.9404	0.0265	28.23
7	0.9257	0.0223	23.36

TABLE V
PERFORMANCE OF JADE ($N = 3$ AND $q_J^* = 0.47$) VERSUS THE EARLY-STOP ALGORITHM AT DIFFERENT VALUES OF s. THE SIMULATED SCENARIO IS THE HIGH HOSTILITY WITH EXTREMELY AGGRESSIVE ATTACKS

	JADE			Early Stop		
s	P_d	P_{fa}	Succ. Rate (%)	P_d	P_{fa}	Succ. Rate (%)
0.040	0.9761	0.1856	94.18	0.9705	0.0323	98.27
0.045	0.9760	0.1888	93.88	0.9717	0.0270	95.35
0.050	0.9753	0.1807	84.89	0.9707	0.0307	95.14
0.055	0.9762	0.1845	77.97	0.9725	0.0287	93.39
0.06	0.9758	0.1790	53.07	0.9722	0.0271	82.76

listed in Table V. We note that, although the P_d for JADE is high, it suffers a high P_{fa}. On the other hand, the early-stop algorithm has a high P_d and a low P_{fa}. Therefore, the early-stop scheme outperforms JADE in terms of the success rate. Also, one can see that the success rate of JADE degrades dramatically starting from $s = 0.05$ s. The reasoning is that, for higher switching cost values, when a channel switch is decided, the remaining time until the deadline will not be sufficient to accomplish the costly switch. On the other hand, we can see that the early-stop algorithm does not suffer this problem, because the time threshold is computed taking into account the value of s. For instance, for larger s, t^* is smaller, so that the switch can take place earlier to meet the deadline.

VII. CONCLUSION

In this paper, we presented a detailed mathematical model for the transmission time under the existence and absence of a jamming attack. The mathematical model is used to design a Bayesian jamming detector where an optimal transmission-time threshold is derived. When transmission time for a certain packet exceeds this threshold, a jammer is detected, and transmission switches to a safe channel. The remaining time before the deadline is reached in utilized in retransmitting the packet over the safe channel. A main advantage of the proposed detection scheme is that it is a general framework that can be applied to many situations. We applied the proposed framework to the DCF medium access mechanism specified by the 802.11 standard. Through simulation results, we showed that the proposed scheme achieves significant performance gains in terms of the percentage of successfully delivered packets, probability of detection and probability of false alarm. Also, we showed that it outperforms JADE, a state-of-the-art jamming detection scheme for time-critical applications.

REFERENCES

[1] J. Lin *et al.*, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[3] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[4] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IoT) technologies, applications and challenges," in *Proc. IEEE Smart Energy Grid Eng. (SEGE)*, Aug. 2016, pp. 381–385.

[5] J. E. Siegel, S. Kumar, and S. E. Sarma, "The future Internet of Things: Secure, efficient, and model-based," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2386–2398, Aug. 2018.

[6] H. A. B. Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Spectrum assignment in cognitive radio networks for Internet-of-Things delay-sensitive applications under jamming attacks," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1904–1913, Jun. 2018.

[7] H. Farag, M. Gidlund, and P. Österberg, "A delay-bounded MAC protocol for mission- and time-critical applications in industrial wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 6, pp. 2607–2616, Mar. 2018.

[8] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1746–1759, Aug. 2014.

[9] K. Trivedi, *Probability and Statistics With Reliability, Queuing, and Computer Science Applications*, 2nd ed. Chichester, U.K.: Wiley, 2002.

[10] K. Trivedi and A. Bobbio, *Reliability and Availability Engineering, Modeling, Analysis, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[11] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*, 6th ed. Boston, MA, USA: Pearson, 2012.

[12] X. Chang, S. Lv, R. J. Rodriguez, and K. Trivedi, "Survivability model for security and dependability analysis of a vulnerable critical system," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–6.

[13] R. W. B. Fricks, H. H. Tseng, M. Pajic, and K. S. Trivedi, "Transient performance & availability modeling in high volume outpatient clinics," in *Proc. Annu. Rel. Maintainability Symp. (RAMS)*, Jan. 2017, pp. 1–6.

[14] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[15] D. Adamy, *A First Course in Electronic Warfare*. Boston, MA, USA: Artech House, 2001.

[16] R. A. Poisel, *Modern Communications Jamming Principles and Techniques*. London, U.K.: Artech House, 2006.

[17] L. Xue, X. Cao, C. Sun, and S. Jin, "Optimal jamming attack strategy against wireless state estimation: A game theoretic approach," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 5989–5995.

[18] G. Rezgui, E. V. Belmega, and A. Chorti, "Mitigating jamming attacks using energy harvesting," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 297–300, Feb. 2019.

[19] J. Ng, Z. Cai, and M. Yu, "A new model-based method to detect radio jamming attack to wireless networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–6.

[20] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2007, pp. 1307–1315.

[21] M. Strasser, B. Danev, and and S. Čapkun, "Detection of reactive jamming in sensor networks," *ACM Trans. Sensor Netw.*, vol. 7, no. 2, p. 16, 2010.

[22] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. 6th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Urbana, IL, USA, 2005, pp. 46–57. [Online]. Available: http://doi.acm.org/10.1145/1062689.1062697

[23] B. Yu and L.-Y. Zhang, "An improved detection method for different types of jamming attacks in wireless networks," in *Proc. 2nd Int. Conf. Syst. Informat. (ICSAI)*, 2014, pp. 553–558.

[24] K. Siddhabathula, Q. Dong, D. Liu, and M. Wright, "Fast jamming detection in sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 934–938.

[25] E. Bayraktaroglu *et al.*, "On the performance of IEEE 802.11 under jamming," in *Proc. 27th Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2008, pp. 1265–1273.

[26] Z. Lu, W. Wang and C. Wang, "From jammer to gambler: Modeling and detection of jamming attacks against time-critical traffic," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2011, pp. 1871–1879.

[27] Y. Wu, B. Wang, and K. J. R. Liu, "Optimal defense against jamming attacks in cognitive radio networks using the Markov decision process approach," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[28] R. Halloush and H. Liu, "Modeling and performance evaluation of jamming-tolerant wireless systems," *J. Ambient Intell. Humanized Comput.*, pp. 1–18, Nov. 2018. [Online]. Available: https://doi.org/10.1007/s12652-018-1113-8

[29] C. D. L. Scharf, *Statistical Signal Processing Detection, Estimation, and Time Series Analysis*. Reading, MA, USA: Addison-Wesley, 1991.

[30] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York, NY, USA: Springer-Verlag, 1994.

[31] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[32] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-1997, pp. 1–445, Nov. 1997.

[33] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[34] *MATLAB and Statistics Toolbox Release 2012b*. Natick, MA, USA: MathWorks, 2012.

Rami D. Halloush (GS'09–M'12) received the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2012.

He is currently an Assistant Professor with the Telecommunication Engineering Department, Yarmouk University, Irbid, Jordan. His current research interests include stochastic modeling of computer systems, computer security, wireless communications, computer networking, and cognitive radio.