This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

Real-Time Misbehavior Detection and Mitigation in Cyber-Physical Systems over WLANs

Xianghui Cao, *Member, IEEE*, Lu Liu, *Student Member, IEEE*, Wenlong Shen, *Student Member, IEEE*, Aurobinda Laha, Jin Tang, *Member, IEEE*, and Yu Cheng, *Senior Member, IEEE*

***Abstract*—In cyber-physical system (CPS) over IEEE 802.11e based wireless local area networks (WLANs), a misbehaving node can gain significant advantage over other normal nodes in terms of resource sharing by deliberately manipulating its protocol parameters. Due to the random spectrum-access nature of the protocol, it is challenging to detect the misbehaving node accurately and in real-time. Moreover, many existing misbehavior detectors, primarily designed for traditional IEEE 802.11 networks, become inapplicable in IEEE 802.11e networks with heterogeneous network configurations. In this paper, we propose novel real-time and light-weight countermeasures including a hybrid-share misbehavior detector and a packet-dropping based misbehavior mitigation mechanism for IEEE 802.11e based CPS. We develop mathematical models for the performance of the proposed detector and mitigation mechanisms. Extensive simulation results show that the proposed mechanisms can achieve a high detection rate and punish a misbehaving node with a high packet dropping rate.**

***Index Terms*—Cyber physical system; IEEE 802.11e; misbehavior detection; mitigation; false positive rate; detection rate**

I. Introduction

Cyber physical systems (CPSs) are systems where cyber and physical subsystems interact intimately to provide ubiquitous data retrieving from and convenient control of physical environments. Integrated with the rapidly developing wireless and networking technologies, CPS are envisioned to facilitate intimate interactions between human and the physical world in a large variety of applications such as environment monitoring, building automation, transportation systems, industrial automation and smart grid [1]–[4].

The IEEE 802.11 based wireless local area networks (WLANs) are one of the most commonly used type of wireless networks for short-range communications. WLANs are suitable for monitoring/surveillance applications in various forms of CPS such as multi-media sensor networks, body area networks and cyber-physical vehicular ad hoc networks (VANETs) [5]–[7]. They also find rapidly growing applications in industrial control systems [8], [9], where the central control unit can act as the access point (AP) for receiving and processing sensor data. Advances in 802.11g and 802.11ah standard can support machine-to-machine (M2M) communications with little human intervention and provide efficient access scheme for wireless devices [10].

In CPS, the exchanged data are often of different types and may have different priorities. For example, in wireless networked control systems, there are sensor measurements, control decision packets and network management (e.g., time synchronization) messages that need to be exchanged among the control unit(s), actuators, sensors and other devices. In VANETs, there are traffic information (e.g., packets conveying road congestion and emergence accident information) and commercial advertisement packets that are transmitted in an ad hoc manner. For timely reacting to the accident, emergence packets should be given high priorities over other ones. The diverse priorities require service differentiation, which however is not supported by traditional IEEE 802.11 protocol. This problem is solved by the IEEE 802.11e standard which adopts the Enhanced Distributed Channel Access (EDCA) mechanism to provide media access control (MAC) level differentiation in quality of service (QoS) [11]. With EDCA, network traffic is prioritized and classified into several access categories (ACs). Service differentiation is realized by assigning different parameters for each AC, including the minimum and maximum contention window sizes (i.e., CWmin and CWmax, respectively), the arbitration inter-frame space (AIFS) number and transmission opportunity (TXOP) limit [11]. By default, four ACs are specified in the standard, namely background ($\mathrm{AC_{BK}}$), best effort ($\mathrm{AC_{BE}}$), video ($\mathrm{AC_{VI}}$) and voice ($\mathrm{AC_{VO}}$), in ascending priority order. IEEE 802.11e has been applied in industrial control and remote healthcare CPSs [12], [13]. In industrial control applications, it has been detailed in [14] that the following four types of data communications can be classified into the four default ACs in practice: urgent asynchronous notifications ($\mathrm{AC_{VO}}$), process data sent on a predictable schedule ($\mathrm{AC_{VI}}$), process data sent on a sporadic schedule ($\mathrm{AC_{BE}}$), and parameterization services ($\mathrm{AC_{BK}}$).

Recently, the security aspect of CPS has gained significant attention. As both communication and control decisions are carried out in the cyber space, CPS is substantially vulnerable to cyber-attacks [15]. As one of the most important security issues, misbehavior detection and prevention have been investigated in various CPS applications [16]–[19]. In IEEE 802.11e based CPS, to provide desired QoS, it is important to guarantee

Manuscript received February 13, 2015. Revised September 18, 2015. Accepted for publication October 10, 2015.

This work was supported in part by the National Natural Science Foundation of China under grants 61203036 and 61573103, and the NSF under grants CNS-1117687 and CNS-1320736.

X. Cao is with School of Automation, Southeast University, China. (E-mail: xhcao@seu.edu.cn).

L. Liu, W. Shen, A. Laha and Y. Cheng are with Department of Electrical and Computer Engineering, Illinois Institute of Technology, USA. (E-mails: {lliu41,wshen7,alaha}@hawk.iit.edu; cheng@iit.edu).

J. Tang is with AT&T Labs. E-mail: jin.tang@att.com.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

fairness of sharing the spectrum resource among the nodes in one access category. However, since the wireless media are inherently open-access, a selfish node can easily gain advantage over other normal nodes by deliberately manipulating its MAC parameters. For example, by using a shorter contention window, a node will spend less time in backoff state and can more frequently access the medium. Also, it can use a smaller arbitration inter-frame space (AIFS) to wait for shorter time than others in the same AC before accessing the medium. Thus, its throughput can be higher than expected. Moreover, the misbehaving node will also impose more interference to other ones and hence apply denial of service attacks to them to reduce their throughput. The impact of such misbehavior can be significant in CPS. For example, in networked control systems, if a misbehaving sensor sends abundant packets to the control unit, which in turn may reduce the successful transmission rates of the control decisions, the system may frequently become uncontrolled due to losses of those control decision packets.

For IEEE 802.11e protocol, due to the random access nature of the carrier sense and multiple access with collision avoidance (CSMA/CA) based MAC protocol, it is challenging to detect and mitigate the impact of such misbehavior. Usually, we need to monitor every node for a sufficiently long period of time to judge whether it is misbehaving or not. Since it is difficult to extract necessary information from collided transmissions, information conveyed in successful transmissions is one of the few sources of available information that can be utilized for detection. There have been detectors proposed in [20], [21] for misbehavior detection in 802.11 networks by exploiting the fair share property among nodes. However, with multiple ACs in an IEEE 802.11e WLAN, the network-wide fairness as achieved in traditional IEEE 802.11 based networks is broken [22], making the above detectors generally inapplicable. Moreover, existing 802.11e misbehavior detectors [23]–[25] suffer from long detection delay or the difficulty in correctly measuring desired values. Thus, efficient and real-time misbehavior mitigation mechanisms for CPS with QoS differentiation based on IEEE 802.11e are still open issues.

In this paper, we analyze the misbehavior strategy and its impact in IEEE 802.11e networks and propose both a novel real-time light-weight detector and mitigation mechanism to deal with misbehavior in IEEE 802.11e based CPS. The major contributions in this paper can be summarized as follows.

- We propose a mathematical model of the percentage of resource sharing for a node in each priority class. Based on this, we design a novel hybrid-share detector in which the detector keeps updating its state based on every successful transmission and makes detection decisions by comparing its state with a threshold. We also develop analytical results of the detector performance in terms of false positive rate and average detection rate.
- Further, we propose a light-weight misbehavior mitigation mechanism in which the AP randomly drops the packets of a node if the node is identified as misbehaving. The performance of the mechanism is also analyzed and the long-term average packet dropping rate is derived analytically.
- Finally, we present extensive simulation results to demonstrate the performance of the proposed detector and mitigation mechanism in terms of various performance metrics including detection rate, false positive rate, detection delay and packet dropping rate.

Some primary results have been published in [26]. In this paper, we have incorporated significant new contributions. Firstly, the work in [26] only focused on misbehavior detection, while this work further proposes a mitigation mechanism that allows the AP punish misbehaving nodes by randomly dropping its packets. As will be discussed, rational misbehaving nodes will finally be driven to well-behaving when such punishment is applied. Secondly, we develop analytical models for average packet dropping rate of the mitigation mechanism. Thirdly, we conduct extensive simulations based on the OMNeT++ network simulator and show that our analytical models have a good accuracy.

The remainder of this paper is organized as follows. More related work is reviewed in Section II. Section III overviews the problem. Following the mathematical MAC model in Section IV, the detector and mitigation mechanism are designed and analyzed in Section V and VI, respectively. Simulation results are presented in Section VII, while more discussion about the proposed mechanisms is provided in Section VIII. Concluding remarks are presented in Section IX.

II. Related work

Enhanced from the IEEE 802.11 protocol, the IEEE 802.11e protocol is a promising technology for many industrial applications [14]. There have been a number of recent studies on the performance of IEEE 802.11e based WLANs for industrial CPS when real-time communications are considered [9], [12]. However, the misbehavior impact and countermeasures in such systems have not been well studied in the literature.

Information conveyed in successful transmissions can be utilized for misbehavior detection in 802.11 based networks. Toledo *et al.* proposed to detect backoff misbehavior by checking whether the idle time between consecutive successful transmissions from a target node obeys the normal distribution [20]. Exploiting the fairness property across the network, a light-weight fair-share detector is design in [21], which does not rely on the idle time distribution. Instead, it counts the number of successful transmissions (or throughput) of each node and identifies a node if its throughput is much higher than others. However these detectors are not inapplicable in an IEEE 802.11e WLAN, since the network-wide fairness cannot be achieved in the service differentiation scenarios.

To detect backoff misbehavior in IEEE 802.11e networks, Szott *et al.* proposed a χ^2 detector by comparing the measured and expected backoff values [23]. However, the exact values of backoff periods followed by unsuccessful transmissions may be hard to measure. The detector in [24], however, takes advantage of the fact that the interval between two consecutive successful transmissions is uniformly distributed in [0, CWmin) providing that the packet in the second transmission was not retransmitted before. Nevertheless, the detector delay could be very high. In addition, the information whether a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

packet is retransmitted or not is difficult to obtain or can be manipulated by the misbehavior to cheat the detector. While there are works well addressed the TXOP misbehavior [25], efficient and real-time detection of contention window and AIFS misbehavior still remains open.

III. Problem Overview

To support QoS differentiation in CPS, the IEEE 802.11e protocol defines a heterogeneous network operating environment as detailed as follows.

A. IEEE 802.11e Protocol

In traditional IEEE 802.11 networks, the channel access among nodes is coordinated by the distributed coordination function (DCF) based on the CSMA/CA mechanism. Before transmission, each node should first sense the channel idle for an inter-frame space time (DIFS) and then wait until a backoff timer counts down to 0 before starting transmission. Each node takes the binary exponential backoff strategy to access the channel with the backoff timer at each backoff stage initialized at a value randomly chosen from $[0, CW-1]$. Here, the contention window size CW is initialized at CWmin and is doubled (until CWmax) once a transmission is unsuccessful (thus the backoff stage increases by 1). A packet will be retransmitted at most for a certain number of times. The backoff timer counts down by 1 if the channel is sensed idle for a backoff slot. Otherwise if the channel is sensed busy, the timer will be suspended until the channel becomes idle for DIFS time again. Since all nodes use the same parameters, the channel contention is fair for them.

However, the EDCA specification in IEEE 802.11e supports hybrid backoff parameters and AIFS. With EDCA, a node with a small contention window size can gain a high opportunity to win the channel access contention game among other nodes, thus achieving a high throughput. However, a node with a large AIFS should wait a longer time before starting channel access contention, thus giving way to high-priority nodes. By default, there are four priority classes (or access categories) defined in IEEE 802.11e EDCA [11], as shown in Table I.

TABLE I
EDCA DEFAULT SETTINGS.

Access category	CWmin	CWmax	AIFSN
Background AC_{BK}	aCWmin	aCWmax	7
Best Effort AC_{BE}	aCWmin	aCWmax	3
Video AC_{VI}	(aCWmin+1)/2-1	aCWmin	2
Voice AC_{VO}	(aCWmin+1)/4-1	(aCWmin+1)/2-1	2

In this paper, we consider the general cases that there are c priority classes, each of which is assigned with contention window size parameters CWmin_i and CWmax_i, and inter-frame space AIFS_i=AIFSN_i*aSlotTime+aSIFSTime, where AIFSN is the number of slots, after a short inter-frame space duration, a node should defer before either invoking a backoff or starting a transmission. The parameters are assigned by the AP.

Normally, after a packet transmission, each node should conduct a new channel contention process in order to transmit the next packet. However, the IEEE 802.11e offers the TXOP option to allow a node retaining the channel access and continuing transmitting packets after a successful transmission. The duration of such TXOP time is controlled by the TXOP Limit parameter.

B. Misbehavior Model and Analysis

A misbehaving node may use MAC parameters different from those assigned by the AP, to gain a higher share of the resource. As discussed above, in order to improve its throughput, a misbehaving node can manipulate its CWmin, CWmax and AIFSN to be smaller (or the TXOP Limit to be larger) than the corresponding values assigned by the AP. It can also completely disobey the rules, e.g., by using a fixed backoff window size other than the binary exponential backoff strategy. In the following, we assume that TXOPLimit$= 0$ and that TXOP misbehavior does not present. We defer the discuss about relaxing these assumptions and the detection of TXOP misbehavior to Section VIII-A.

To demonstrate the impact of the misbehavior, consider an example network of 10 normal nodes and one misbehaving node. Each node is saturated, that is, it always has packets to transmit. Each normal node uses the following MAC parameters: $\text{CWmin} = 15, \text{CWmax} = 1023$ and $\text{AIFSN} = 2$; while the misbehaving node takes $\text{CWmin} = 1 \sim 32, \text{CWmax} = 1023$ and $\text{AIFSN} = 0 \sim 2$. Fig. 1 illustrates the impact of the misbehaving node in terms of the percentages of resource sharing (defined as the ratio of the throughput contribution from a particular node over the total network throughput). As shown in this figure, the misbehaving node can gain significant advantage over the other nodes by manipulating its MAC parameters. Moreover, the impacts of CWmin and AIFSN are different. For example, in order to achieve a 10% higher throughput, the misbehaving node needs to reduce its CWmin to a much smaller value (e.g., from 15 to less than 7); while, this can also be achieved by simply reducing its AIFSN from 2 to 1. In other words, the misbehaving impact on the network is more sensitive to AIFSN than CWmin.

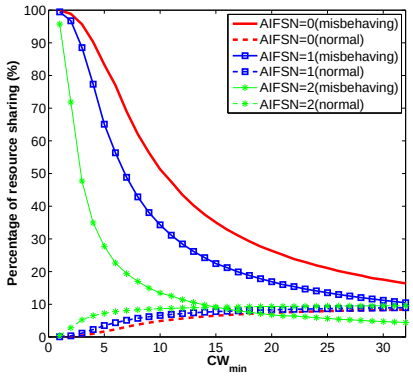


Fig. 1. Impact of MAC misbehavior.

Because the backoff size is chosen randomly, it is difficult to tell a contention window- or AIFS- misbehaving node from other normal nodes based on analyzing a single or very few transmissions. For instance, a misbehaving node with CWmin=7 may generate the same backoff window size as another normal one with CWmin=15. Also, since the AIFS duration often comes before or after the backoff, an AIFS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

misbehaving node is able to pretend to be a normal one that chooses small backoff sizes. In particular, a node can dynamically switch between misbehaving and being normal. Therefore, to detect such misbehavior, it requires to monitor the packet transmissions of each node for a long time.

In this paper, we focus on both CWmin and AIFSN manipulating misbehavior. While the proposed detector is also effective for CWmax misbehavior, a misbehaving node may prefer to manipulate CWmin over CWmax since the former strategy has greater impact. Also, we focus on saturated traffic cases; otherwise, for a light-loaded network with unsaturated traffic, since the medium is less likely to be crowded, a misbehaving node may not have much impact on the other normal nodes. Our goal is to detect such misbehavior at the AP and counteract with it in real-time.

C. Fair-Share Detector and Challenges

Consider an IEEE 802.11e based CPS with one AP and n nodes located within each other's communication range. The nodes compete for sending packets to the AP. In traditional IEEE 802.11 standard, the DCF mechanism guarantees that on average each node will share the same portion of the medium resource and maintains fairness across the network. For an arbitrary node v, let a binary variable I_v be the indicator of whether a packet received by the AP is from node v or not. In normal cases, due to the network-wide fairness guarantee, we have probability $\mathbb{P}[I_v = 1] = \frac{1}{n}$.

A misbehaving node can gain unfair share of the resource by manipulating its backoff parameters, e.g., using a smaller CWmin. If the AP records all the received packets, it can notice that more packets are from the selfish one, i.e., $\mathbb{P}[I_v = 1] > \frac{1}{n}$. The work in [21] takes advantage of this feature and designs a nonparametric cumulative sum (CUSUM) based fair-share misbehavior detector (called FS detector) to detect such misbehavior in real-time, which is described as follows.

For a target node v, let X_k be the state of the detector for v. X_k initializes at 0, i.e., $X_0 = 0$. For the kth packet received by the AP, the state of the detector is updated as follows.

$$X_{k+1} = [X_k + (nI_k - 1)]^+, \tag{1}$$

where x^+ is x if $x > 0$ and is 0 if otherwise. If the packet is from v, we have $I_k = 1$ and $X_{k+1} = X_k + n - 1$; otherwise, $I_k = 0$ and $X_{k+1} = X_k - 1$. The idea behind is that, due to fair sharing, the nodes roughly take turns to transmit packets. Therefore, the detector state X_k is likely to be bounded. In presence of misbehaving nodes, since $\mathbb{P}[I_v = 1] > \frac{1}{n}$, the unfair portion of channel sharing will accumulate such that the state of the FS detector associated with each misbehaving node finally becomes unbounded. Thus, a detection threshold h is employed to decide whether v is misbehaving (i.e., $\delta_k = 1$) or not (i.e., $\delta_k = 0$) as follows.

$$\delta_k = \begin{cases} 1, & \text{if } X_k \geq h, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Because every received packet is counted by the AP in making detection decisions, the FS detector can identify the misbehaving node much faster than many existing detectors. Moreover, the FS detector is nonparametric and lightweight in terms of computation complexity, thereby it is able to provide real-time misbehavior detection services [21].

Since the underlying assumption of network-wide fairness is broken in the EDCA situation, the FS detector cannot be directly applied in networks with hybrid priority classes. However, since sub-network fairness still can be achieved among the nodes in the same class, a natural extension is to design a distinct FS detector for each class. Specifically, for a node in class i, the associated detector should use the number of nodes in this class other than a common n of the whole network, as in (1). Nevertheless, such an extended FS detector encounters some challenges. For example,

- If there is only one node in a class, its misbehavior may not be detected. To see this, substituting $n = 1$ into (1) we can obtain that $X_{k+1} = [X_k - (1 - I_k)]^+ \equiv 0$ if $X_0 = 0$. As a result, $\delta_k \equiv 0$.
- If all the nodes in one class misbehave, some or all of them may not be detected. Specifically, if they use the same manipulated MAC parameters, the detector sees that none of them is misbehaving; otherwise, at least the one with the lowest throughput among them will be considered as a normal node.

Therefore, to overcome the shortcomings of the FS detector, only considering a single priority class is not enough. In the following, we propose a novel hybrid-share detector based on the following analytical MAC model.

IV. MAC Analytical Model

For an arbitrary node v in class i, denote s_i as its percentage of resource sharing. In this section, we propose an analytical model for calculating s_i. We assume that there are c priority classes; each class (say class i) contains n_i nodes that compete for channel access using parameters $W_i = \text{CWmin}_i$, $\text{CWmax}_i = 2^{m_i}(W_i + 1) - 1$ and AIFS_i, where $1 \leq i \leq c$ and $m_i = \log_2 \frac{\text{CWmax}_i + 1}{\text{CWmin}_i + 1}$ is the maximum backoff stage. For simplicity, we assume that the maximum retransmission limit for node v is the same as m_i[1]. Thus, the contention window size of this node in its jth backoff stage is

$$W_{i,j} = 2^j(W_i + 1) - 1. \tag{3}$$

Assume that all nodes including the AP can hear each other's transmissions. No capture effect is assumed, thus a transmission is successful only if no other transmissions happen simultaneously. Let p_i and τ_i be the frame blocking probability (i.e., the probability that node v senses a busy channel (and thereby suspends its backoff timer countdown) in a generic time slot) and transmitting probability (i.e., the probability that v transmits in a generic slot), respectively. The

[1]This assumption can be relaxed to obtain a more complicated version of the MAC model; however, the modeling methodology is similar and the hybrid-share detector proposed in this paper will be still valid.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

transmitting probability has been given in [27] as follows:

$$\begin{aligned}\tau_i &= \frac{1-p_i^{m_i+1}}{(1-p_i)\sum_{j=0}^{m_i} p_i^j \left[1+\frac{1}{1-p_i}\sum_{k=1}^{W_{i,j}} \frac{W_{i,j}-k}{W_{i,j}}\right]} \\ &= \frac{1-p_i^{m_i+1}}{\sum_{j=0}^{m_i} p_i^j \left(1-p_i+\frac{W_i}{2}\right)} \\ &= \frac{2(1-p_i)(1-2p_i)}{(1-2p_i)^2+(W_i+1)(1-p_i)\frac{1-(2p_i)^{m_i+1}}{1-p_i^{m_i+1}}}. \end{aligned} \quad (4)$$

Let $\Delta A_i = \text{AIFSN}_i - \text{AIFSN}_{\min}$, where $\text{AIFSN}_{\min} = \min\{\text{AIFSN}_j | j = 1, \ldots, c\}$. Due to the differentiation in AIFS, a node of low priority must wait for a longer idle time than a high-priority node after a busy channel period before resuming its backoff timer countdown. That is, for node v, it has to sense $\Delta A_i + 1$ idle slots before resuming backoff timer countdown [28]. Therefore, the frame blocking probability of node v is equivalent to the probability that none of $\Delta A_i + 1$ slots is occupied by transmissions from other nodes, i.e.,

$$p_i = 1 - \left(\frac{1-p_b}{1-\tau_i}\right)^{\Delta A_i+1}, \quad (5)$$

where p_b is the probability that the channel is busy in a random slot, which can be easily measured by the AP. By definition, p_b can be given as follows.

$$p_b = 1 - \prod_{k=1}^{c} (1-\tau_k)^{n_k}. \quad (6)$$

With the equations (4)-(6), the AP can solve the probabilities τ_i and p_i numerically. In the special case that $m_i = 0$, the AP simply gets $\tau_i = \frac{2}{W_i+1}$ and $p_i = 1 - \frac{W_1+1}{W_i-1}(1-p_b)$.

In a generic slot, the probability that node v successfully transmits a packet to the AP is

$$\begin{aligned} p_{s,i} &= \tau_i (1-\tau_i)^{n_i-1} \prod_{k=1, k\neq i}^{c} (1-\tau_k)^{n_k} \\ &= \frac{\tau_i}{1-\tau_i}(1-p_b). \end{aligned} \quad (7)$$

Therefore, the percentage of resource sharing of node v (which is also the probability that a successful transmission to the AP is from v) is given by

$$s_i = \frac{p_{s,i}}{p_s} = \frac{p_{s,i}}{\sum_{k=1}^{c} n_k p_{s,k}} = \frac{\frac{\tau_i}{1-\tau_i}}{\sum_{k=1}^{c} \frac{n_k \tau_k}{1-\tau_k}}, \quad (8)$$

where $p_s = \sum_{k=1}^{c} n_k p_{s,k}$ is the probability of a successful transmission (from any node).

We can further obtain the total throughput η, i.e., the average number of packets (from any node) received by the AP in one slot, and the throughput of the target node v as

$$\begin{aligned} \eta &= \frac{\text{Probability of a successful transmission}}{\text{Average length of a slot time}} \\ &= \frac{p_s}{1-p_b+p_s T_s+(p_b-p_s)T_c}, \end{aligned} \quad (9)$$

$$\eta_i = s_i \eta, \quad (10)$$

respectively, where $p_b - p_s$ is the probability of a collided transmission. $1 - p_b$ is the channel idle probability, i.e., the probability that none of the nodes transmits. T_s and T_c are the numbers of empty slots[2] of a successful transmission and a collision, respectively. In the case of basic access (without RTS/CTS handshaking), we have [28], [29]:

$$T_s = \text{AIFSN}_{\min} + L + 2L_{\text{SIFS}} + L_{\text{ACK}} + 2\delta, \quad (11)$$

$$T_c = \text{AIFSN}_{\min} + L + L_{\text{SIFS}} + L_{\text{ACK}} + \delta, \quad (12)$$

where L is the length of a packet including the MAC and PHY headers[3]. In the above, L_{SIFS} and L_{ACK} are durations of a short inter-frame space and an ACK transmission period, respectively. δ represents the propagation delay. The units of both T_s and T_c are numbers of empty slots. Then, the average number of empty slots between two successive transmissions can be given by

$$T = \frac{1}{\eta}. \quad (13)$$

Note that, if each misbehaving node is treated as a distinct priority class, the above analysis is able to accommodate both normal and misbehaving nodes.

V. Hyrid-Share Detector Design

A. HS Detector Design

The AP runs a distinct detector algorithm for each node. Specifically, for a target node belonging to priority class i, the hybrid-share (or HS for short) detector is designed as follows. Based on the above analytical model, the numerical solution of s_i may introduce some error, say ϵ_i. Let $\bar{s}_i$ be the approximated solution of s_i, i.e.,

$$s_i = \bar{s}_i + \epsilon_i, \quad (14)$$

where ϵ_i accounts for the approximation error. The purpose of using $\bar{s}_i$ instead of s_i will be discussed in Remark 1.

In the following, we shall omit the subscript i. The detector maintains a state X_k with initial state $X_0 = 0$. Once a packet arrives at the AP, the detector state is updated according to

$$X_{k+1} = \min\left\{\left[X_k + (I_k - \bar{s})\right]^+, \bar{m}\sigma\right\}, \quad (15)$$

where I_k is defined below Eq. (1) and $\mathbb{P}[I_k = 1] = s$. $\bar{m}$ and σ will be defined later in Section V-B1. Once X_k hits its maximal value (i.e., $\bar{m}\sigma$), it is reset to 0 immediately, which results in that $X_{k+1} = (I_k - \bar{s})^+$. With such resettings, the detector is frequently refreshed to be prepared for detecting real-time misbehavior.

In normal cases, X_k is expected to remain in $[0, 1]$. We introduce a new detection threshold h and make the detection decision by computing

$$\delta_k = \begin{cases} 1, & \text{if } X_k \geq h, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

Similar as above, $\delta_k = 1$ indicates that the target node is misbehaving, while $\delta_k = 0$ indicates normal behaving.

Note that, in normal cases, X_k may be able to hit 1 if the AP receives a number of successive packets from the target

[2]An empty slot is specified by parameter aSlotTime in the standard [11].

[3]We assume all the packets are of the same length. The case with diverse packet lengths is discussed in [29].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

node. Therefore, for the sake of correct detection, h can be set to larger than 1. More discussion about selecting h will be provided later in Section VII.

We call X_k the state of the HS detector in step k. Note that the step size may vary from time to time because the packet arrivals at the AP are generally random. However, from (13) we can obtain the average step size as T.

When applying the proposed detector, the AP only needs to compute the MAC model and calculate the percentage of resource sharing once, as long as the network configuration and MAC parameters assigned to each node do not change. As shown in (15), the computation complexity of the proposed detector itself is very low. Therefore, it is worth noting that the proposed detector is light-weight. Moreover, since all received packets are utilized by the detector, misbehavior can be detected in a real-time manner.

B. Performance Analysis

Definition *1:* To evaluate the performance of the HS detector, we define the following metrics.

- The *false positive rate* p_f is the conditional probability that the target node is indicated misbehaving when in effect none of the nodes is misbehaving.
- The *detection rate* $p_d(D)$ of the HS detector is the probability that a misbehaving node will be detected within D time slots (slots are defined in IEEE 802.11e).

p_f can be viewed as the rate of false alarms, while $p_d(D)$ reflects the effectiveness and real-time performance of the HS detector. Below we analyze the detector performance by mathematically modeling p_f and $p_d(D)$.

1) False positive rate: For any $\bar{s} > 0$, there exists $\sigma > 0$ such that both $\frac{\bar{s}}{\sigma}$ and $\frac{1-\bar{s}}{\sigma}$ are integers (say L_0 and L_1, respectively). For example, we can use the precision of $\bar{s}$ to determine the above two integers. For any step k between two adjacent detector state resettings, suppose there are k_1 times that $I_\kappa = 1$ and k_0 times that $I_\kappa = 0$, where κ is between the last resetting step and k. Thus, based on (15), $X_k \in \{0, X_{k-1}-\bar{s}, X_{k-1}+1-\bar{s}\}$. Furthermore, $X_k \le k_1(1-\bar{s}) = k_1 L_1 \sigma$, which yields that

$$X_k \in \{0, \sigma, 2\sigma, \ldots, k_1 L_1 \sigma\}. \tag{17}$$

Since X_k is a set of multiples of σ, its largest possible value is $\bar{m}\sigma$ where $\bar{m} = \lceil \frac{h}{\sigma} \rceil$ (otherwise X_k is reset). Therefore, the support of X_k can be denoted as

$$\begin{aligned}\mathcal{M} &= \left\{0, m_1\sigma, m_2\sigma, \ldots, \bar{m}\sigma \middle| m_j \in \mathbb{N}^+, m_j < m_{j+1} < \bar{m}\right\} \\ &\subseteq \{0, \sigma, 2\sigma, \ldots, \bar{m}\sigma\}.\end{aligned} \tag{18}$$

Clearly, $\mathcal{M}$ is a finite set.

According to (15), X_{k+1} depends only on X_k, thus the sequence $\{X_k\}$ forms a homogeneous Markov chain. Since the support of X_k may vary from step to step, to calculate the probabilities of the chain's states at any step k, we can consider the bigger set $\{0, \sigma, 2\sigma, \ldots, \bar{m}\sigma\}$ without loss of generality. Define the following vector:

$$\boldsymbol{x}_k = \left(\mathbb{P}[X_k = 0],\ \mathbb{P}[X_k = \sigma],\ \ldots,\ \mathbb{P}[X_k = \bar{m}\sigma]\right), \tag{19}$$

Obviously, $\boldsymbol{x}_k \mathbf{1} = 1$, where $\mathbf{1}$ is the $(\bar{m}+1)\times 1$ vector with all elements equal to 1. Initially, we let $\boldsymbol{x}_k = [1, 0, \ldots, 0]$ which corresponds to $X_0 = 0$. Due to the homogeneity of the chain, we can have $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k \mathbf{P}$, where $\mathbf{P}$ is the step-independent probability transition matrix. $\mathbf{P}$ depends only on s and thus can be represented as $\mathbf{P}(s)$. Let $P_{i,j}$ be the $(i+1, j+1)$th entry of $\mathbf{P}$, which is the probability of the transition from $x_{i,k}$ to $x_{j,k+1}$, $\forall k \ge 0$. To calculate $\mathbf{P}$, consider the following scenarios of state transitions from $X_k = i$ to $X_{k+1} = j$:

- If $i = 0$, according to (15), j can only be either $1-\bar{s}$ (if $I_k = 1$) or 0 (if $I_k = 0$). Therefore,

$$P_{0,L_1} = s \quad \text{and} \quad P_{0,0} = 1 - s. \tag{20}$$

- If $i \in (0, L_0]$, j will become 0 when $I_k = 0$. Therefore,

$$P_{i,0} = 1 - s \quad \text{and} \quad P_{i,\min\{\bar{m}, i+L_1\}} = s. \tag{21}$$

- If $i \in (L_0, \bar{m})$, (15) reduces to $X_{k+1} = \max\{X_k + (I_k - \bar{s}), \bar{m}\sigma\}$. Therefore,

$$P_{i,i-L_0} = 1 - s \quad \text{and} \quad P_{i,\min\{\bar{m}, i+L_1\}} = s. \tag{22}$$

- Otherwise, $i = \bar{m}$. Since X_k is reset to 0 immediately when it reaches $\bar{m}$, this scenario is similar to the first one (i.e., when $i = 0$). Thus,

$$P_{\bar{m},L_1} = s \quad \text{and} \quad P_{\bar{m},0} = 1 - s. \tag{23}$$

Then, with the initial value $\boldsymbol{x}_0$, all $\boldsymbol{x}_k$ can be calculated by iterating $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k \mathbf{P}$. Let us consider the steady-state distribution of the chain $\{X_k\}$: $\boldsymbol{\pi} = \lim_{k\to\infty} \boldsymbol{x}_k = [\pi_0, \pi_1, \ldots, \pi_{\bar{m}}]$. With $\mathbf{P}$ given as above, we can get a unique $\boldsymbol{\pi}$ by solving $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$. Then, based on Definition 1, the false positive rate is given by

$$p_f = \pi_{\bar{m}}. \tag{24}$$

Remark *1:* In the above, the value of σ directly determines the dimensions of $\boldsymbol{P}$ and $\boldsymbol{\pi}$. For the approximated value $\bar{s}$ with less precision, we can get a relatively larger σ and hence lower computation complexity in obtaining $\boldsymbol{\pi}$. Simulation results in Section VII show that a precision of 0.1 can already achieve a satisfactory false positive rate.

2) Average Detection Rate: Suppose the target node starts to misbehave at step 0 and that the associated Markov chain $\{X_k\}$ in the normal case before step 0 has reached its steady distribution $\boldsymbol{\pi}$. Note that, with the target node misbehaving, the parameters (e.g., CWmin and AIFSN) of the MAC model change. Hence, we add superscript * to the variables defined in previous sections to distinguish the case with the target node misbehaving from the normal case. Since whether a node misbehaves or not is not pre-known to the detector, it shall assume that the target node is normal and still use $\bar{s}$ to update its state. Thus, the support of X_k (and also σ, L_0, L_1 and $\bar{m}$) remains the same as before. The only difference lies in the probability of $I_k = 1$ (i.e., $\mathbb{P}[I_k = 1] = s^*$), which in turn changes the probability transition matrix from $\mathbf{P}(s)$ to $\mathbf{P}^* = \mathbf{P}(s^*)$. Let $\boldsymbol{\pi}^*$ be the stationary probability associated with $\mathbf{P}^*$, i.e., $\boldsymbol{\pi}^* = \boldsymbol{\pi}^*\mathbf{P}^*$. The following theorem derives the detection rate for which the proof is presented in Appendix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

Theorem *1:* The average detection rate is:

$$p_d(D) = \lfloor \frac{D}{T^*} \rfloor \pi^*_{\bar{m}}, \tag{25}$$

where $\lfloor \frac{D}{T^*} \rfloor$ is the average number of steps in time D (referring to (13)). $\pi^*_{\bar{m}}$ is the last element of $\boldsymbol{\pi}^*$.

VI. Design of A Mitigation Mechanism

Once a node is detected as misbehaving, the AP can punish the node by dropping its subsequent packets. Since our detector may yield false positive rate, it is possible that a normal node is mistakenly found as a misbehaving one. To prevent the packets of a normal node from being blocked by the AP, once a node is deemed as misbehaving, its subsequent packets are not completely dropped. Instead, the node will be blacklisted and its subsequent packets will be dropped with probability determined as follows:

$$\rho_k(Y_k) = \min\{1, (Y_k)^+\}, \tag{26}$$

where Y_k is another state associated with the target node and is induced by the detector state X_k. Y_k is initialized at 0 and evolves as follows.

$$Y_k = \max\{\min\{Y_{k-1} - \alpha 1_{X_k<h} + \beta 1_{X_k \geq h}, Y_{\sup}\}, Y_{\inf}\}, \tag{27}$$

In the above, α, β, $Y_{\sup}$ and $Y_{\inf}$ are real numbers. Particularly, $Y_{\inf} \leq 0 < \alpha < \beta < 1 \leq Y_{\sup}$. $1_{\text{condition}}$ equals to 1 if "condition" is true and is 0 if otherwise. Without loss of generality, assume that there exists $\tilde{\sigma} > 0$ such that $\alpha = n_\alpha \tilde{\sigma}$, $\beta = n_\beta \tilde{\sigma}$, $1 = n_1 \tilde{\sigma}$, $Y_{\sup} = n_{\sup} \tilde{\sigma}$ and $Y_{\inf} = n_{\inf} \tilde{\sigma}$, where n_α, n_β, n_1, $n_{\sup}$ and $n_{\inf}$ are integers. Thus, $Y_k \in \mathcal{Y} \triangleq \{n_{\inf}\tilde{\sigma}, (n_{\inf}+1)\tilde{\sigma}, \ldots, 0, \tilde{\sigma}, 2\tilde{\sigma}, \ldots, n_{\sup}\tilde{\sigma}\}$.

The basic idea of the above misbehavior mitigation mechanism is to gradually increase the dropping rate of a misbehaving node and reduce that of a normal node. Specially, as seen from (27), the state Y_k will increase by β once the detector state X_k hits h. Thus, Y_k will grow towards $Y_{\sup}$ and the packet dropping probability will grow to 1 if the detector state of the corresponding node frequently hits the upper bound h. Otherwise, if the state X_k remains within $[0, h)$, Y_k will gradually decrease, and once it reduces to or below 0, the packet dropping rate will become 0; in this case, the target node will be deemed as a normal one and will be removed from the blacklist of the AP.

A. Performance Analysis

In the following, we analyze the performance of the proposed countermeasure on the target node. Similar to $\boldsymbol{x}_k$, define

$$\boldsymbol{y}_k = \left(\mathbb{P}[Y_k = n_{\inf}\tilde{\sigma}], \ldots, \mathbb{P}[Y_k = n_{\sup}\tilde{\sigma}]\right). \tag{28}$$

By (27), we know that $\{Y_k\}$ forms a Markov chain. Let the probability transition matrix be $\boldsymbol{T}_k$, i.e., $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k \boldsymbol{T}_k$. We let $T_{i,j,k}$ denote the (i,j)-th entry of $\boldsymbol{T}_k$, which is the transition probability from $Y_{k-1} = (i-1+n_{\inf})\tilde{\sigma}$ to $Y_k = (j-1+n_{\inf})\tilde{\sigma}$. Apparently, $\boldsymbol{T}_k$ depends on $\boldsymbol{x}_k$, i.e., the probability vector for X_k. We have

$$T_{i,j,k} = \begin{cases} 1 - x_{\bar{m},k}, & \text{if } j = \max\{1, i - n_\alpha\}, \\ x_{\bar{m},k}, & \text{if } j = \min\{n_{\sup} - n_{\inf} + 1, i + n_\beta\}, \\ 0, & \text{otherwise}, \end{cases} \tag{29}$$

where $x_{\bar{m},k}$ is the last element of $\boldsymbol{x}_k$ which is calculated in Section V-B1. In the above equation, the first and second lines account for the cases that $X_k < h$ and $X_k \geq h$, respectively. $\boldsymbol{y}_k$ is initially $[1, 0, \ldots, 0]$. In a long-run, since the steady state of $\boldsymbol{x}_k$ will be $\boldsymbol{\pi}$, we have $\lim_{k\to\infty} \boldsymbol{T}_k = \boldsymbol{T}$, where $\boldsymbol{T}$ can be obtained by replacing $x_{\bar{m},k}$ with $\pi_{\bar{m}}$ in (29). Furthermore, $\boldsymbol{y}_k$ will evolve to its steady state $\boldsymbol{y}$ such that $\boldsymbol{y} = \boldsymbol{y}\boldsymbol{T}$. With $\boldsymbol{T}$ obtained as above, $\boldsymbol{y}$ can be calculated and the long-term average packet dropping rate for the target node is

$$\begin{aligned} \bar{\rho} &= \sum_{i=1}^{n_{\sup}-n_{\inf}+1} y_i \rho\left((i-1+n_{\inf})\tilde{\sigma}\right) \\ &= \sum_{i=2-n_{\inf}}^{n_1+1-n_{\inf}} (i-1+n_{\inf})\tilde{\sigma} y_i + \sum_{i=n_1+2-n_{\inf}}^{n_{\sup}-n_{\inf}+1} y_i, \end{aligned} \tag{30}$$

where the second equality holds because of (26). Note that, for both a misbehaving node and a normal node, the long-term average packet dropping rate is calculated in the same way as above.

Taking packet dropping into account, the throughput that the target node achieves becomes

$$\bar{\eta}_i = \eta_i(1 - \bar{\rho}), \tag{31}$$

where η_i is the throughput without the above countermeasure which is given in (10).

Remark *2:* If the target node keeps misbehaving, the throughput is given as $\bar{\eta}^*_i$. Otherwise, if it turns to well-behave, the throughput is $\bar{\eta}_i$. Therefore, the parameters of the proposed mechanism in (27) should be chosen such that

$$\bar{\eta}^*_i < \bar{\eta}_i, \tag{32}$$

which means that the target node will obtain no benefit by misbehaving; rather, its performance in terms of throughput will get even worse due to misbehaving. In this sense, a rational misbehaving node will decide not to always misbehave. Since the set of misbehaving strategies is finite, the above inequality should be ensured under the most light-weight misbehaving strategy.

Remark *3:* We suppose that the AP will warn the target node with a short message about the packet dropping probability ρ_k, if $\rho_k > 0$. Thus, with probability

$$\mathbb{P}[\rho_k > 0] = 1 - \sum_{i=1}^{1-n_{\inf}} y_{i,k}, \tag{33}$$

a misbehaving node will be informed that its packets will be dropped randomly. For a rational node, it will hence degrade its misbehaving intensity by increasing $\text{CW}_{\min}$ or the AIFSN. Thus, the proposed mechanism is potentially able to finally drive the misbehaving node to well-behaving.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

Otherwise, an irrational malicious node may keep misbehaving and ignore the warning packets from the AP. In this case, the detector state X_k will frequently hit h since it sends more packets than expected. Thus, its associated state Y_k will gradually increase to Y_{sup}, resulting in that the packet drop rate $\rho = 1$ (notice that $Y_{\text{sup}} \geq 1$). In other words, irrational misbehaving nodes will finally get blocked by the AP if it does not respond to the AP's warning.

VII. Performance Evaluation

In this section, we evaluate the performance of the HS detector and the proposed countermeasure through extensive computer simulations based on the OMNeT++ network simulator. As shown in Fig. 2, we consider a CPS with 15 wireless sensors (e.g., WiFi cameras, flowmeters and thermometers) and one AP. The sensors transmit data to the AP using the IEEE 802.11e protocol. The AP then transmits the sensors data to a programmable logic controller (PLC) for control purposes. The sensors are categorized into three classes. In class 1, there are $n_1 = 6$ nodes using MAC parameters $W_1 = 31$, $\text{CWmax}_1 = 1023$ and $\text{AIFSN}_1 = 3$. For the other two classes, we set $n_2 = 6$, $W_2 = 15$, $\text{AIFSN}_2 = 3$, $n_3 = 3$, $W_3 = 15$, and $\text{AIFSN}_3 = 2$. The data of class 3 sensors has the highest priority, so they can contain urgent information such as emergence notifications. The sensors in the other two classes can include normal process measurements data in their packets. CWmax is fixed at 1023 for all nodes. Other parameters used in the simulations are listed in Table. II. Each simulation runs for 60 seconds. There is one node (the target node) in class 2 which misbehaves. If the target node conducts misbehavior aggressively (e.g., using a very small CWmin), the data from other sensors may experience severe packet collisions or delay, which may further results in bad control performance of the whole CPS due to lack of data from other sensors. Therefore, for the health of the CPS, it is important to detect and mitigate the impact of the misbehaving node.

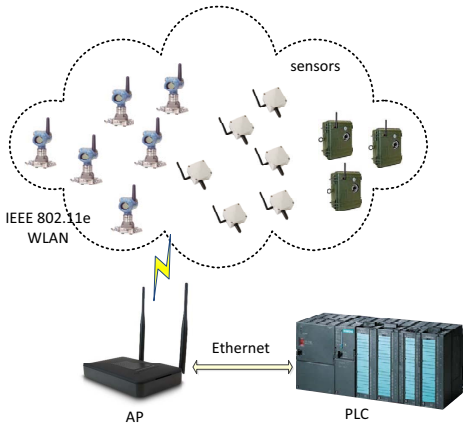


Fig. 2. Illustration of the simulating CPS. All the sensors report data to the AP based on the IEEE 802.11e protocol.

A. Performance of the detector

1) False positive rate: To evaluate the false positive rate p_f of the HS detector, we consider the case that the target node well-behaves. First, as shown in Fig. 3, our analytical results (calculated based on the proposed MAC model and the analytical performance model in Section V-B1) are accurate as compared with the simulation results. As also can be observed, p_f decreases as the detection threshold h increases. This is simply because the higher h is, the less opportunity that the detector state X_k will hit its maximal value (i.e., $\bar{m}\sigma$ as in (19)). The figure also shows that the numerical solution error of the analytical model, as indicated by ϵ in (14), has an impact on the rate p_f: a smaller error can result in lower false positive rate. However, since the curves are very close, we can see that a precision of 0.1 is enough to deliver satisfactory results.

TABLE II
Simulation Parameters

Parameter	value
deployment area	80m×80m square area
channel frequency band	2.4 GHz
bit rate	11 Mbps (IEEE 802.11b)
packet length	100 byte (same for all nodes)
transmit power	20.0 mW (same for all nodes)
SINR threshold	4 dB

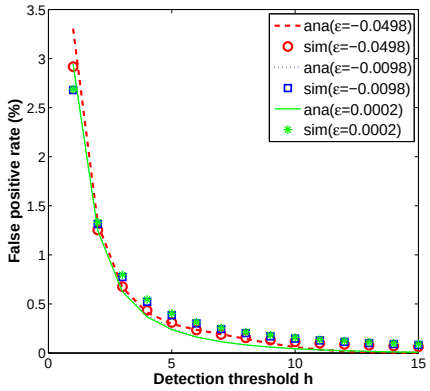


Fig. 3. False positive rate p_f under various detection thresholds. In the figure, the error $\epsilon = -0.0498, -0.0098$ and 0.0002 correspond to that $\sigma = 0.1, 0.02$ and 0.01, respectively.

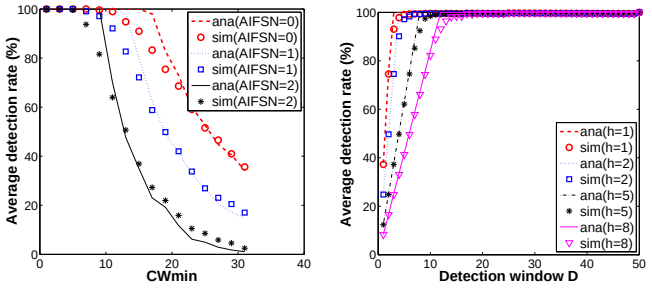
2) Detection rate: We then evaluate the detection rate of the HS detector by considering the misbehaving node with various misbehaving strategies. Again, the results shown in Fig. 4 confirm that our analytical models are of high accuracy. Fig. 4(a) shows the average detection rate $p_d(D)$ under different misbehaving intensities, where we fix $D = 20$[4] and $h = 5$. As the misbehavior is becoming intensified (i.e., the target node uses a smaller AIFSN and/or CWmin), more packets as received by the AP are from the target node. Hence, the detector state increases more frequently and is more likely to hit its maximal value. As a result, the average detection rate increases, which is clearly depicted in this figure.

Fig. 4(b) shows the performance of the HS detector associated with the target node under different D and h, where the misbehaving strategy is $\text{CWmin} = 7$ and $\text{AIFSN} = 0$. As can be observed from this figure, in all misbehaving cases, the detector becomes more reliable with a higher detection rate as the detection window D gets longer. The misbehavior will be

[4]For ease of exposition, we use D to stand for DT^*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

captured almost surely when D is larger than 30. However, a larger D indicates a longer detection delay. In this sense, we should keep D small in order to detect real-time misbehavior.



(a) Under different misbehavior intensities (b) With different detection window D

Fig. 4. Average detection rate $p_d(D)$.

It is clear that the detector performance is affected by the value of the parameter h. For the similar reason as we discussed above about Fig. 3, when h increases, the detector state associated with a target node becomes less frequently to hit h. This means that the average detection rate will decrease (i.e., a misbehaving node is less likely to be detected), as shown in Fig. 4(b), and that the false positive rate will increase (i.e., normal nodes become more likely to be wrongly warned), as shown in Fig. 3. Since p_f and $p_d(D)$ are two conflict objectives, h should be carefully chosen in order to achieve a balance between these two rates. For example, in order to achieve a detection rate higher than 95% while keeping false positive rate lower than 1%, we can choose $h = 5$ and $D = 10$.

3) Detection delay: The detection delay is measured as the interval between the time when the misbehaving node starts to misbehave and the time when it is detected by the AP. In our simulations, the misbehaving node randomly picks time instances to start misbehaving. The average detection delay along with the 95% confidence interval error bars are plotted in Fig. 5. As shown in this figure, the detection delay decreases as the misbehavior intensity increases (i.e., either CWmin or AIFSN decreases). Roughly speaking, as the intensity increases, the misbehaving node gets higher throughput which results in that its corresponding detector state X_k will get increased more frequently. In turn, the time that X_k hits its maximal becomes shorter, which means lower detection delay. In other words, as indicated by both Fig. 4 and 5, aggressive misbehavior will be quickly and accurately detected by the HS detector. This implies that, for an aggressive misbehaving node, it will be identified with a high probability even if it changes its misbehaving strategy (or switches between aggressive misbehaving and normal behaving) from time to time. On the other hand, however, the detector becomes insensitive for inconspicuous misbehavior (e.g., the cases with CWmin>20 and AIFSN=2 as shown in Fig. 5). For example, since the detection delay is relatively high for inconspicuous misbehavior (but still less than 0.2s), a misbehaving node may escape from being caught by using a short misbehaving period and performing moderate misbehavior during that period. Nevertheless, such a misbehaving node will get much less throughput advantage over other normal ones.

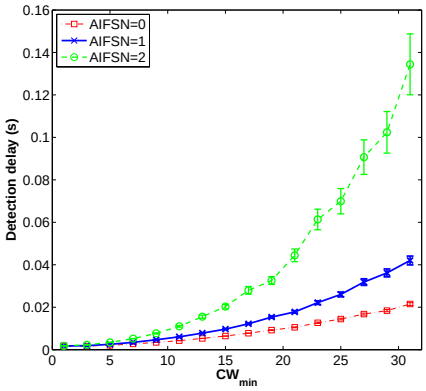


Fig. 5. Simulated detection delay with 95% confidence interval error bars.

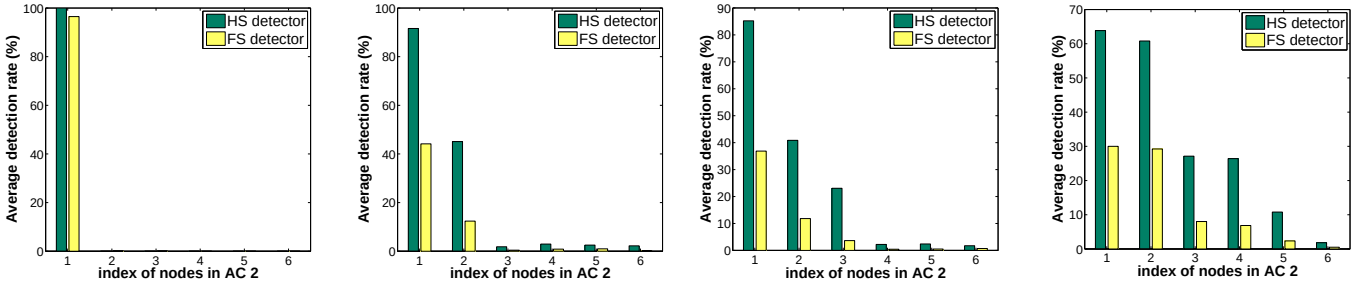
4) Comparison with FS detector: We compare the proposed detector with the existing FS detector. We consider the following four cases in the comparison study: the parameter of CWmin for all the 6 nodes in AC2 is chosen as $[5, 7, 15, 15, 15, 15]$ in case 1, $[1, 3, 5, 15, 15, 15]$ in case 2, $[5, 7, 9, 15, 15, 15]$ in case 3 and $[5, 5, 7, 7, 9, 15]$ in case 4. For all the four cases, we fix AIFSN=3 and CWmax=1023 for all the nodes in AC2. Note that those nodes in AC2 with CWmin different from the normal value (i.e., 15) are misbehaving nodes. The other parameters are set same as above. In case 1, since there is only one misbehaving node, the performance of HS and FS detectors is similar. However, in the other three cases, we can clearly observe that the proposed HS detector outperforms the existing FS detector. When multiple nodes misbehave, the HS detector can identify all of them with much higher probabilities than the FS detector. Basically, the FS detector makes detection decisions based on the relative resource sharing among the nodes in AC2. Thus, as discussed in Section III-C, a misbehaving node may not be identified if its relative resource sharing is low as compared to other misbehaving nodes in the same AC. Whereas, our HS detector make decisions based whether the target node's sharing exceeds the expected value. In addition, the FS detector only uses the received packets from the nodes in AC2 for detection, while the HS detector uses all received packets, which means the latter utilizes more information for making decisions.

However, Fig. 6 suggests that the proposed HS detector generates lower detection rates for mild misbehaving nodes (e.g., node 5 in case 4) than for aggressive misbehaving ones (e.g., node 1 in case 4), because the impact of mild ones is overwhelmed by the aggressive ones. We defer discussion about handling multiple misbehaving nodes for Section VIII.

B. Performance of the countermeasure

The performance of the proposed countermeasure is also evaluated and the results are shown in Fig. 7. As seen from Fig. 7(a), the long-term average packet dropping rate of the misbehaving node by the AP is increasing as the misbehaving intensity increases (i.e., CWmin decreases). In particular, for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

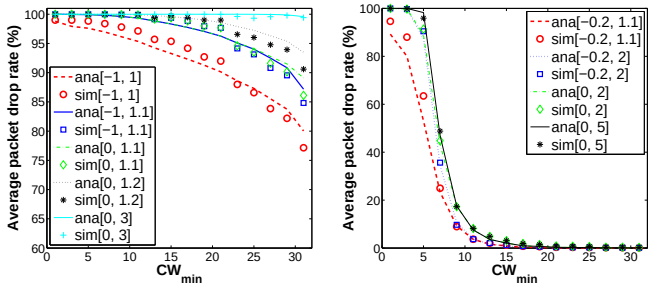


(a) Case 1: node 1 misbehaves (b) Case 2: nodes 1 and 2 misbehave (c) Case 3: nodes 1,2,3 misbehave (d) Case 4: nodes 1,2,3,4 misbehave

Fig. 6. Comparison between HS and FS detectors.

a small CWmin, the packets from the misbehaving node will be almost surely dropped by the AP, which means that the misbehaving node will be completely blocked by the AP. The figure also demonstrates that the dropping rate is significantly affected by the parameters Y_{inf} and Y_{sup} (i.e., the lower and upper bounds of Y_k, respectively). As Y_{sup} increases, the dropping rate will increase. This is reasonable because Y_k will be more likely to stay at a state higher than 1, which contributes to a higher dropping rate, referring to (26). Therefore, we can tune Y_{sup} in order to achieve a high dropping rate for the misbehaving node. Similarly, if we reduce Y_{inf}, the dropping rate will decrease; however, the impact of Y_{inf} is less significant, which can be seen by comparing the curves corresponding to $[0, 1.1]$ and $[-1, 1.1]$ in Fig. 7(a).

As shown in Fig. 7(b), when we choose $[Y_{\text{inf}}, Y_{\text{sup}}] = [-0.2, 2]$, the average packet dropping rate for a normal node (corresponding to CWmin=15 and AIFSN=3) is almost 0, which means that, under such a parameter setting, the normal node will not be affected by the proposed countermeasure. In sum, by carefully choosing Y_{inf} and Y_{sup}, we can achieve a satisfactory dropping rate as to punish the misbehaving node while protecting its throughput when it is well-behaving.



(a) AIFSN=0 (b) AIFSN=3

Fig. 7. Average packet dropping rate $\bar{\rho}$. The different curves correspond to different settings of $[Y_{\text{inf}}, Y_{\text{sup}}]$.

VIII. Discussions

A. Handling TXOP misbehavior

In this paper, we have assumed TXOP=0 for all nodes and that the TXOP misbehavior is not present. For TXOP misbehavior, the basic detection idea is quite straightforward, i.e., to locate the starting and ending time of the TXOP period and compare the duration with the prescribed value as determined by the parameter TXOPLimit [25]. Since there is no much randomness, the detection decision can be made by analyzing a few packets, which to some extent is easier than detecting contention window- and AIFS- misbehavior. The above idea of TXOP misbehavior detection can be further extended to handle scenarios of consecutive TXOP, referring to the method in [25]. On the other hand, in cases with nonzero TXOP parameters and in presence of possible TXOP misbehavior, the detection and mitigation methods proposed in this paper are still valid as long as the percentage of resource sharing s for each node under normal cases (i.e., all nodes do not misbehave) can be predicted. This can be achieved by extending the MAC model in Section IV to accommodate the cases with TXOPLimit> 0 (refer to the technique in [30] for the extension).

B. Multiple misbehaving nodes

As has been shown in Fig. 6, the proposed HS detector is able to identify multiple misbehaving nodes; whereas, the detection rates for mild misbehaving nodes are lower than those for aggressive misbehaving ones. This brings the possibility that an aggressive misbehaving node can screen other mild ones. Consider two misbehaving nodes with mild and aggressive misbehaving strategies, respectively, for instance. It is possible that the impact of the mild misbehaving node is overwhelmed by the aggressively misbehaving one, such that the percentage of resource sharing of the mild node is lower than the expected value s in view of the AP. In this case, the AP may correctly detect the aggressive misbehavior but ignore the mild one. However, together with the mitigation mechanism, the aggressive one will be punished and eventually driven to well-behaving, as long as it is rational (see Remark 3). Then, the mild one will be exposed to the detector and will be punished. Suppose that all nodes are rational and none of them wants to take the risk of being punished by the mitigating mechanism to screen others, each misbehaving node is unwilling to be too aggressive than others. Thus, misbehaving nodes will take similar strategies, but still, they will be punished simultaneously. In this sense, the proposed mechanism with both detection and mitigation is effective against multiple misbehaving nodes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

IX. Conclusion

We have investigated the problem of misbehavior detection and mitigation in cyber physical systems over IEEE 802.11e based networks where the nodes are able to choose different priority levels and different MAC parameters. Based on a mathematical model of the percentage of resource sharing of each node, we proposed a both hybrid-share real-time detector and a mitigation mechanism. Theoretical performance of the detector and the mechanism has been analyzed. Through extensive simulations, we demonstrated that the false positive rate is sensitive to the detection threshold but tolerable to the error involved in computing the MAC model. We also show that, by choosing appropriate parameters for the mitigation mechanism, the packets from a misbehaving node will be dropped with a high probability while those from a normal node will be almost not affected. In our future, we will extend the detector to detecting misbehavior in multihop networks.

Appendix

Proof of Theorem 1: Consider the stationary distribution $\boldsymbol{\pi}^*$ of the homogeneous Markov chain of $\{X_k\}$. Starting at a generic time, say $t = 1$ without loss of generality, the detection rate can be expressed as

$$\begin{aligned} p_d(D) &= 1 - \mathbb{P}[\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_{\lfloor \frac{D}{T^*} \rfloor}] \\ &= \sum_{t=1}^{\lfloor \frac{D}{T^*} \rfloor} \mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \widehat{X}_t] \\ &\triangleq \sum_{t=1}^{\lfloor \frac{D}{T^*} \rfloor} q_t, \end{aligned}$$

where $\widehat{X}_k$ and $\overline{X}_k$ stand for the events $X_k = \bar{m}\sigma$ and $X_k \neq \bar{m}\sigma$ (i.e., the events that the misbehavior is detected or not detected), respectively. Clearly, $q_1 = \pi^*_{\bar{m}}$.

$$\begin{aligned} q_t &= \mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \widehat{X}_t] \\ &= \mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \widehat{X}_t, \overline{X}_{t+1}] + \mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \widehat{X}_t, \widehat{X}_{t+1}] \\ &= \mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \overline{X}_t, \widehat{X}_{t+1}] + q_t \mathbb{P}[\widehat{X}_{t+1} | \widehat{X}_t] \\ &= q_{t+1} + q_t P^*_{\bar{m},\bar{m}}, \end{aligned}$$

In the third equality, since we are considering stationary distribution and t is generic, we can moving the horizon to review the probability $\mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \widehat{X}_t, \overline{X}_{t+1}]$ as $\mathbb{P}[\overline{X}_1, \ldots, \overline{X}_{t-1}, \overline{X}_t, \widehat{X}_{t+1}]$. This can be also seen by examining the reversibility of the Markov chain of states $\widehat{X}_k$ and $\overline{X}_k$. Then, the above equation indicates that

$$q_{t+1} = q_t(1 - P^*_{\bar{m},\bar{m}}) = q_t = \pi^*_{\bar{m}},$$

where $P^*_{\bar{m},\bar{m}} = 0$ as implied by (23). It then follows that (25) holds, which completes the proof. ∎

References

[1] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure," *Proc. IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

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

[3] K. Ota, M. Dong, Z. Cheng, J. Wang, X. Li, and X. S. Shen, "ORACLE: mobility control in wireless sensor and actor networks," *Computer Communications*, vol. 35, no. 9, pp. 1029–1037, 2012.

[4] J. Chen, Q. Yu, P. Cheng, Y. Sun, Y. Fan, and X. Shen, "Game theoretical approach for channel allocation in wireless sensor and actuator networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2332–2344, 2011.

[5] F.-J. Wu, Y.-F. Kao, and Y.-C. Tseng, "From wireless sensor networks towards cyber physical systems," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, 2011.

[6] W. Shen, L. Liu, X. Cao, Y. Hao, and Y. Cheng, "Cooperative message authentication in vehicular cyber-physical systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 84–97, 2013.

[7] K. Ota, M. Dong, S. Chang, and H. Zhu, "MMCD: cooperative downloading for highway VANETs," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 34–43, 2015.

[8] P. P. Parikh, T. S. Sidhu, and A. Shami, "A comprehensive investigation of wireless LAN for IEC 61850–based smart distribution substation applications," *IEEE Tran. Ind. Informat.*, vol. 9, no. 3, pp. 1466–1476, 2013.

[9] R. Viegas, L. A. Guedes, F. Vasques, P. Portugal, and R. Moraes, "A new MAC scheme specifically suited for real-time industrial communication based on IEEE 802.11e," *Computers & Electrical Engineering*, vol. 39, no. 6, pp. 1684–1704, 2013.

[10] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11ah: the WiFi approach for M2M communications," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 144–152, 2014.

[11] IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service enhancements," 2005.

[12] G. Cena, L. Seno, A. Valenzano, and C. Zunino, "On the performance of IEEE 802.11e wireless infrastructures for soft-real-time industrial applications," *IEEE Tran. Ind. Informat.*, vol. 6, no. 3, pp. 425–437, 2010.

[13] A. Soomro and D. Cavalcanti, "Opportunities and challenges in using WPAN and WLAN technologies in medical environments," *IEEE Commun. Mag.*, vol. 45, no. 2, pp. 114–122, 2007.

[14] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino, "Industrial applications of IEEE 802.11e WLANs," in *Proc. International Workshop on Factory Communication Systems (WFCS)*, 2008, pp. 129–138.

[15] M. Cheminod, L. Durante, and A. Valenzano, "Review of security issues in industrial networks," *IEEE Tran. Ind. Informat.*, vol. 9, no. 1, pp. 277–293, 2013.

[16] J. Chen, J. Li, and T. H. Lai, "Energy-efficient intrusion detection with a barrier of probabilistic sensors: Global and local," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4742–4755, 2013.

[17] R. Mitchell and I.-R. Chen, "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 16–30, 2015.

[18] W. Zeng and M.-Y. Chow, "Resilient distributed control in the presence of misbehaving agents in networked control systems," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2038–2049, 2014.

[19] S. Ntalampiras, "Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 104–111, 2015.

[20] A. Lopez Toledo and X. Wang, "A robust Kolmogorov-Smirnov detector for misbehavior in IEEE 802.11 DCF," in *Proc. IEEE ICC*, 2007, pp. 1564–1569.

[21] J. Tang, Y. Cheng, and W. Zhuang, "Real-time misbehavior detection in IEEE 802.11-based wireless networks: an analytical approach," *IEEE Trans. Mobile Comput.*, vol. 13, no. 1, pp. 146–158, 2014.

[22] G. Bianchi, I. Tinnirello, and L. Scalia, "Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations," *IEEE Netw.*, vol. 19, no. 4, pp. 28–34, 2005.

[23] S. Szott, M. Natkaniec, and R. Canonico, "Detecting backoff misbehaviour in IEEE 802.11 EDCA," *European Transactions on Telecommunications*, vol. 22, no. 1, pp. 31–34, 2011.

[24] P. Serrano, A. Banchs, V. Targon, and J. Kukielka, "Detecting selfish configurations in 802.11 WLANs," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 142–144, 2010.

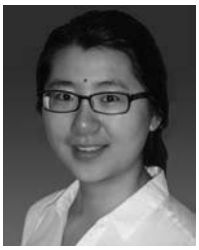
[25] Y. W. Ahn, J. Baek, A. M. K. Cheng, P. S. Fisher, and M. Jo, "A fair transmission opportunity by detecting and punishing the malicious wireless stations in IEEE 802.11e EDCA network," *IEEE Syst. J.*, vol. 5, no. 4, pp. 486–494, 2011.

[26] X. Cao, L. Liu, W. Shen, J. Tang, and Y. Cheng, "Real-time misbehavior detection in IEEE 802.11e based WLANs," in *Proc. IEEE Globecom*, 2014, pp. 631–636.

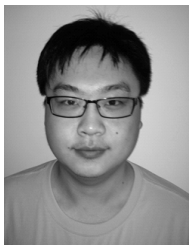
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2015.2499123, IEEE Transactions on Industrial Informatics

[27] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1506–1515, 2005.
[28] K. Kosek-Szott, M. Natkaniec, and A. R. Pach, "A simple but accurate throughput model for IEEE 802.11 EDCA in saturation and non-saturation conditions," *Computer Networks*, vol. 55, no. 3, pp. 622–635, 2011.
[29] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.
[30] D. Xu, T. Sakurai, and H. L. Vu, "An access delay model for IEEE 802.11e EDCA," *IEEE Trans. Mobile Comput.*, vol. 8, no. 2, pp. 261–275, 2009.

Xianghui Cao (IEEE S'08-M'11) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. From December 2007 to June 2009, he was a Visiting Scholar with Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. From July 2012 to July 2015, he was a Senior Research Associate with Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. Currently he is an Associate Professor with School of Automation, Southeast University, Nanjing, China. His research interests include wireless sensor and actuator networks, wireless network performance analysis and network security. He serves as Publicity Co-chair for ACM MobiHoc 2015, Symposium Co-chair for IEEE/CIC ICCC 2015, and TPC Member for a number of conferences. He also serves as an Associate Editor of several journals, including KSII Transactions on Internet and Information Systems, Security and Communication Networks and International Journal of Ad Hoc and Ubiquitous Computing. He was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc 2014.



Lu Liu (IEEE S'13) received the B.S. degree in Automation from Tsinghua University, Beijing, China, in 2010, and the M.S. degree in Electrical Engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2012. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at Illinois Institute of Technology, Chicago, IL, USA. Her current research interests include energy efficient networking and communications, performance analysis and protocol design of wireless networks.



Wenlong Shen (IEEE S'13) received the B.E. degree in Electrical Engineering from Beihang University, Beijing, China, in 2010, and the M.S. degree in Telecommunications from University of Maryland, College Park, USA, in 2012. He is currently pursuing the Ph.D. degree in the department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. His research interests include vehicular ad hoc networks, mobile cloud computing, and network security.

Aurobinda Laha received his B.Tech degree in Electronics and Communication Engineering from National Institute of Technology, Durgapur, India in 2010 and his M.S. degree in Electrical Engineering from Illinois Institute of Technology, Chicago, IL, USA in 2012. He is currently pursuing his PhD degree in the Electrical and Computer Engineering Department at Illinois Institute of Technology, Chicago, IL, USA. His research interests include vehicular ad hoc networks, plug-in hybrid electric vehicles (PHEVs) and energy efficient networking.

Jin Tang (IEEE S'10-M'13) received the B.S. degree in Computer Science from Fudan University, Shanghai, China, in 2004, the Master's degree in Information Technology and Management from Illinois Institute of Technology, Chicago, IL, USA, in 2007, and the Ph.D. degree in Computer Engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2012. He is now with AT&T Labs. His current research interests include wireless network security, intrusion detection and security in VoIP applications. He received a Best Paper Award from IEEE ICC 2011.

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