applied sciences

MDPI

Article

Detection of Misconfigured BYOD Devices in Wi-Fi Networks

Jaehyuk Choi

Department of Software, Gachon University, 1342 Seongnamdaero, Seongnam-si 1320, Korea; jchoi@gachon.ac.kr; Tel.: +82-31-750-8657

Received: 15 September 2020; Accepted: 12 October 2020; Published: 15 October 2020

Abstract: As Bring Your Own Device (BYOD) policy has become widely accepted in the enterprise, anyone with a mobile device that supports Wi-Fi tethering can provide an active wireless Internet connection to other devices without restriction from network administrators. Despite the potential benefits of Wi-Fi tethering, it raises new security issues. The open source nature of mobile operating systems (e.g., Google Android or OpenWrt) can be easily manipulated by selfish users to provide an unfair advantage throughput performance to their tethered devices. The unauthorized tethering can interfere with nearby well-planned access points (APs) within Wi-Fi networks, which results in serious performance problems. In this paper, we first conduct an extensive evaluation study and demonstrate that the abuse of Wi-Fi tethering that adjusts the clear channel access parameters has strong adverse effects in Wi-Fi networks, while providing the manipulated device a high throughput gain. Subsequently, an online detection scheme diagnoses the network condition and detects selfish tethering devices by passively exploiting the packet loss information of on-going transmissions. Our evaluation results show that the proposed method accurately distinguishes the manipulated tethering behavior from other types of misbehavior, including the hidden node problem.

Keywords: network monitoring; dynamic sensitivity control; Wi-Fi tethering; sensing of misbehavior

1. Introduction

Bring Your Own Device (BYOD), which refers to a policy of permitting employees to bring their personal devices to the workplace and to use those devices by connecting them to the internal network, has become a common practice in the enterprise [1]. As of 2016, six out of 10 companies had a BYOD-friendly policy in the workplace and 87% of companies rely on their employees having access to business apps from their personal devices, according to Syntonic's recent survey [2]. Although it offers benefits, such as increased employee productivity and greater employee satisfaction, adopting a BYOD policy poses several potential security holes to the organization, since BYOD devices can bypass the managed network without the controls in place. One of the most relevant wireless technologies is Wi-Fi tethering, which uses a mobile device (e.g., a smartphone or a tablet) as a Wi-Fi hotspot and provides a wireless Internet connectivity to other devices [3].

While Wi-Fi tethering allows for mobile users to go online almost anywhere, even on-the-move, the ease of setting up a tethered Wi-Fi hotspot and the open source nature of mobile Operating Systems (OSes) can pose serious performance problems to well-planned Wi-Fi networks, such as enterprise and campus networks [4]. Because the tethered Wi-Fi hotspot can potentially open up the network with an arbitrary channel number without restriction from the network administrator, it may interfere with nearby well-planned access points (APs) and cause unpredictable performance degradation, e.g., availability and throughput. Even worse, the open and customizable nature of mobile OSes, such as Google Android and OpenWrt, can be further abused by selfish users in order to provide their tethering an unfair advantage in throughput performance. For example, by rooting or jailbreaking

mobile OSes, the channel access functions of Wi-Fi protocol, i.e., IEEE 802.11, can be manipulated "selfishly", at the cost of other nearby well-behaving Wi-Fi device performance. Note that the newest IEEE 802.11ax standard (aka Wi-Fi 6) [5] has been investigating adaptive carrier sensing (CS), which is also known as dynamic sensitivity control (DSC), in order to improve the spatial reuse in dense deployments [6]. This implies that the parameters of Wi-Fi tethering, particularly Clear Channel Assessment (CCA) thresholds are readily tunable and can be manipulated in order to gain an unfair advantage in throughput performance. Thus, it is essential to design an efficient mechanism in order to detect unauthorized and misconfigured Wi-Fi tethering.

In this paper, we observe the network throughput behavior in a multi-AP environment in the presence of a tethering device, and study the impact of selfish tethering on other users' throughput performance. We show that the abuse of Wi-Fi tethering that adjusts the CCA threshold has strong adverse effects in Wi-Fi networks, while providing the tethered device a high throughput gain. Our evaluation results reveal that the symptoms of selfish tethering resembles that of the well-known hidden node problem—they both result in a high frame transmission error rate. However, the main difference is that the frame losses at legitimate nodes that are caused by the selfish tethering cannot be easily resolved by the RTS/CTS mechanism, because the selfish nodes may not recognize or sense the RTS/CTS frames due to their manipulated CCA thresholds. Therefore, it is imperative to design a detection mechanism that can identify the root cause of the throughput degradation and pinpoint the selfish node in the network.

To this end, we present an online detection mechanism, called CUBIA, which can accurately detect misbehaving Wi-Fi tethering nodes in a multi-AP Wi-Fi network. CUBIA runs at each AP and passively monitors the ongoing data traffic to detect any abnormal changes caused by target selfish tethering. In particular, it monitors any noticeable increase in frame loss rate and identifies the root cause of the frame loss, e.g., selfish tethering or hidden node problem. For this, we formulate the selfish tethering detection problem as a hypothesis testing problem and employs a modified CUSUM algorithm. We evaluate CUBIA with an in-depth simulation in various wireless environments, and our evaluation results show high detection accuracy. To best of our knowledge, this is the first paper to consider the problem of detecting selfishly misconfigured Wi-Fi tethering devices in a multi-AP network.

Contributions. This paper makes several contributions, as follows.

- We observe that concurrent transmission mechanisms, e.g., PHY capture or message-in-message (MIM), can be abused by selfish nodes, and study the impact of the selfish Wi-Fi tethering behavior on the throughput performance of other nearby well-behaving nodes.
- We propose a selfish misbehavior detection mechanism, called CUBIA, which can accurately identify the cause of frame losses and detect selfish behavior under strict detection latency requirements.
- We design a two-step detection process using an online change detection algorithm, i.e., CUSUM, which is based on passive monitoring of frame loss rates at multiple APs. We also study the impact of detection thresholds on detection latency/accuracy performance.
- We perform extensive simulation-based evaluation for various network conditions. Our evaluation results show that CUBIA can promptly detect selfish behavior with high accuracy in realistic wireless environments.

Organization. The remainder of paper is organized as follows. Section 2 describes the system model and illustrates the impact of selfish manipulation of the CCA threshold (CST) on the throughput performance in multi-AP environments. Section 3 identifies the unique features of selfish tethering and proposes CUBIA that accurately detects the selfish behavior. Section 4 presents the evaluation results. We summarize related research work in Section 5 and Section 6 concludes the paper.

2. Selfish Configuration in Unauthorized Wi-Fi Tethering

In this section, we first introduce the system model, including the adversary model and assumptions. Subsequently, we present the problem illustrating the impact of the selfish configuration of Wi-Fi tethering while using the CCA manipulation in a managed multi-AP environment.

2.1. System Model and Assumption

We consider a scenario where an unauthorized Wi-Fi tethering system creates a Wi-Fi hotspot within a multi-AP Wi-Fi network, as illustrated in Figure 1. The tethering consists of an Internet-connected (e.g., through 4G or 5G cellular connection) mobile device (e.g., a smartphone or a tablet) and a tethered Wi-Fi-enabled device, which forms a Wi-Fi tethering link. Henceforth, we will refer to the mobile device and the tethered Wi-Fi device as host and guest nodes, respectively; the host node shares its cellular Internet connection with the guest nodes via the tethered Wi-Fi link. These nodes contend for channel access with nearby Wi-Fi systems.

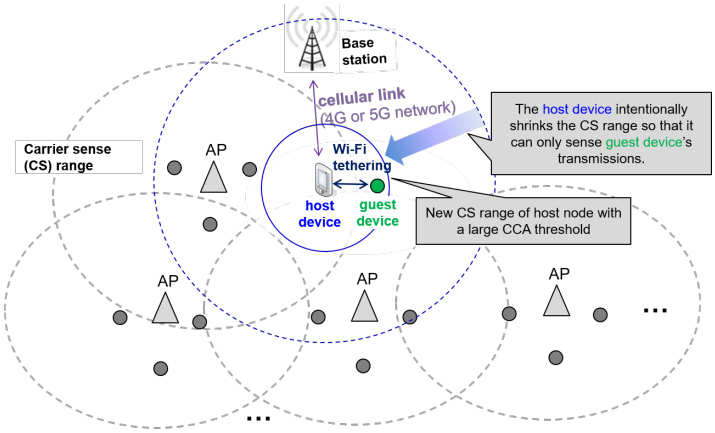


Figure 1. Illustration of the problem: a misbehaving unauthorized Wi-Fi tethering sets up the network in a managed multi-access point (AP) network.

Figure 2 describes the adversary model. We assume that an adversary user manipulates the CCA threshold (CST) of the host node's Wi-Fi interface in the tethering whereas the guest nodes (e.g., laptop, tablet, or etc.) are legitimate, i.e., guest nodes use the default CCA threshold. The CCA threshold controls the sensitivity of carrier sense, that is, a node with a higher CCA threshold can access channel with a higher interference/noise level [6,7]. The channel is considered to be idle when the sensed energy is lower than the CCA threshold. We assume that the host node selects the maximum possible CCA threshold without losing the tethering connectivity, which is likely a little lower than the observed strength of received signals transmitted by the guest nodes.

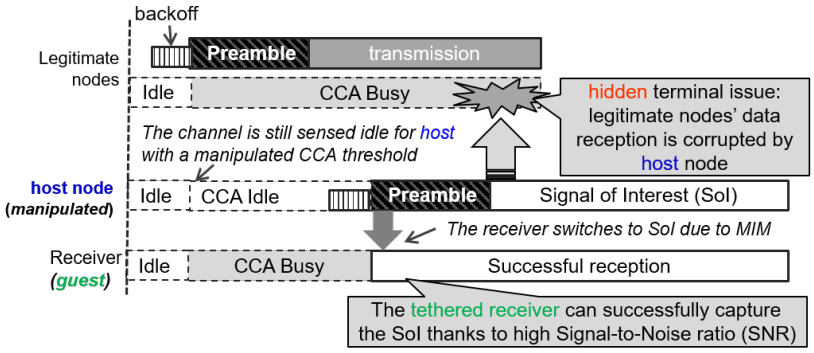
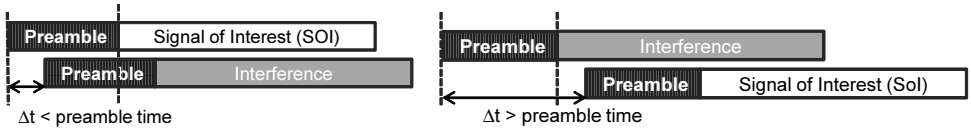


Figure 2. Adversary model: selfish behavior with Clear Channel Assessment (CCA) manipulation will not freeze its back-off counter even if other nodes are transmitting.

Recall a property of Wi-Fi tethering: in general, a tethered hotspot is formed for communication between personally owned devices that are placed close-by while in use and, thus, the link distance between the communicating devices is highly controllable and typically short (less than 10 m or 30 ft [8]).

When multiple independent transmissions occur simultaneously to a conventional receiver, the receiver is likely to fail to receive the signal of interest (SoI) due to severe interference. However, for a tethered link with a short distance, it is possible for its tethered receiver in order to successfully capture the SoI thanks to two well-known concurrent transmission technologies: PHY capture effect [9] and message-in-message (MIM) [10,11].

Figure 3 illustrates the main difference between PHY capture and MIM.



(**a**) PHY capture effect (**b**) MIM (Message-In-Message)

Figure 3. (**a**) PHY capture effect allows a receiver to successfully capture the signal of interest (SoI) if its Tx power is sufficiently higher than the sum of interferences. (**b**) MIM (Message-In-Message) allows for a receiver to disengage from an ongoing packet reception, and engage in a new, stronger packet.

The PHY capture effect is the property of 802.11 radios, where an SoI can be successfully decoded, even when the interference arrives at the same time, as long as the overlap within the preamble detection stage and the SoI is stronger than a specific threshold, which we call the capture threshold. MIM is an enhanced PHY-layer capability that enables a receiver to decode an SoI, even if the SoI arrives after the preamble time of the interference. Specifically, MIM allows for a receiver to disengage from the current on-going frame reception and re-engage in a new, stronger frame. However, MIM requires a higher received signal to interference ratio (SIR) than the PHY capture, which we call this SIR value the MIM threshold, β_m, for modulation scheme m (e.g., m = BPSK, QPSK, 16QAM, and 64QAM), where we consider multiple data rates with various modulation and coding scheme (MCS) values as in 802.11 n/ac.

Following the experimental results in [10], throughout the paper, we configure the SIR threshold γ_m and MIM threshold β_m for a given modulation scheme m. We assume that the receiver can successfully decode a frame when the received SIR is consistently above β_m, even when multiple independent transmissions occur simultaneously.

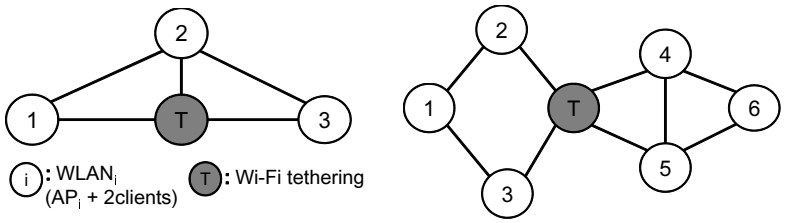
2.2. Selfish Configuration in Wi-Fi Tethering

In order to understand the impact of the selfish behavior of a Wi-Fi tethering system using CCA tuning, we performed extensive simulations with different transport-layer protocols (i.e., TCP and UDP, in multi-AP environments) while using a network simulator [12].

Multi-AP Topology: first, we study the impact when a selfish tethering launches within a well-planned multi-AP network with two representative topologies. The topologies used in the simulations are shown in Figure 4. We sampled the topologies from a popular Wi-Fi database *Wigle* [13]. For example, the topology that is shown in Figure 4a corresponds to the well-known FIM (Flow-In-the-Middle) topology [14], which can be easily found in real-world AP deployments [15]. An edge in the figure represents the neighboring interference, meaning that an AP is in the carrier sensing range of its connected APs. The vertex denoted by 'T' in the figure represents the tethering system launched within the network.

We compare the performance for two cases while using UDP and TCP protocols: (i) Normal scenario, in which the host node uses the default CCA threshold and (ii) Selfish scenario, in which the host node manipulates the MAC protocol by increasing the CCA threshold as high as possible without losing the connectivity to the guest. We used the following settings in our simulations. Each Wi-Fi has an AP and two associated client nodes (i.e., 2 AP–clients flows per Wi-Fi), and all of the nodes operate on the same channel. We considered typical IEEE 802.11n MAC/PHY parameters with the maximum speed of 54 Mbps, and set the capacity B_{cel} of the cellular backhaul link of tethering to 5, 10,

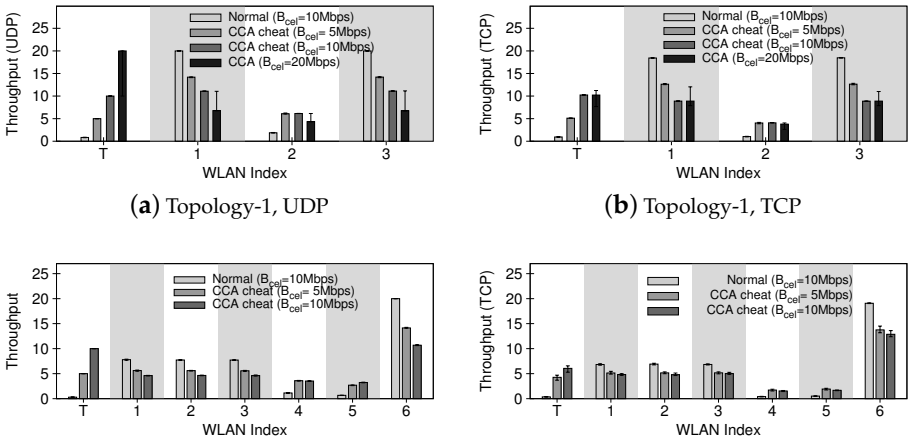
and 20 Mbps. Traffic is generated by a constant bit rate (CBR) traffic generator for the UDP protocol (1 KB packet size, uplink direction), and an FTP download application is used to create TCP flows (1.5 KB packet size, downlink direction). For UDP flows, we assume 10 Mbps for flows in infrastructure APs and 20 Mbps for tethering flows.



(**a**) Topology-1 (**b**) Topology-2

Figure 4. AP Interference graph of two simulated topologies.

Figure 5 shows the throughput of the tethering link and APs. In Normal case, we can see a throughput imbalance among APs and, especially, the tethering flow is shown to experience starvation due to the FIM problem [14]. With the CCA cheating, on the other hand, we can see that the selfish tethering achieves a significant throughput gain at the cost of significant throughput degradation for other nearby well-behaving APs, for both UDP and TCP flows. Although we only manipulate the CCA threshold on the host side, i.e., the guest node is legitimate, the selfish tethering achieves a high throughput gain, even with the TCP protocol, which is a bi-directional, transfer-based protocol. This is attributed to the closed-loop TCP-ACK mechanism, i.e., the more data packets (or ACKs), the legitimate guest successfully receives from the misconfigured host, the more outstanding uplink ACKs (or data packets) the guest can transmit. The figure shows that the throughput gain increases proportional to the cellular backhaul link bandwidth, B_{cel}, for both UDP and TCP flows.



(**a**) Topology-1, UDP (**b**) Topology-1, TCP

(**c**) Topology-2, UDP (**d**) Topology-2, TCP

Figure 5. Impact of selfish carrier sense on throughput of transport-layer protocols over various cellular backhaul link capacities B_{cel} for tethering in two multi-AP topologies.

One can expect that selfish CCA manipulation would increase the collision probability of the tethering link, since the host node initiates packet transmission, even in the present of other nodes' transmissions. Nevertheless, we observe a significant gain of selfish behavior using CCA manipulation, as seen from the above results. We investigate the property of the successful receptions at the tethering receiver node, i.e., guest node in order to understand why the selfish tethering link can achieve a significant gain with CCA manipulation. Table 1 plots the collision probability, and the percentage of successful receptions at the receiver in the simulation with B_{cell} = 10 Mbps corresponding to the

results that are shown in Figure 5a,c, respectively. We observe that, despite the high collision ratios (71.5% and 91.8% for Topology 1 and 2, respectively), the tethered receiver can capture the signal of interest (SoI), thanks to PHY capture effect and MIM. A majority of the successful receptions are due to MIM, i.e., 53.6% and 63.5% for Topology 1 and 2, respectively. Note that the link distances between host and guest nodes are very short, which makes the received signal at the guest node sufficiently stronger than the sum of interferences.

Table 1. Statistics of receptions at the receiver (in case of UDP and B_{cell} = 10 Mbps).

Topology	Collision Prob.	Capture Effect	MIM
1	71.5	17.9	53.6
2	91.8	28.3	63.5

These results clearly demonstrate that the selfish behavior using CCA manipulation abuses the short link distance property of Wi-Fi tethering and, thus, makes it possible to exploit the benefits of MIM. Consequently, the selfish tethering can achieve an unfair throughput gain, regardless of the network condition surrounding the tethering.

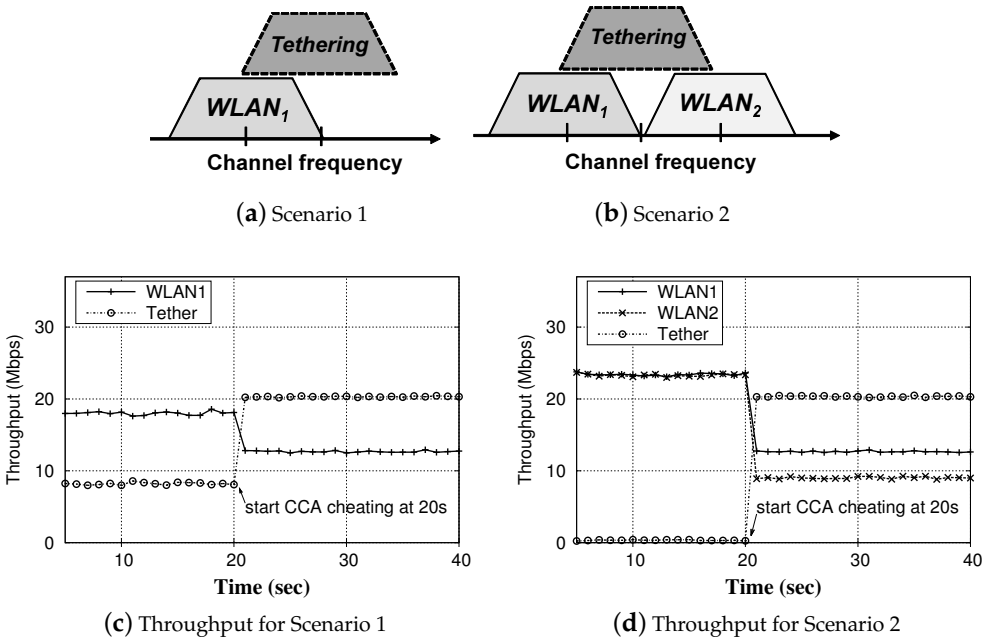
Impact of Tethering Channel: Next, we study the impact of the CCA manipulation on performance when tethering is launched on a channel partially overlapping with nearby APs. As mentioned above, the tethered Wi-Fi hotspot can potentially set up the network with an arbitrary channel number, which may cause serious interference to nearby well-planned APs. We used two simple scenarios in [16] and the channel model thta is presented in [17] for our simulation. Figure 6a illustrates a simple case that the channel selected by the tethering is partially overlapping with a nearby AP. Because two overlapping channels are sensed by each other by the CCA mechanism of 802.11, its effective spectrum usage is only 20 MHz [16]. Figure 6b depicts the case when the tethering shares the spectrum with two adjacent orthogonal channels. Note that, in this case, the tethering link may suffer from channel starvation [16]; the tethering link can only transmit when both $WLAN_1$ and $WLAN_2$ are idle, but the probability of the two outer channels being idle at the same time is very low, because the channel activities on the two outer channels are asynchronous and may overlap randomly.

We performed simulations with following setting in order to evaluate the impact of the selfish behavior using CCA manipulation in these scenarios. Each Wi-Fi consists of two client nodes where the traffic on each flow is generated with 10 Mbps downlink CBR over UDP. The capacity B_{cel} for tethering is configured to 20 Mbps (with 20 Mbps CBR/UDP traffic). Initially, the tethering link is set to be legitimate. The CCA manipulation is activated at 20 s.

Figure 6c,d plot the resulting throughput showing the effect of selfish behavior while using CCA manipulation. When the tethering is legitimate, it achieves a fair share of shared medium with two flows in $WLAN_1$ in the first scenario. In the second case shown in Figure 6b, the tethering is almost starved due to the channel starvation problem [16]. After the CCA value is selfishly configured, despite the same unfair channel condition, the tethering link achieves a significant throughput gain at the cost of significant reduction in throughput of other flows.

In the same scenario that is depicted in Figure 6b, we also compare the impact of CCA manipulation with different types of selfish behavior manipulating other MAC parameters, in particular the manipulation of the backff mechanism using smaller values of CW_{min}, where the manipulation of backoff mechanism is widely adopted by selfish users [18].

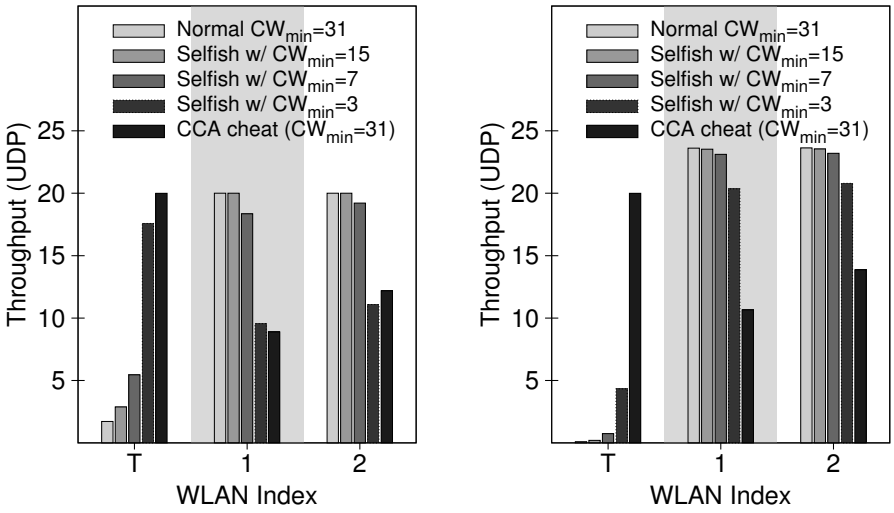
Figure 7 shows the simulation results with different values of CW_{min} = 31, 15, 7, and 3. The figure indicates that the selfish behavior using CCA manipulation achieves throughput gain above those with smaller values of CW_{min} and even higher than very aggressive setting with $CW_{min} = 3$. The results imply that the manipulation of CCA threshold is a simple, yet more attractive approach that can be abused by adversaries in a tethering environment.



(**a**) Scenario 1 (**b**) Scenario 2

(**c**) Throughput for Scenario 1 (**d**) Throughput for Scenario 2

Figure 6. Impact of launching tethering on a partial-overlapped channel.



(**a**) 2 client nodes per WLAN (**b**) 5 client nodes per WLAN

Figure 7. Throughput comparison with selfish configurations of the CW_{min} parameter.

Impact of Cellular Backhaul Link Capacity: Finally, we evaluate the impact of the backhaul link capacity of selfish tethering.

Figure 8 shows the throughput gain of the selfish behavior as a function of backhaul link capacity for TCP and UDP downlink traffic in a network with a high density of APs consisting of 10 APs and 30 client nodes. The figure indicates that the throughput gain is proportional to the backhaul link capacity of the maximum achievable goodput determined by the transport-layer protocol. Note that, in our simulation, when the backhaul apacity is larger than 20 Mbps, the maximum throughput is bounded by the Wi-Fi link capacity in this simulation. This is because the higher backhaul link capacity the tethering is connected to, the more outstanding packets the selfish node can transmit.

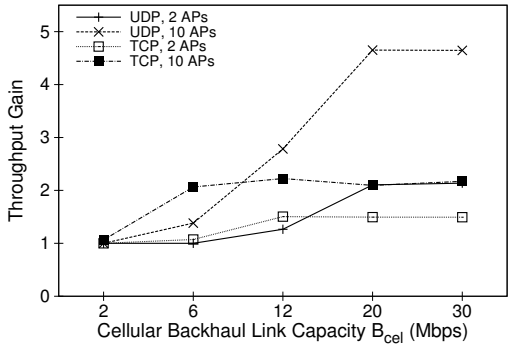


Figure 8. Throughput gain of selfish carrier sensing over various cellular backhaul link capacities and AP densities.

3. A Proposed Online Detection Algorithm

The symptoms of selfish tethering resembles those of the well-known hidden node problem—they both result in excessive frame loss of nearby legitimate APs. Thus, the successful detection of selfish misbehavior depends largely on a AP's ability to identify the root cause of its frame losses. Note that in 802.11 Wi-Fi, there are four main causes of frame losses: (i) PHY-layer link quality degradation, (ii) MAC-layer collision, (iii) hidden nodes, and (iv) selfish misbehavior (e.g., manipulation of the CCA thresholds).

Fortunately, selfish tethering exhibits unique features that can facilitate distinguishing the selfish carrier sensing of a tethering host from other causes of frame losses, especially from the hidden node problem. The main difference is that the frame losses at legitimate nodes that are caused by selfish tethering cannot be easily resolved by the RTS/CTS mechanism. This is because the selfish nodes may not recognize the RTS/CTS frames, owing to their manipulated CCA thresholds (i.e., their short sensing ranges) or RTS/CTS frames can be intentionally ignored by the selfish nodes.

Based on this observation, we propose a simple, yet efficient online detection algorithm, called CUBIA (*CUsum-Based Interference inference with Adaptive RTS/CTS*) that can accurately distinguish selfish behavior with CCA manipulation from other types of network problems, such as severe network congestion (i.e., collisions) and the hidden node problem. CUBIA operates at each AP and diagnoses the network condition by passively monitoring the ongoing traffic with its client nodes. Specifically, if CUBIA detects severe and persistent frame transmission failures, then it employs the RTS/CTS exchange before each data transmission to eliminate the possibility of the hidden node problem. For this, CUBIA employs the CUSUM (CUmulative SUM) algorithm [19] to quantify the duration of the frame losses. CUSUM algorithm suits our needs, because it is simple and light-weight and has been widely used for the detection of state changes. In the following, we describe the detailed procedure of the detection mechanism whlie using the CUSUM algorithm.

CUBIA monitors the frame error rate (FER) for every m transmissions to detect any abrupt changes in network condition, which could indicate the possibility of selfish tethering nodes. Let p_k denote the frame error rate for the k-th measurement period at the AP, given by $p_k = e_k/m$, where e_k denotes the number of transmission failures. Subsequently, we calculate the average error probability $E[p]$ by a moving average to reflect the network dynamics as:

$$E[p] = \lambda \cdot E[p] + (1 - \lambda) \cdot p_k. \tag{1}$$

We define the CUSUM change detection filter c_k^i of the AP, as:

$$\begin{aligned} c_k &= \max(\ 0,\ c_{k-1} + p_k - v\ ), \\ c_0 &= 0, \end{aligned} \tag{2}$$

where v is a drift parameter, which is a filter design parameter. v is configured differently, according to the value of c_k, as:

$$v = \begin{cases} E[p] + \mathcal{T}_{FER}, & \text{if } c_k < \theta_{AS} \\ \mathcal{T}_{FER}, & \text{if } \theta_{AS} \leq c_k \leq \theta_S, \end{cases} \quad (3)$$

where θ_{AS} and θ_S denote the first alarm threshold for asymmetric carrier sensing and the detection (second) alarm threshold for inferring selfish carrier sensing, respectively. CUBIA issues the first alarm to the AP when $c_k > \theta_{AS}$. Subsequently, CUBIA adaptively activates the RTS/CTS exchange mechanism and continues to track the change detection with $v = \mathcal{T}_{FER}$ in Equations (2) and (3).

Note that the first alarm can be caused by a sudden severe MAC layer congestion or a hidden node problem. However, in such a case, the magnitude of c_k tends to decrease with RTC/CTS frames, because the collisions are filtered out [20] and the hidden terminal problem are mitigated with RTS/CTS exchange. The frame error rate decreases accordingly.

Conversely, if the first alarm is caused by the selfish carrier sensing problem, a frame transmission following successful RTS/CTS exchanges will be interfered with and, hence, even with RTS/CTS, the magnitude of c_k would continue to increase, even beyond θ_S. Recall that, under a normal condition, the PHY-layer link quality is stable and has a certain upper bound $\mathcal{T}_{FER}$, namely, the target FER, which is guaranteed by the underlying rate-adaptation scheme that adjusts the modulation schemes to meet the target FER (e.g., $\mathcal{T}_{FER}$ = 0.05 is used in the evaluation). For example, practical rate adaptations [21] adjust the modulation schemes to meet the target FER (frame error rate) and, thus, guarantee the average FER performance to be maintained around the target value.

Consequently, the AP issues a detection alarm to the central controller in the network (see Figure 1) if $c_k > \theta_S$, which may trigger a follow-up action at the controller level to solve the selfish carrier sensing problem within the managed network.

Remark. Although the focus of this paper is to propose an AP-level detection mechanism, it is important to cope with the selfish misbehavior detected within the network at the system level. Here, we briefly discuss how the controller determines when to take follow-up actions to cope with the selfish misbehavior detected within the network. In typical managed Wi-Fi networks, the information that is obtained at APs is integrated on the controller. Subsequently, the controller utilizes the information to improve the detection accuracy. When the selfish tethering is present, the majority of nearby APs will be likely to experience the similar severe packet errors simultaneously. Consequently, the victim APs send the detection alarm to the controller at the same time. By exploiting the spatial and temporal correlation in those alarms, the controller identifies the root of the problem more effectively and accurately. For example, if a certain condition is satisfied, then the controller can take various follow-up actions, which can be (i) localizing the rogue interfering node [22], and (ii) remedying victim APs (e.g., interference-aware channel reassignment), etc. However, the detailed follow-up actions are beyond the scope of this paper.

4. Performance Evaluation

We now evaluate the performance of the proposed detection algorithm via simulation. We have implemented the proposed CUBIA in a network simulator [12].

4.1. Simulation Setup

In the simulation, the multi-AP network is deployed in a 200 × 200 m^2 area, where five APs with 10 client nodes are randomly generated (the link distance between a client and its AP is randomly selected from the range of 1–35 m); this represents a densely-populated configuration fully covering the entire area. The transmission range and carrier sensing range of legitimate nodes are set to 75 m, and 150 m, respectively. A selfish tethering pair is placed at the center of the area, whose link distance and carrier sensing are set to 1 m and 10 m, respectively. Table 2 lists the parameter values that were used in the simulation study.

Table 2. Parameters used in performance evaluation.

Parameter	Value
Transmission range	75
Carrier range	150
Data rate/ACK rate	54 Mbps/6 Mbps
CBR rate per AP-client pair (UDP)	20 Mbps
payload size of UDP , TCP	1000, 1500 bytes
$\mathcal{T}_{FER}$ (in Equation (3))	0.05
λ (in Equation (1))	0.1
m (number of transmissions for p_k in Equation (1))	10
the maximum allowed latency of detection	2 s

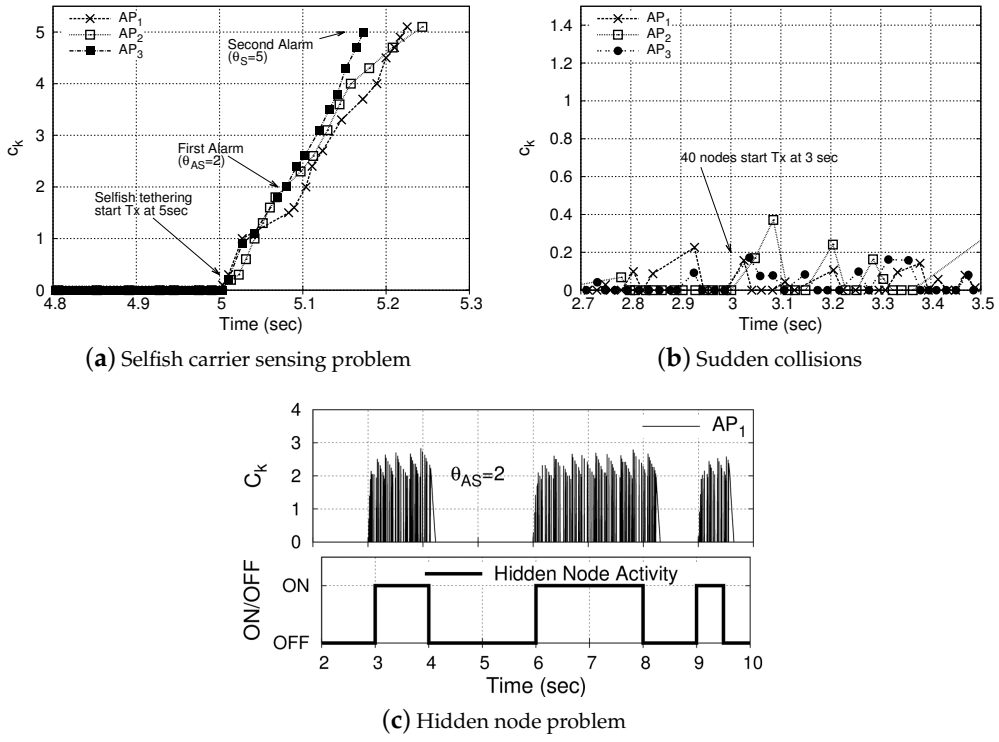
The performance is evaluated in terms of detection accuracy and time for TCP and UDP protocols. We consider a downlink scenario, where each AP transmits frames to its client nodes.

4.2. Detection Performance

4.2.1. Accuracy of Frame Loss Differentiation

To demonstrate the efficacy of CUBIA in distinguishing the selfish carrier sensing problem, we consider three testing scenarios: (i) selfish carrier sensing, (ii) hidden node problem, and (iii) collisions.

Figure 9 shows the temporal behavior of CUSUM change detection filter c_k for the three testing scenarios. Figure 9a plots the results of CUBIA over time for the case of selfish problem in the topology that is depicted in Figure 9b. We can observe that the detection filters of APs in the interference range continue to increase, where the selfish node starts transmissions at 5 s.



(**a**) Selfish carrier sensing problem

(**b**) Sudden collisions

(**c**) Hidden node problem

Figure 9. The dynamics of CUBIA toward three difference types of frame losses.

Figure 9b,c show CUBIA's ability of filtering the collisions and the hidden node problem, respectively. To simulate an abrupt change in the network state, initially, 10 AP–clients pairs are considered until an additional 40 nodes are abruptly activated at 3 s. Figure 9b demonstrates that

CUBIA effectively filters out the MAC-layer collisions. We generated ON/OFF traffic on a hidden node in order to simulate the impact of the hidden node problem, as illustrated in Figure 9c. In the figure, we can see that the detection filter of CUBIA promptly reacts to the hidden node, increasing above θ_{AS}, but the value of c_k remains low and does not exceed θ_S. This is because the effect of a hidden node is mitigated by adaptive RTS/CTS changes, thus imparting a negative drift to the detection filter.

Overall, our proposed scheme CUBIA accurately distinguishes the selfish misbehavior with CCA manipulation from collisions and the hidden node problem.

4.2.2. Detection Performance

We now evaluate the detection performance of CUBIA. It is important to meet the detectability requirements, such as the maximum allowed latency of detection. In this simulation study, we assume that the maximum allowed latency of detection is 2 s, which is, if the selfish misbehavior is not detected within 2 s, then we consider this case as a mis-detection. In practice, the detection latency requirement can be adjusted based on specific needs/conditions of the network and protocol stack. For example, an extended exposure to selfish carrier sensing can cause TCP congestion control and fast recovery algorithms to kick in, which make it very difficult to recover the throughput performance. There also exists a tradeoff between the detection sensitivity and false-alarm rate, which is part of our future work. We run the simulation 150 times for each set of tests with various backhaul link capacities B_{cel} of selfish tethering and alarm thresholds.

Tables 3 and 4 show the detection results for UDP and TCP protocols, respectively. We can observe that our scheme can detect the selfish behavior with high backhaul capacity with very high accuracy. Note that B_{cel} implies the intensity of selfish behavior, because, the larger the B_{cel}, the more outstanding packets the selfish node can transmit, which causes severe interference, as observed in Section 2.2. However, the results imply that, in many cases, the detection decision takes more than 2 s with low selfish intensity, i.e., small B_{cel}. This is because the impact of such a moderately selfish node on the network performance is not significant, i.e., the selfish node only achieves a small throughput gain over the legitimate nodes. Consequently, the moderate selfish node is not immediately detectable by CUBIA within 2 s, since it takes more samples for the AP to accurately detect such selfish nodes. Figure 10 shows the average throughput degradation ratio of well-behaving nodes due to the selfish node for various values of selfish intensity (i.e., B_{cel}). The figure implies that a higher selfish intensity incurs a severer interference on well-behaving nodes. The throughput degradation can be ignored when the backhaul link capacity B_{cel} of the selfish node is small.

Table 3. Detection performance for the UDP protocol.

$(\theta_{AS}, \theta_S)B_{cel}$	20 Mbps	10 Mbps	5 Mbps	2 Mbps
(2, 3)	1.00	1.00	0.99	0.73
(2, 4)	1.00	1.00	0.99	0.74
(2, 5)	1.00	1.00	0.97	0.61
(3, 4)	1.00	0.94	0.05	0.00
(3, 5)	1.00	0.93	0.07	0.00
(3, 6)	0.99	0.92	0.07	0.00
(4, 5)	1.00	0.27	0.00	0.00
(4, 6)	1.00	0.33	0.00	0.00
(4, 7)	0.95	0.27	0.00	0.00

Table 4. Detection performance for the TCP protocol.

$(\theta_{AS}, \theta_S)B_{cel}$	20 Mbps	10 Mbps	5 Mbps	2 Mbps
(2, 3)	1.00	1.00	1.00	0.99
(2, 4)	1.00	1.00	1.00	0.47
(2, 5)	1.00	1.00	1.00	0.03
(3, 4)	1.00	1.00	0.99	0.80
(3, 5)	1.00	1.00	0.98	0.28
(3, 6)	1.00	1.00	0.93	0.03
(4, 5)	0.87	0.86	0.11	0.01
(4, 6)	0.84	0.84	0.07	0.00
(4, 7)	0.63	0.72	0.02	0.00

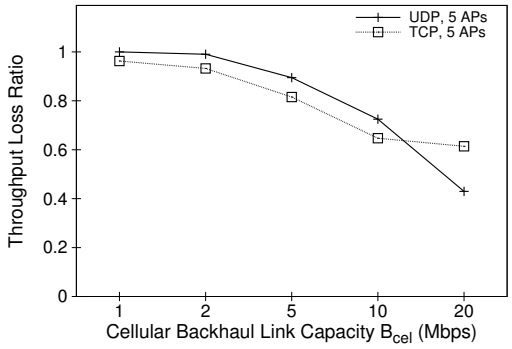
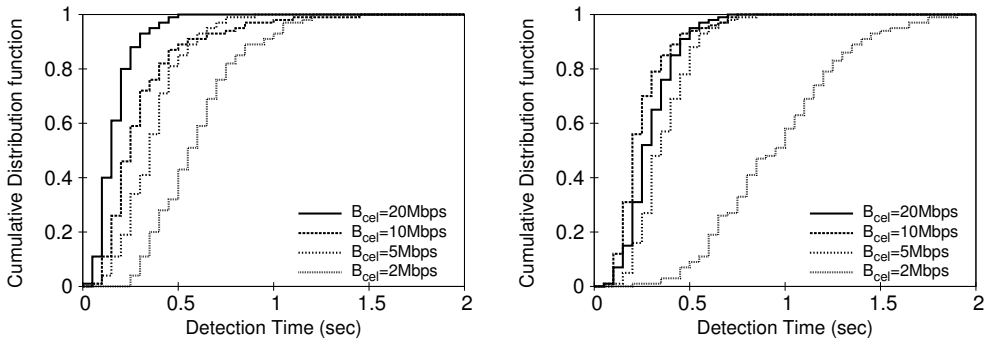


Figure 10. Impact of selfish intensity on the performance of well-behaving nodes.

We next evaluate the impact of the backhaul link capacity B_{cel} of selfish tethering on the detection performance for B_{cel} = 2, 5, 10, and 20 Mbps. Figure 11 shows the cumulative distribution of detection time under various B_{cel}. The results indicate that more aggressive selfish behavior, i.e., higher B_{cel}, is detected more quickly by CUBIA for both TCP and UDP protocols. This indicates that CUBIA can quickly detect aggressive selfish behaviors, which is a very important design requirement for any good detection scheme since such an aggressive behavior can seriously degrade the performance of well-behaving nodes.

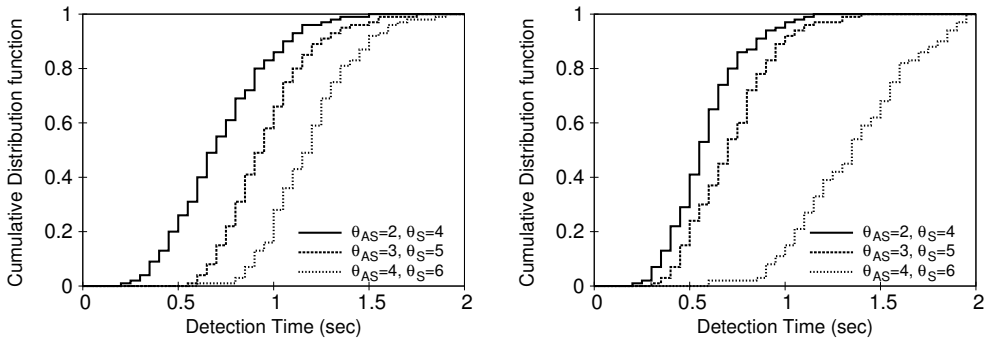


(**a**) UDP (**b**) TCP

Figure 11. Impact of B_{cel} on detection time.

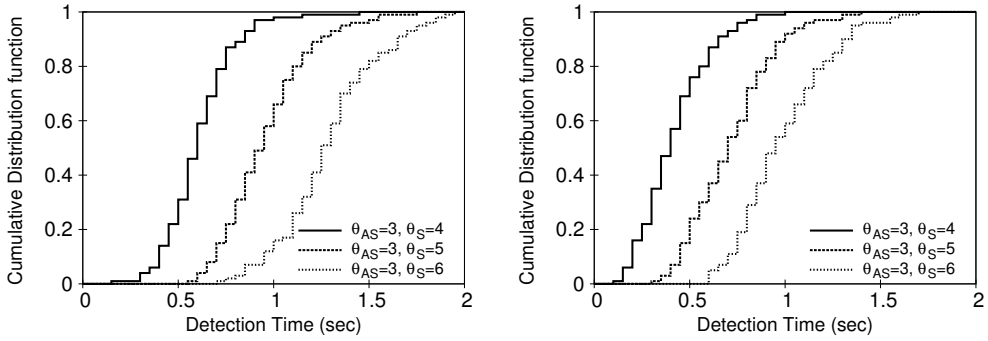
Finally, we study the impact of alarm thresholds, i.e., θ_{AS} and θ_S, on the detection time. Figure 12 depicts the distribution of detection time for three different values of the *first alarm threshold* θ_{AS}. It is straightforward that the use of a smaller value of θ_{AS} may issue the first alarm too frequently and may incur an unnecessary overhead for RTS/CTS exchange, which results in performance degradation. Meanwhile, the results show that it takes less detection time with a smaller θ_{AS}.

Similarly, the impact of the *second alarm threshold* θ_S can be seen in Figure 13. The figure compares the detection time for different values of θ_S with the given θ_{AS} = 3. We can see that the detection time increases proportional to θ_S. However, there is a tradeoff between the false alarm ratio and the detection time according to the choice of θ_S. For example, in case of the temporal behavior of the detection filter in Figure 9c, the use of a small value of θ_S can cause false alarms (if θ_S is set to be less than $\theta_{AS} + 1$, it would issue several false alarms.) although it can reduce the detection time unless the hidden node exists. Thus, it is recommended to use a value of θ_S larger than $\theta_{AS} + 1$, in order to avoid false alarms, although it might take more time to detect.



(**a**) UDP (**b**) TCP

Figure 12. Impact of the first alarm threshold θ_{AS} on detection time for UDP and TCP protocols with B_{cel} = 20 Mbps.



(**a**) UDP (**b**) TCP

Figure 13. Impact of the second alarm threshold θ_S for the given θ_{AS} = 3 on detection time with B_{cel} = 20 Mbps.

5. Related Work

Selfish and malicious misbehavior in wireless networks has been extensively studied under various scenarios in different communication layers. The majority of previous work has concentrated on detecting or punishing MAC-layer misbehavior [18,23–25]. In [18], Kyasanur and Vaidya studied the MAC-layer misbehavior of selecting always small backoff values rather than random selection, and showed that such selfish misbehavior can seriously degrade the network performance. Raya et al. [23] investigated multiple misbehavior policies in 802.11 MAC protocol, including backoff manipulations, and presented a detection framework, called DOMINO, which considers all possible strategies jointly and improves the detection accuracy. Several statistic-based frameworks for detecting misbehavior also have been introduced [24,25]. Their common detection approach is to measure the inter-arrival time of target stations in terms of the number of backoff slots and verify whether the backoff time of the stations follows a legitimate pattern or not. The authors in [26] proposed a non-parametric CUSUM test in order to detect real-time selfish misbehavior in

802.11 networks. The problem of selfish misbehavior has also been addressed in several different layers and perspectives, including the coexistence between LTE and Wi-Fi systems in unlicensed bands [27], the emerging machine-to-machine (M2M) communications [28], relay networks [29], routing layers [30,31], multi-channel protocols [32], and game theory-based behaviors [28,33]. The problem of misbehavior using the unfair coexistence between LTE-LAA and Wi-Fi system is relatively new and unexplored in the literature. In [27], the problem of detecting misbehaving of LTE/Wi-Fi has been addressed. The authors of [34] presented a new model for analyzing the throughput performance for Wi-Fi and LTE-LAA coexistence, which can act as a baseline model to be exploited to distinguish misbehaving coexistence from the legitimate behavior. In [35], the authors have identified the effect of an 802.11 node's selfish behavior by manipulating the CCA threshold. They have shown that the selfish behavior can achieve higher throughput than other well-behaving nodes that are based on a game-theoretical model. The authors in [36,37] addressed the selfish carrier sense problem in 802.11 Wi-Fi networks. The authors performed experimentation on a real-world testbed and showed that such selfish behaviors can cause extremely unfair allocations of the wireless medium. In [37], the detection of misbehavior in dense Wi-Fi networks has been addressed and a detection method that is based on a statistics has been presented. However, none of these studies considered the problem of the selfish problem exploiting Wi-Fi tethering environments.

6. Conclusions and Future Work

In this paper, we present `CUBIA`, a novel Wi-Fi tethering misbehavior detection mechanism that can accurately detect selfish behavior, e.g., manipulating CCA threshold, via AP-level collaboration in multi-AP network environments. We show that the benefits of MIM can be fully exploited and abused further by a selfish tethering node via CCA manipulation combined with its short link-distance. We also observe that the consequence of the selfish tethering behavior resembles that of the hidden node problem, but selfish tethering nodes tend to ignore the RTS/CTS mechanism. CUBIA employs a CUSUM algorithm for the online detection of abnormal network behavior and inject RTS/CTS frames to avoid mis-diagnosing the hidden node problem as a selfish tethering. Our simulation-based evaluation results show that CUBIA accurately distinguishes the selfish tethering behavior from other types of misbehavior including the hidden node problem.

In the future, we would like to extend our work to pinpoint and localize the selfish node based on the cooperation among APs. It would also be interesting to study effective system-level follow-up actions against selfish tethering misbehavior, such as jamming-resilient dynamic channel re-assignment.

Funding: This work was supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2C1013308) and the Gachon University research fund of 2019 (Grant No. GCU-2019-0776).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Access Point
CCA	Clear channel assessment
CST	Carrier sensing threshold
CUSUM	cumulative sum
FER	frame error rate
MIM	Message-In-Message
SIR	signal to interference ratio
SoI	the Signal of Interest
WLAN	Wireless LAN

γ_m *SIR threshold* for modulation scheme m
β_m *MIM threshold* for modulation scheme m
$\mathcal{T}_{FER}$ target frame error rate

References

1. Bitglass. 2018 BYOD Security Report. Available online: https://pages.bitglass.com/MissionImpossibleSecuringBYOD_LP.html (accessed on 1 August 2020).
2. Syntonic. Syntonic 2016 Employer Report: BYOD Usage in the Enterprise. Available online: https://syntonic.com/wp-content/uploads/2016/09/Syntonic-2016-BYOD-Usage-in-the-Enterprise.pdf (accessed on 1 August 2020).
3. Hoffman, C.; Summerson, C. How to Tether Your Android Phone and Share Its Internet Connection with Other Devices. Available online: https://www.howtogeek.com/170302/the-htg-guide-to-tethering-your-android-phone/ (accessed on 1 August 2020).
4. Doherty, J. *Wireless and Mobile Device Security*; Jones & Bartlett Publishers: Boston, MA, USA, 2015.
5. Khorov, E.; Kiryanov, A.; Lyakhov, A.; Bianchi, G. A Tutorial on IEEE 802.11ax High Efficiency WLANs. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 197–216. [CrossRef]
6. Aijaz, A.; Kulkarni, P. On performance evaluation of dynamic sensitivity control techniques in next-generation wlans. *IEEE Syst. J.* **2019**, *13*, 1324–1327. [CrossRef]
7. Wang, W.; Zhang, F.; Zhang, Q. Managing channel bonding with clear channel assessment in 802.11 networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
8. Choi, J.; Shin, K.G. Out-of-band sensing with ZigBee for dynamic channel assignment in on-the-move hotspots. In Proceedings of the 2011 19th IEEE International Conference on Network Protocols, Vancouver, BC, Canada, 17–20 October 2011; pp. 216–225.
9. Leentvaar, K.; Flint, J. The capture effect in FM receivers. *IEEE Trans. Commun.* **1976**, *24*, 531–539. [CrossRef]
10. Lee, J.; Kim, W.; Lee, S.; Jo, D.; Ryu, J.; Kwon, T.T.; Choi, Y. An experimental study on the capture effect in 802.11a networks. In Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, Montreal, QC, Canada, 9–14 September 2007; pp. 19–26.
11. Manweiler, J.; Santhapuri, N.; Sen, S.; Choudhury, R.; Nelakuditi, S.; Munagala, K. Order matters: Transmission reordering in wireless networks. *IEEE/ACM Trans. Netw.* **2012**, *20*, 353–366. [CrossRef]
12. The Network Simulator-ns2. Available online: www.isi.edu/nsnam/ns (accessed on 3 September 2018).
13. Wiggle: Wireless Geographic Logging Engine. Available online: http://www.wigle.net/ (accessed on 3 September 2018).
14. Garetto, M.; Salonidis, T.; Knightly, E.W. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. *IEEE/ACM Trans. Netw.* **2008**, *16*, 864–877. [CrossRef]
15. Choi, J.; Shin, K.G. QoS provisioning for large-scale multi-AP WLANs. *Ad Hoc Netw.* **2012**, *10*, 174–185. [CrossRef]
16. Zhang, X.; Shin, K.G. Adaptive subcarrier nulling: Enabling partial spectrum sharing in wireless lans. In Proceedings of the 2011 19th IEEE International Conference on Network Protocols, Vancouver, BC, Canada, 17–20 October 2011; pp. 311–320.
17. Mishra, A.; Rozner, E.; Banerjee, S.; Arbaugh, W. Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage. In Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, Berkeley, CA, USA, 19–21 October 2005; p. 29.
18. Kyasanur, P.; Vaidya, N.H. Selfish MAC layer misbehavior in wireless networks. *IEEE Trans. Mob. Comput.* **2005**, 4, 502–516. [CrossRef]
19. Gustafsson, F. *Adaptive Filtering and Change Detection*; John Wiley & Sons, Ltd.: London, UK, 2000.
20. Kim, J.; Kim, S.; Choi, S.; Qiao, D. CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs. In Proceedings of the IEEE INFOCOM 2006, Barcelona, Spain, 23–29 April 2006; pp. 1–11.
21. Jensen, T.; Kant, S.; Wehinger, J.; Fleury, B. Fast link adaptation for MIMO OFDM. *IEEE Trans. Veh. Technol.* **2000**, *59*, 3766–3778. [CrossRef]
22. Joshi, K.; Hong, S.; Katti, S. PinPoint: Localizing interfering radios. In Proceedings of the 10th USENIX NSDI, Lombard, IL, USA, 2–5 April 2013; pp. 241–254.

23. Raya, M.; Aad, I.; Hubaux, J.-P.; Fawal, A.E. DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots. *IEEE Trans. Mob. Comput.* **2006**, *5*, 1691–1705. [CrossRef]
24. Radosavac, S.; Baras, J.; Koutsopoulos, I. A Framework for MAC Protocol Misbehavior Detection in Wireless Networks. In Proceedings of the WiSE05: 2005 ACM Workshop on Wireless Security (Co-Located with Mobicom 2005 Conference), Cologne, Germany, 2 September 2005; pp. 33–42.
25. Toledo, A.; Wang, X. Robust Detection of Selfish Misbehavior in Wireless Networks. *IEEE J. Sel. Areas Commun.* **2007**, *25*, 1124–1134. [CrossRef]
26. Tang, J.; Cheng, Y. Selfish Misbehavior Detection in 802.11 Based Wireless Networks: An Adaptive Approach Based on Markov Decision Process. In Proceedings of the IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 1357–1365.
27. Samy, I.; Lazos, L.; Xiao, Y.; Li, M.; Krunz, M. LTE misbehavior detection in Wi-Fi/LTE coexistence under the LAA-LTE standard. In Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec'18), Stockholm, Sweden, 18–20 June 2018; pp. 87–98.
28. Liew, J.T.; Hashim, F.; Sali, A.; Rasid, A.; Cumanan, K. Performance Evaluation of Backoff Misbehaviour in IEEE 802.11ah Using Evolutionary Game Theory. In Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–7.
29. Szott, S.; Konorski, J. Selfish Attacks in Two-Hop IEEE 802.11 Relay Networks: Impact and Countermeasures. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 658–661. [CrossRef]
30. BenSalem, N.; Buttyan, L.; Hubaux, J.P.; Jakobsson, M. A Charging and Rewarding Scheme for Packet Forwarding in Multihop Cellular Networks. In Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc), Annapolis, MD, USA, 1–3 June 2003; pp. 13–24.
31. Pu, C.; Lim, S.; Chae, J.; Jung, B. Active detection in mitigating routing misbehavior for MANETs. *Wirel. Netw.* **2017**, *25*, 1669–1683. [CrossRef]
32. Zhang, Y.; Lazos, L. Countering selfish misbehavior in multi-channel MAC protocols. In Proceedings of the IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 2787–2795.
33. Akella, A.; Seshan, S.; Karp, R.; Shenker, S. Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP. In Proceedings of the ACM SIGCOMM, Pittsbugh, PA, USA, 19–23 August 2002; pp. 117–130.
34. Gao, Y.; Roy, S. Achieving Proportional Fairness for LTE-LAA and Wi-Fi Coexistence in Unlicensed Spectrum. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3390–3404. [CrossRef]
35. Yang, E.; Choi, J.; Lee, S. On selfish behavior using asymmetric carrier sensing in IEEE 802.11 wireless networks. In Proceedings of the 2008 33rd IEEE Conference on Local Computer Networks (LCN), Montreal, QC, Canada, 14–17 October 2008; pp. 527–529.
36. Pelechrinis, K.; Yan, G.; Eidenbenz, S.; Krishnamurthy, S.V. Detecting Selfish Exploitation of Carrier Sensing in 802.11 Networks. In Proceedings of the IEEE INFOCOM, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 657–665.
37. Afaqui, M.; Brown, S.; Farrell, R. Detecting MAC Misbehavior of IEEE 802.11 Devices within Ultra Dense Wi-Fi Networks. In Proceedings of the 2018 25th International Conference on Telecommunications (ICT), St. Malo, France, 26–28 June 2018; pp. 213–219.