On Selfish Configuration in Wi-Fi Tethering

Jaehyuk Choi, *Member, IEEE,* Alexander W. Min, *Member, IEEE,* and Kang G. Shin, *Life Fellow, IEEE*

***Abstract*—Due to the ease of setting up a Wi-Fi hotspot and the availability of readily configurable mobile operating systems, Wi-Fi tethering can be abused by misbehaving users to gain an unfair advantage in throughput performance. In this paper, we study the problem of detecting selfish nodes in Wi-Fi tethering environments. In particular, selfish nodes can manipulate carrier sensing thresholds to exploit recent advances in physical-layer concurrent transmission technologies as well as short link distance in a tethered network so as to achieve unfair throughput gains while evading detection. We propose a new MAC-layer detection algorithm based on frame sequence numbers that accurately detects selfish nodes.**

***Index Terms*—Selfish behavior, Wi-Fi tethering, IEEE 802.11, sequence number, detection algorithm.**

I. Introduction

DUE the recent rapidly growing number of smartphones and tablets equipped with Wi-Fi, Wi-Fi tethering has been gaining popularity as a convenient, on-the-move, and cost-effective wireless Internet access technology. Wi-Fi tethering refers to creation of a Wi-Fi hotspot by using the 3G/4G data connection of mobile devices, such as smartphones and tablets, to provide network services to another Wi-Fi-enabled devices [1]. The main advantage of this Wi-Fi tethering is that it provides ubiquitous, yet high-speed Internet connectivity anywhere, anytime, and even on-the-move.

However, the ease of setting up Wi-Fi tethering can pose serious performance problems to well-planned Wi-Fi networks, such as enterprise and campus networks [2]. Since the tethered Wi-Fi hotspot can potentially open up the network with an arbitrary channel number, it may interfere with nearby well-planned APs and cause performance problems inside the network. Even worse, the open and customizable nature of mobile operating systems (OSes), such as Android, can be further abused by misbehaving devices/users to gain an unfair advantage in throughput performance by manipulating the tethering function. 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 devices' performance. Therefore, detecting unauthorized and misconfigured tethering Wi-Fi is becoming a very important task in most organizations.

In this paper, we address the problem of detecting selfish nodes in Wi-Fi tethering environments, and propose a simple, yet accurate, algorithm for detecting them. A common way for a selfish user to increase his chance of channel access is to modify the operation of the 802.11 protocol or change channel-attempt parameters, such as CW_{min}, CW_{max}, IFS (inter-frame space), and the carrier sensing threshold, defined in the standard [3]. Our key finding is that tuning the carrier sense threshold, i.e., clear channel access (CCA) threshold—which is tunable for better spatial reuse—is an effective means to be abused by selfish users due mainly to the recent advances in concurrent transmission techniques, i.e., capture effect and message-in-message, and relatively short link distance in a tethered Wi-Fi system. We first study the impact of selfish misconfiguration of a Wi-Fi tethering system via extensive simulations. Based on the observations made from this study, we then propose a novel lightweight scheme for detecting selfish tethering nodes that selfishly increase their CCA thresholds.

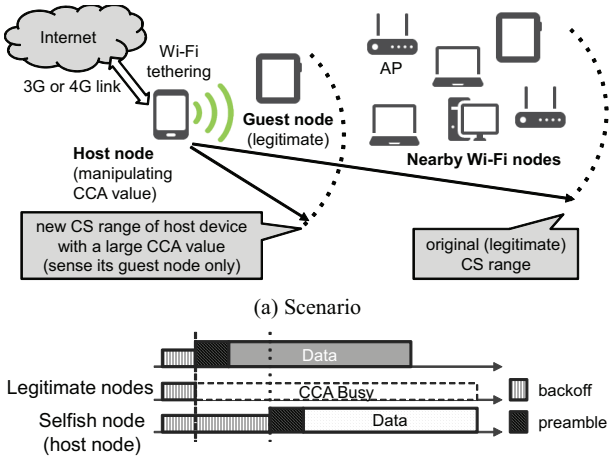


Fig. 1. Illustration of CCA threshold manipulation in Wi-Fi tethering.

II. Selfish Configuration in Wi-Fi Tethering

In this section, we study the selfish CCA tuning in a Wi-Fi tethering system that consists of a 3G/4G-capable mobile phone (e.g., smartphone) and a tethered Wi-Fi device. We will henceforth refer to the mobile device and the tethered Wi-Fi device as *host* and *guest* nodes, respectively. The host node shares its 3G/4G connection with the guest node via its Wi-Fi interface. We consider a representative scenario where a selfish user manipulates the CCA threshold of the host node's Wi-Fi interface, as shown in Fig. 1. We assume that the guest node is legitimate. By increasing this threshold, the host node decreases the communication range and becomes aware less of other concurrent transmissions. It can thus access the medium more frequently without deferring its backoff countdown even in the presence of other nodes transmissions (Fig. 1(b)). We assume that the selfish node selects the maximum possible CCA threshold without losing the tethering connectivity, which is

Manuscript received November 7, 2012. The associate editor coordinating the review of this letter and approving it for publication was G. Giambene.

J. Choi is with the Department of Software Design and Management, Gachon University, Seongnam, Korea 461-701 (e-mail: jchoi@gachon.ac.kr).

A. W. Min is with the Circuits and Systems Research, Intel Labs, Hillsboro, OR 97124 (e-mail: alexander.w.min@intel.com).

K. G. Shin is with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2121 (e-mail: kgshin@eecs.umich.edu).

This work was supported by the Gachon University Reseach Fund of 2012 (GCU-2011-R091).

Digital Object Identifier 10.1109/LCOMM.2013.040213.122488

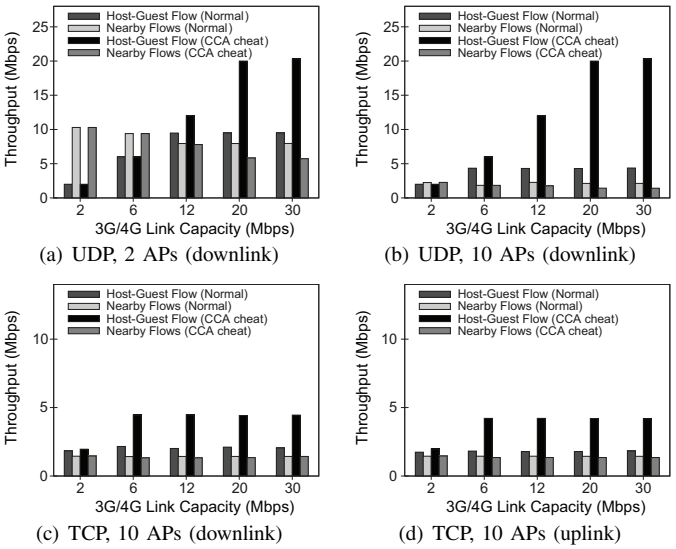


Fig. 2. Impact of selfish carrier sense on throughput of transport-layer protocols on 802.11g PHY/MAC over various 3G/4G link capacities and AP densities.

likely a little lower than the observed strength of received signals that were transmitted by the guest node.

To demonstrate the impact of this selfish CCA tuning of a Wi-Fi tethering system on the throughput performance, we conducted ns-2 simulation with transport-layer protocols, i.e., TCP and UDP, in a multi-AP environment. Our evaluation setup consists of a tethered hotspot with two communicating nodes (i.e., the host and guest nodes) which are surrounded by multiple APs (2 and 10 APs) and their associated clients (3 nodes per AP). For the simulation, we considered the typical WLAN configuration with IEEE 802.11g MAC/PHY parameters, i.e., maximum speed of 54 Mbps, and set the capacity of the 3G/4G link for tethering to 2, 6, 12, 20, and 30 Mbps.

Fig. 2 shows the average throughput of the tethering nodes (host-guest flow) and well-behaving nodes as a function of 3G/4G link capacity for low and high AP densities, i.e., 2 APs (6 AP-client flows) and 10 APs (30 AP-client flows). We compared the performance of the flows in the selfish scenarios (CCA cheat) with those of the natural mode (Normal), using UDP and TCP protocols. From the results for UDP traffic shown in Figs. 2(a) and 2(b), we can see that the misconfigured tethering flow achieves a significant throughput gain over the well-behaving flows. Its performance is also found close to the maximum achievable throughput, i.e., bandwidth of the bottleneck link—3G/4G or Wi-Fi link capacity, regardless of the number of contending nodes.[1]

Figs. 2(c) and 2(d) show the average throughput for TCP downlink and TCP uplink traffic, respectively. Note that the TCP is a bidirectional transfer-based protocol and the CCA function is manipulated only on the host side, i.e., the guest is legitimate. Nevertheless, the guest with the misconfigured tethering achieves a high throughput gain. This is attributed to the closed-loop TCP-ACK mechanism, that is, 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.

[1]Note that when 3G/4G capacity is larger than 20 Mbps, the maximum throughput is bounded by the Wi-Fi link capacity in this simulation setting.

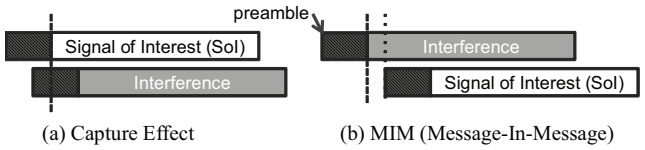


Fig. 3. Physical (PHY) capture and MIM: MIM-capable receiver can decode a strong signal of interest (SoI) even if it arrives later than the interference [5].

TABLE I
RATIO OF SUCCESSIVE RECEPTIONS AT THE RECEIVER (20MBPS BANDWIDTH OF 3G/4G)

number of APs	Tx w/o collision	Capture effect	MIM
2 APs (6 clients)	41.6	17.1	41.3
10 APs (30 clients)	20.7	22.6	56.7

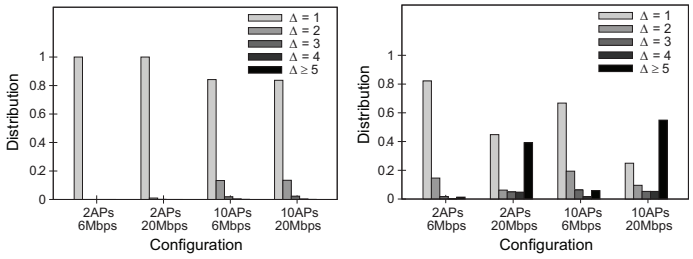
It is worth mentioning that this unfair channel access opportunity with CCA manipulation does not guarantee higher throughput to the selfish node [4] because the transmissions initiated when other transmissions are in progress can result in collisions at the receiver. However, our key observation is that, in a tethering environment, the received signal strength at the receiver is sufficiently larger than the sum of interferences due to short link distance (less than 10 m/30 ft [1]),[2] which enables the receiver to successfully capture the signal of interest (SoI) even when multiple independent transmissions occur simultaneously. Moreover, the receiver can decode the SoI even when the signal arrives after the receiver has already locked on to the interference signal. Such PHY-layer phenomena are mainly due to two well-known concurrent transmission techniques: capture effect and message-in-message (MIM) [5]. MIM is an enhanced PHY-layer capability that enables a receiver to decode an SoI even if the SoI arrives later than the interference. Fig. 3 illustrates the main difference between PHY capture and MIM. Table I shows the percentage of successful receptions at the selfish tethering node's receiver (i.e., guest node) in the simulation with 20 Mbps 3G/4G link capacity performed in Figs. 2(a) and 2(b). We can see that a very large portion of the successful receptions is due to MIM; 41.3 % and 56.7 % for low and high AP density, respectively. It is also observed that the high percentage (i.e., 17.1 % and 22.6 %) of successful reception is attributed to the PHY capture effect.

The results imply 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.

III. A Simple Detection Mechanism

Many existing solutions [3] for detecting selfish behavior in wireless networks employ behavior-based anomaly detection. Their common approach is to monitor the inter-packet arrival time or backoff timeslot distribution, and validate whether the behavior follows the legitimate pattern or not. However, in the case of selfish carrier sensing that exploits the tethering environment, it is infeasible for a monitor node to accurately obtain such metrics (i.e., inter-arrival time or backoff timeslots), because not all successful transmissions of the selfish

[2]In general, a tethered hotspot is formed for communication between personally owned devices, which are placed closely while in use, and thus the link distance between the communicating devices is relatively short, implying a high SINR at the receiver.



(a) Legitimate Node (b) Selfish Node

Fig. 4. Pattern of inter-frame sequence number gaps of legitimate and selfish nodes under various tethering conditions (100 sec simulations).

node are observable at the monitor node. That is, if the selfish node transmits a packet while the channel is busy, the packet (SoI) is successfully captured only by its receiver node thanks to MIM (thus gaining an unfair advantage in throughput). Other neighboring nodes (including the monitoring node)—that are already engaged in receiving a different signal—may interpret the SoI as an interfering signal, without being able to decode it (thus failing to detect the existence of CCA cheating). This is because the signal strength of the SoI at other nodes may not strong enough to disengage from the current reception and re-engage the SoI, where the required SINR for MIM is relatively high (e.g., $>$ 10 dBm [5]). Thus, if a monitor node keeps track of the transmission of the host node, it may not be able to fully capture the misbehavior of the tethering hotspot; the successful transmissions due to MIM are not visible to the monitor node, whereas a large portion of the successful transmissions from the selfish node are due to MIM, as shown in Table I.

Based on the above observation, we propose a new, yet simple, detection mechanism that runs at a monitoring node (AP or client) to determine whether a node in the communication range is misconfigured or not. We use MAC-layer information, in particular, the sequence number field in the 802.11 MAC header. The sequence number of each frame from a node is incremented by one whenever the node sends out a frame and its value is modulo 4096 [6].

Let ΔS_i denote the inter-frame sequence number gap between the i-th frame and the (i-1)-th frame, i.e., $\Delta S_i = s_i - s_{i-1}$, where s_i is the sequence number of the i-th transmitted frame. The difference ΔS_i for a legitimate node should differ by one modulo 4096. Our key idea to identify the misconfigured node is that, since the successful transmissions due to MIM are not visible to a monitor node, the inter-frame sequence number gap ΔS_i between two successively observed frames coming from the selfish node is likely to be greater than 1. In other words, if a node is legitimate, the probability $Pr(\Delta S_i \geq 2)$ is relatively small, and vice versa. Fig. 4 compares the pattern of inter-frame sequence number gaps for legitimate and selfish nodes.

Note that the gap ΔS_i of a legitimate tethering node also can be greater than 1 due to collisions, when the collision is captured at the receiver (i.e., guest node), but is not visible to the monitor node. Therefore, we check the distribution with a threshold parameter θ, i.e., $Pr(\Delta S_i \geq 2) > \theta$, to determine whether the node is legitimate or not. The decision parameter θ reflects the network condition; we adjust the threshold θ according to the collision probability of the network at a given time. Algorithm 1 describes the detection procedure of the monitor node.

Algorithm 1 Detection Algorithm for node x

procedure $OnEventRecvFrom(x)$
1: // Upon receipt of i^{th} frame P_i from node x
2: s_i = sequence number of P_i
3: $\Delta S_i = (s_i - s_{i-1}) \bmod 4096$
4: $C_x[\Delta S_i] = C_x[\Delta S_i] + 1$
5: // For every K samples, perform misbehavior test (K=100)
6: **if** $(i \bmod K) == 0$ **then**
7: $\theta = f$(estimated number of active nodes) // collision prob.
8: **if** $C_x[1]/K \leq 1 - \theta$ **then**
9: node x is marked as a selfish node
10: **end if**
11: reset $C_x[k] = 0$ for all k
12: **end if**

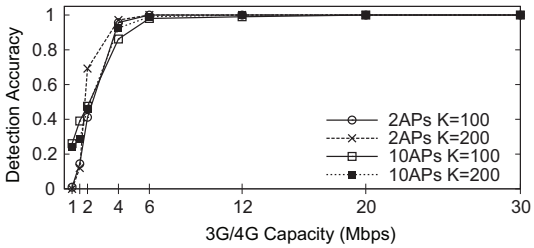


Fig. 5. Detection accuracy against 3G/4G capacity.

This algorithm is evaluated under various configurations. Fig. 5 illustrates the accuracy of our algorithm over various 3G/4G capacities for different AP densities and the detection sample sizes (i.e., K). The results show that our algorithm detects the misbehavior with high accuracy, especially for high 3G/4G capacity, demonstrating that the inter-frame sequence number gap is a simple, yet efficient metric for validating the correctness of CCA configuration.

IV. Conclusion

In this paper, we investigated the problem of detecting selfish nodes in Wi-Fi tethering environments. To the best of our knowledge, this is the first attempt addressing the selfish node problem in a tethering environment. We showed 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 proposed a MAC-layer sequence number-based detection algorithm that can effectively detect selfish CCA.

References

[1] J. Choi, and K. G. Shin, "Out-of-band sensing with ZigBee for dynamic channel assignment in on-the-move hotspots," in *Proc. 2011 IEEE ICNP*.
[2] H. Dwivedi, C. Clark, and D. Thiel, *Mobile Application Security*. McGraw-Hill, 2010.
[3] A. L. Toledo, and X. Wang, "Robust detection of selfish misbehavior in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1124–1134, Aug. 2007.
[4] K. Park, J. Choi, K. Kang, and Y. Hu, "Malicious or selfish? Analysis of carrier sense misbehavior in IEEE 802.11 WLAN," in *Proc. 2009 QSHINE*.
[5] J. Manweiler, N. Santhapuri, S. Sen, R. R. Choudhury, S. Nelakuditi, and K. Munagala, "Order matters: transmission reordering in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 353–366, Apr. 2012.
[6] IEEE 802.11 WG, IEEE Std 802.11-2007 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-1999, 2007.