Journal of Computer Science 4 (11): 951-958, 2008
ISSN 1549-3636

Enhancing Wireless Medium Access Control Layer Misbehavior Detection System in IEEE 802.11 Network

Ali Mohammed Alsahag and Mohamed Othman
Department of Communication Tech and Network, University Putra, Malaysia
43400 UPM Serdang, Selangor D.E., Malaysia

Abstract: Wireless Medium Access Control (MAC) protocols such as IEEE 802.11 use distributed contention resolution mechanisms for sharing the wireless channel. In this environment, selfish hosts that fail to adhere to the MAC protocol may obtain an unfair throughput share. For example, IEEE 802.11 requires hosts competing for access to the channel to wait for a "back-off" interval, randomly selected from a specified range, before initiating a transmission. Selfish hosts may wait for smaller back-off intervals than well-behaved hosts; thereby obtaining an unfair advantage. We show in this thesis that a greedy user can substantially increase his share of bandwidth, at the expense of the other users, by slightly modifying the driver of his network adapter. This study is a complementary of DOMINO System model to enhance the detection system in the MAC layer of IEEE 802.11; our enhanced system is a piece of software to be installed in or near the Access Point. The system can detect and identify greedy stations without requiring any modification of the standard protocol. We illustrate these concepts by simulation results.

Key words: MAC, IEEE 802.11, misbehavior, wireless LAN, hotspot

INTRODUCTION

IEEE 802.11[1] wireless LANs were originally meant to be deployed in (relatively) protected locations such as corporate offices; as a result, security, billing, and guarantee of fair access received limited attention. But, over the last few years, IEEE 802.11 has also become the dominating solution for hotspots, which provide public wireless access to the Internet. Furthermore, the increased level of sophistication in the design of protocol components, together with the requirement for flexible and readily reconfigurable protocols has led to the extreme where wireless network adapters and devices have become easily programmable. As a result, it is feasible for a network peer to tamper with software and firmware, modify its wireless interface and network parameters and ultimately abuse the protocol

In[2], Wireless Medium Access Control (MAC) protocols such as IEEE 802.11 use distributed contention resolution mechanisms for sharing the wireless channel.

We show in this study that a greedy user can substantially increase his share of bandwidth, at the expense of the other users [3], by slightly modifying the driver of his network adapter. We explain how easily this can be performed, in particular with the new generation of adapters. We then present a new enhanced system to detection greedy behavior in the MAC layer IEEE 802.11, a piece of software to be installed in the Access Point. The new system can detect and identify greedy stations, without requiring any modification of the standard protocol at the AP and without revealing its own presence. We illustrate these concepts by simulation results.

Related works: Deviation from legitimate MAC Layer protocol operation in wireless networks has received considerable attention from the research community in recent years. Recent research has investigated misbehavior at the network layer [4-6] in wireless networks. One approach is to identify misbehaving nodes and avoid such nodes in routing [7]. Another approach is to design protocols that encourage cooperation by penalizing misbehavior. Network layer mechanisms address network layer misbehavior such as tampering with route discovery/maintenance, dropping, delaying or misrouting packets.

MacKenzie and Wicker [8] study the problem of selfish users in Aloha from a game-theoretic point of

Corresponding Author: Mohamed Othman, Dept of Communication Tech and Network, University Putra Malaysia, 43400 UPM, Serdang Selangor D.E., Malaysia Tel: +603-89466535 Fax: +603-89466577

view. They assume however that all nodes have the same transmission rates and costs

In [9], Cagalj study the scenario of multiple cheaters in an ad hoc network and use game theory to devise optimal cheating strategies. Although their research addresses issues similar to the ones we tackle here.

Konorski[9,10], studies selfish MAC layer misbehavior, where hosts deviate from the specified backoff strategy. Konorski's study assumes that all hosts can accurately measure the duration and originator of each black-burst, which is hard to guarantee in a wireless network and as it requires a new backoff mechanism, different from the current standard, this solution is not practical for current hotspots. Kyasanur and Vaidya their research was an important source of inspiration for our study. Witch proposed a modification to the IEEE 802.11 MAC protocol to facilitate the detection of selfish and misbehaving nodes.

We are present a complementary for DOMINO System model which is enhancing the MAC layer detection system that avoid the modification to the IEEE.802.11 MAC protocol. We present our enhanced system for detecting MAC misbehavior in a way that is transparent to the operation of the network.

IEEE802.11 MAC misbehavior: In the Distributed Coordinating Function (DCF) of the IEEE 802.11 MAC protocol, coordination of channel access for contending nodes is achieved with carrier sense multiple access with collision avoidance (CSMA/CA).

If the channel is perceived to be busy in one slot, the backoff counter freezes. After the backoff counter is decreased to zero, the transmitter can reserve the channel for the duration of data transfer. First, it sends a Request-To-Send (RTS) packet to the receiver, which responds with a Clear-To-Send (CTS) packet. Thus, the channel is reserved for the transmission. Both RTS and CTS messages contain the intended duration of data transmission in the duration field. Other hosts overhearing either the RTS or CTS are required to adjust their Network Allocation Vector (NAV) that indicates the duration for which they will defer transmission. An unsuccessful transmission instance due to collision or interference is denoted by a lack of CTS or acknowledgment (ACK) for the data sent and causes the value of the contention window to double. If the transmission is successful, the host resets its contention window to the minimum value W. IEEE 802.11 DCF favors the node that selects the smallest backoff value among a set of contending nodes. Moreover, due to the exponential increase of the contention window after each unsuccessful transmission, non-malicious nodes are forced to select their future backoffs from larger intervals after every access failure. Therefore, the chance of their accessing the channel becomes even smaller. Although several other deviation strategies exist, this one is the most challenging to detect, and in this study we adhere to protocol deviations that occur due to manipulation of backoff value.

MAC greedy behavior: Selectively scramble frames sent by other stations in order to increase their contention windows. The frames to be targeted can be the following:

CTS frames: In this case the cheater hears an RTS frame destined to another station and intentionally causes collision and loss of the corresponding CTS frame in order to prevent the subsequent long frame exchange sequence (RTS/CTS handshake is used for large frames). As a result, the channel becomes idle after the corrupted CTS and the cheater gets a chance to send its data .ACK and DATA frames: Although this does not result in saving the data frame transmission time, it causes the contention window of the ACK destination (i.e., the DATA source) station to be doubled and consequently makes the latter select larger backoffs. As before, the cheater increases its chances to get access to the channel.

Manipulate protocol parameters: When the channel is idle, transmit after SIFS but before DIFS. When sending RTS or DATA frames, increase the included NAV value in order to prevent the stations in range from contending during this time. Reduce the back-off time this can be done by choosing a small fixed contention window; thus, the backoff is always chosen from this small window. A cheater may also combine several of the above techniques or adaptively change its misbehavior to avoid being detected. We will address this type of cheating.

Security attacks: This category of attacks (e.g., the deauthentication attack) exploits security weaknesses of the MAC protocol (such as flaws in authentication or encryption mechanisms) and targets the access control, confidentiality, or availability of the network. They may be rational or malicious. As this category has been

extensively addressed before, we will not consider it further in this paper. In this study "misbehavior" means greedy behavior of stations and does not relate to the security aspects of wireless networks.

System components: In this section, we present the way to detect the misbehavior techniques by our enhanced system as bellow. The complete detection system is depicted below in Fig. 1 This system has to be implemented only at the AP.

Monitoring period: To avoid overloading the AP with per-frame computations, the data required for detection are collected during configurable intervals of time; at the end of each interval, the detection mechanism is run. Another advantage of this method over a per-frame detection approach is the ability to collect more statistical data and hence increase the accuracy. In addition, the binary exponential backoff algorithm of IEEE 802.11 is unfair in the short term. This would result in false positives if stations were monitored over short term periods even in the absence of misbehavior. Therefore the monitoring period has to be large enough to achieve long term backoff fairness. Taking into account the typical bit rates, monitoring periods have to be accurate to prevent the cheater from gaining large benefits before being detected. The monitoring period has been chosen in our enhanced system is proven by the simulations to avoid false positives. The gathered data are then passed to several tests within the encapsulating System algorithm:

Loop:

```
If monitoring period elapsed since last check then
  for each active station Si do
    for j = 1 to 6 do
      execute Test j
```

The tests described below make use of the following function where x indicates the test number.

Check$_x$ (S_i, condition$_x$):

```
If conditionx is true then
  cheat_countx (Si):= cheat_countx (Si) + 1
  If cheat_countx (Si) > Kx then
    Si is misbehaving
    Call the punishing function
else if cheat_countx (Si) > 0 then
  cheat_countx (Si):= cheat_countx (Si) – 1
```

To decrease the number of false positives, a station should be suspected at least K_x times (i.e., after at least Kx monitoring periods, as defined in Fig. 1) before being considered a cheater. In addition, each time a station does not cheat, its cheat count$_x$ is decremented (until it reaches zero) to reward the correct behavior.

Scrambled frames: The cheater has to scramble a relatively large percentage of CTS, ACK, or DATA frames sent by other stations.

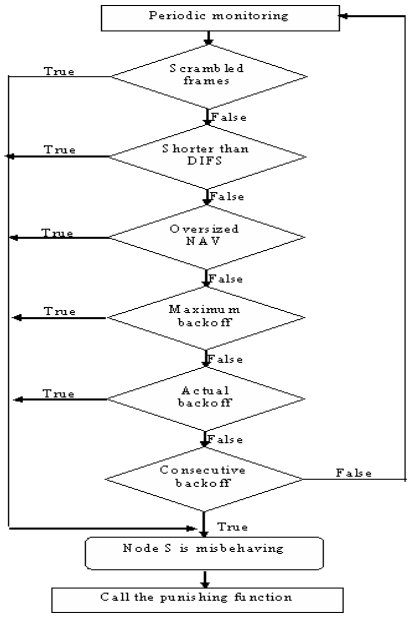

Fig. 1: System Components

As a result, its average number of retransmissions will be less than that of other stations, and it can be detected using Test 1 num_ rtx(S) is the number of times station S retransmitted its last frame.

Test 1 Scrambled frames:

Condition1 := num_rtx(Si)< φ ×Ej≠i[num_rtx(Sj)]

Call check1(Si,condition1)

The system can detect a retransmission by observing a repeated sequence number in the header of RTS or DATA frames when the corresponding CTS or ACK frames are scrambled, respectively.

Detection of manipulated protocol parameters: In the following paragraphs we address misbehavior techniques that alter protocol parameters. We focus mainly on backoff manipulation since it is the easiest to implement and the hardest to detect.

Shorter than DIFS: The AP can monitor the idle period after the last ACK and distinguish any station that transmits before the required DIFS period. After having observed this misbehavior repeatedly for several frames from the same station, the AP can make a reliable decision (Test 2).

Test 2 Shorter than DIFS:

Condition2: = idle_time_after_ACK (Si)<DIFS
call check2 (Si, condition2)

Oversized NAV: By measuring the actual duration of a transmission (including the DATA, ACK, and optional RTS/CTS) and comparing it with the NAV value in the RTS or DATA frame headers, the AP can detect stations that regularly set the NAV to very large values. In Test 3, the tolerance parameter A (greater than 1) ensures that the AP does not mistakenly catch well-behaved stations.

Test 3 Oversized NAV:

condition3 := NAV (Si) > A×tx_duration(Si)
call Check3(Si , condition3)

Maximum back-off: Since the IEEE 802.11 protocol selects backoffs randomly from the range [0; CW-1] (Where CW depends on the number of retransmissions), the maximum selected backoff over a set of frames sent by a given station (without inter leaving collisions). Otherwise the contention window will be doubled) should be close to [CW_{min-1}] if the number of samples is large enough. The maximum backoff test (Test 4) uses this property to suspect stations whose maximum backoff over a set of amplest is smaller than a threshold value threshold max_{bkf}. Clearly, a tradeoff exists between the number of samples and the threshold; if we increase the threshold (its largest value is CW_{min}), we have to increase the number of sampled backoffs to get more distinct values and thus avoid false positives. In our simulations, we use a threshold equal to CWmin/2; thus, the test works if the reduced contention window is in [0; $CW_{min/2 - 1}$].

Test 4 Maximum backoff:

condition4:= max_bkf (Si)<$threshold_{maxbkf}$
call Check4 (Si, condition4)

Actual Back-off: This test (Test 5) consists in measuring the actual backoff as shown in Fig. 2. The main procedures of the test can be summarized as follows:

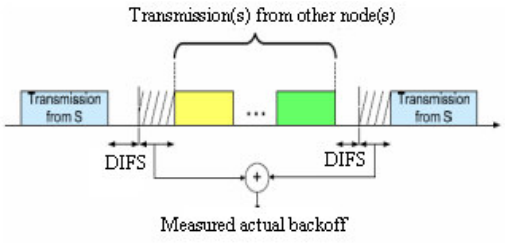

Fig. 2: Actual back-off

If between two transmissions from a station S there are no collisions, we assume that S spent all its idle time backing off (although it may be just part of the S's interframe delay). Then we estimate this backoff by computing the sum.

If a collision happens, it is not possible to know the identities of the senders of the colliding frames and hence the station who's measured actual backoff should be updated. To avoid complexity, collisions are simply not taken into account and both the current backoff and the next one are not measured for any station. Stations that hear frame headers with wrong CRC, caused by a collision, will defer their transmissions by EIFS (Extended InterFrame Spacing). This latter does not interfere with the measurements since all deferrals of all nodes are not taken into account after a collision.

Test 5 Actual backoff:

condition5:= B_{ac} [Si] < $\alpha_{ac} \times B_{acnom}$
call Check5 (Si, condition5)

In Test 5, B_{ac} [Si] denotes the average actual backoff (observed by the AP) of station Si. B_{acnom} is the nominal backoff value, which is equal to the average backoff of the AP if it has enough traffic to compute this value; the inbound traffic from the AP is usually larger than the outbound traffic). If the AP does not have enough data to derive a nominal backoff value from its own traffic, it uses an analytical value E [B_{ac}]. We do not use the analytical value in the first place since it depends on the number of active stations and is computed assuming backlogged sources.

The α_{ac} ($0<\alpha_{ac}\leq 1$) parameter is configurable according to the desired true positive (correct detection) and false positive (wrong detection) percentages (e.g., we use σac = 90% in our simulations). To reduce false positives, we use K5 = 3 (defined in the function $check_x$) in our simulations; this shows that this value can be small enough to quickly detect cheaters without accusing well-behaved stations. As it collects no data during collisions, the actual back off test measures

backoffs that are from the [0; CW_{min-1}] range. Due to its mechanism, this test fails to detect the misbehavior case when the cheater has interframe delays (e.g., a TCP source using congestion control). In fact, the test measures these delays instead of backoffs because it adds up the idle periods between transmissions from the same source. Hence, although the chosen back-offs may be subject to cheating, the monitor will not be able to measure them correctly; the solution to this problem is provided by the consecutive backoff test.

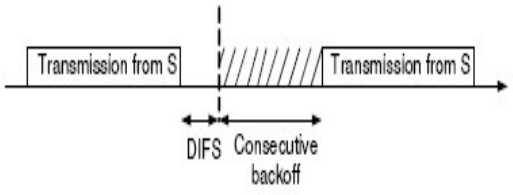

Fig. 3: Consecutive back-off measurement

Figure 3 illustrates this test (Test 6), which works in the case of sources with inter frame delays. In practice, this is mainly the case of TCP sources (in this case the delay is due to the congestion control of TCP), which represent over 91% of traffic in real networks. The actual backoff test for these sources does not yield the correct values (as explained in the previous paragraph), and consequently cannot detect potential cheating. Let us consider a station S sending TCP traffic and being monitored by the system algorithm. We assume that there is enough traffic from other sources on the common channel such that, between two frames sent by S and separated by a transport layer delay, there is at least one interleaving frame from another station. Hence, if the AP observes two consecutive non-interleaved frames from S, it can consider the idle time between them as only a back off in addition to the mandatory DIFS. These consecutive frames are the result of channel contention that may force S to queue packets at the MAC layer even if they were separated by a delay at upper layers. In this situation, S would benefit from cheating with backoff in order to free its MAC layer queue. Thus, the system can collect significant samples of the backoff values chosen by S; we call these samples consecutive back-offs.

Test 6 Consecutive backoff:

condition6:= B_{ac} [Si] < $\alpha_{ac} \times B_{acnom}$
call check6 (Si, condition6)

MATERIALS AND METHODS

In order to study the performance of the proposed solution, we have used ns-2 with the Monarch project extension[11] to simulate our detection system while reducing false positives, the AP has to gather enough statistical data and then make decisions based notably on average values. Therefore, it needs to measure one or more attributes of the transmitting stations based on the default parameters value are in Table 1. In this section, we identify two such attributes, namely throughput and backoff**.**

Table 1: Parameters for DSSS

DIFS	50µs
SIFS	10 µs
Slot Time	20 µs
ACK	112bits+PHY header=203 µs
RTS	160bits+PHY header=207 µs
CTS	112bits+PHY header=203 µs
DATA	MAC header (30b)+DATA (0-2312b)+FCS(4b)
Timeouts	300-350 µs
CWmin	32 time slots
CWmax	1024 time slots

Throughput: Although throughput seems to be the most intuitive metric for distinguishing stations using higher shares of the channel bandwidth than other stations, this metric would face several obstacles if used for detection.

Backoff: as we aim notably at detecting backoff manipulation, backoff measurement is the most direct way to detect cheaters (the next section explains how the AP estimates the backoff chosen by a station by monitoring the channel idle time). It is less dependent than throughput on various factors, some of which have been discussed before.

RESULTS AND DISCUSSION

As the frame scrambling misbehavior is fairly easy to detect using the number of retransmissions, this section examines in detail only the back off manipulation tests and the complete detection mechanism. Although these tests are capable of detecting multiple cheaters as shown in Fig. 4, in the simulations we have focused on the case of a single cheater to simplify the presentation of the results

Simulation topology: We covered these scenarios that represent common traffic types.

UDP traffic: Besides the cheater, there are seven stations sending CBR traffic (the nominal rate is 500 bytes/packet, 200 packets/s).

The cheater is also a CBR source. The cheating technique consists in decreasing the contention window. In any idle slot, there is at least one packet ready for transmission by any of the competing stations.

The time elapsed between two transmissions from the same station (interleaved with transmissions from other stations) is therefore due only to the back off chosen by the IEEE 802.11 protocol.

Results are averaged over 10 simulations, 110s each. The monitoring period is set to 20s, which also corresponds to one decision (cheater or well-behaved) by the AP regarding each station.

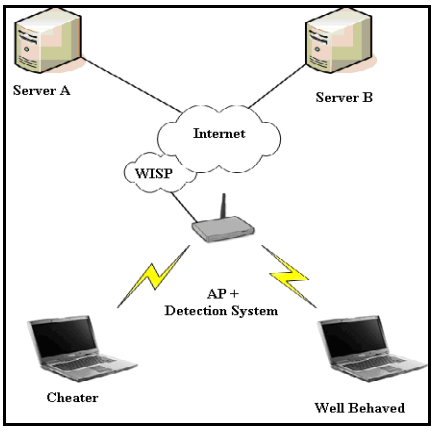

Fig. 4: Network Topology

Thus, each point on the following graphs is averaged over 100 samples with a 95 percent confidence interval; the first 10s of each simulation is an initialization period, where measurements are taken according to the formula (1- CW_{min}/CW_{max})) where the CWmin is a dynamic size. in our simulations we have been working on the following contention window sizes 29,25,21,17,13,9,5,1 and the maximum contention window is driven from the equation 2n-1 where n=5. In the following, the misbehavior coefficient represents the amount of misbehavior. A misbehavior coefficient equal to m means that the corresponding station uses a fixed contention window equal to (1-m)×CW min and then chooses its back off from this new window. Thus, m = 0 means no misbehavior, and m =1 means that the station transmits without any back off.

Figure 5 shows the increasing of throughput for the cheater on the expense of well behaved users by disobeying to the MAC protocol rules and gaining a higher bandwidth that will affect the other users and may cause the denied of service. The developed system resolved the problem via periodically collects traffic traces of active user stations during short intervals of time called monitoring period. A series of tests, each aiming at detecting a particular misbehavior technique, determines if the analyzed traffic presents behavior anomalies. The outputs of these tests are then fed into a Decision Making Component (DMC) that decides whether a given station is cheating. If so, the control is passed to the misbehavior handling mechanism that, as mentioned before, is dependent on the WISP policy. The tests as well as the decision making components will make the use of the bandwidth among the whole users equally as Fig. 6 showed.

TCP Traffic: Each of the eight stations runs an FTP application; one station is cheating by jamming TCP packets and forging the corresponding MAC-ACKs.

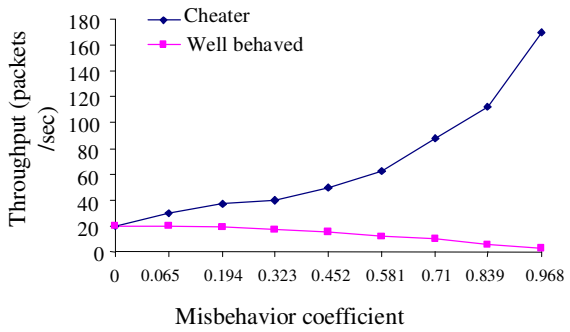

Fig. 5: Throughput of cheater in UDP Traffic

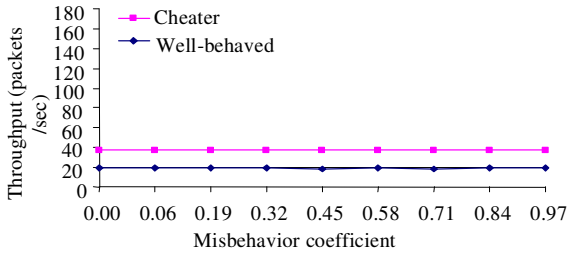

Fig. 6: Improved throughput of UDP

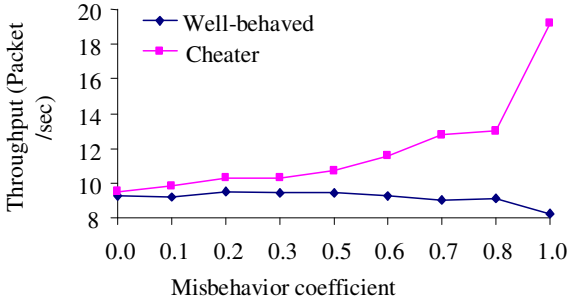

Fig. 7: Throughput of cheater in TCP traffic

This case illustrates the effect of inter-frame delays (due to TCP congestion control) on back off measurement. This is the most realistic scenario. In both cases, the AP generates traffic similarly to one station, i.e., CBR in the first case and FTP in the second to take into account the fading effects present in real channels. We have used the shadowing channel model.

Figure 7 shows that the throughput of the cheater is slightly equal to the well behaved throughput means that our system can detect the misbehavior of the cheater and send him to the penalty functions in order to get the same fairness among the stations.

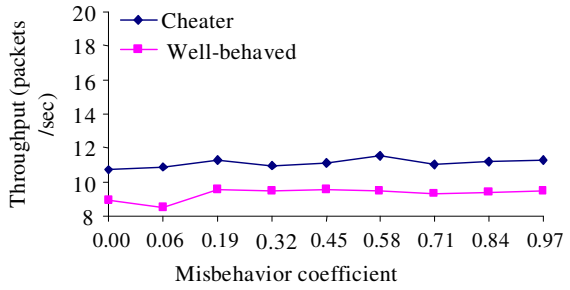

Fig. 8: Improved throughput of TCP traffic

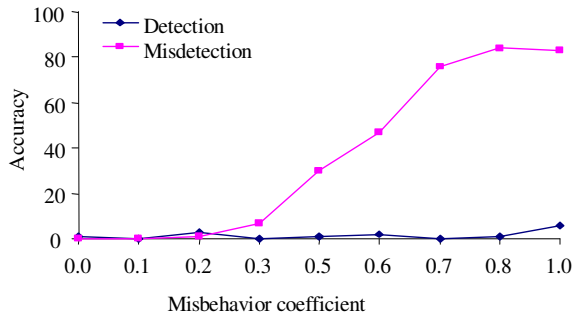

Fig. 9: Actual back-off in UDP traffic type

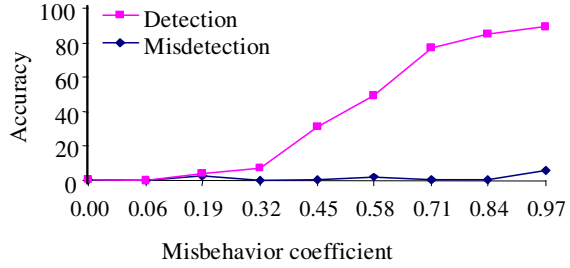

Fig. 10: Consecutive back-off In TCP traffic type

Figure 8 In addition to the TCP congestion control mechanisms and the dependence of the TCP throughput, our system can perform well and can detect the misbehavior for the greedy user.

Actual Back off: From the simulation graphs 10 and 11 we can draw the following observations. In the UDP traffic case, the test performs well, as in Fig. 9, because there is always at least one frame ready for transmission by each station. Hence, the channel idle time between two transmissions from a station is the result of only the back off mechanism (in addition to DIFS).

Consecutive back-off: The performance of this test differs from that of the previous one and confirmed by simulations. In the TCP traffic case, the test yields good results, as Fig. 10 shows. This is due to the presence of other sources that do not allow the source with the inter-frame delay (induced by congestion control) to transmit two frames consecutively without having queued the second one, i.e., the delay does not affect the idle time between two consecutive non-interleaved transmissions from the source. Otherwise, if there is no frame ready in the queue, another source takes control over the channel and transmits at least one frame between two successive frames of the first source.

CONCLUSION

MAC layer misbehavior in IEEE 802.11 networks can lead to severe unfairness in bandwidth distribution. This can become a serious problem in public Internet access hotspots where individual users have to pay for network usage and hence may be motivated to cheat in order to increase their share of the medium. Once a greedy user has implemented an attack, he can make it available on a web site, thus jeopardizing the proper operation of many wireless networks around the globe. In spite of its relevance, this topic is still relatively unexplored in the research community. Handling MAC layer misbehavior is an important requirement in ensuring a reasonable throughput share for well-behaved hosts in the presence of misbehaving hosts. In this thesis, we have classified MAC layer misbehaviors, presented enhanced detection techniques, and provided the corresponding detection mechanisms. In contrast with previous researcher that have proposed modifications to the MAC protocol, thus requiring a modification of existing wireless cards, we have

developed a solution that can be completely integrated in the AP and uses only statistical data analysis. Hence, the main features of the proposed solution are its efficiency and applicability to real networks. Simulation results have indicated that our system provides fairly accurate misbehavior diagnosis.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my supervisor Assoc. Prof. Dr. Mohamed Othman, for his encouragement and guidance throughout this study. I would also like to thank Dr. Maxim Raya who moderated me in completing this study successfully.

REFERENCES

1. IEEE Std. 802.11, Standards Committee "Wireless LAN-Medium Access Control (MAC) and Physical (PHY) Layer Specification", IEEE Standard 802.11, 1999.
2. Bianchi, G., 2000. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Select. Areas Comm., 18: 535-547.
3. Raya, M., J.P. Hubaux and I. Aad, 2004. DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots. In Proceedings of the ACM Conference on Mobile Systems, Applications and Services (MobiSys), Boston, MA, USA. pp: 84-97. http://doi.acm.org/10.1145/990064.990077
4. Hu, Y., A. Perrig and D. Johnson. Ariadne, 2002. A secure ondemand routing protocol for ad hoc networks. In The 8th ACM International Conference on Mobile Computing and Networking, MobiCom, pp: 12-23.
5. Yang, H., X. Meng and S. Lu., 2002. Self-Organized Network Layer Security in Mobile Ad Hoc Networks. In ACM MOBICOM Wireless Security Workshop (WiSe'02).
6. Zhou, L. and Z. J. Haas., 1999. Securing ad hoc networks. IEEE Network, 13: 24-30. Doi: 10.1109/65.806983
7. Marti, S., T.J. Giuli, K. Lai and M. Baker., 2000. Mitigating routing misbehavior in mobile ad hoc networks. In Mobile Computing and Networking, (MCN'02), pp: 255-265. http://www.hpl.hp.com/personal/Mary_Baker/publications/mitigating.pdf
8. MacKenzie, A.B. and S.B. Wicker, 2003. Stability of Multipacket Slotted Aloha with Selfish Users and Perfect Information. Proc. Infocom. 3: 1583-1590. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1209181
9. Cagalj, M., S. Ganeriwal, I. Aad and J.P. Hubaux, 2005. On Selfish Behavior in CSMA/CA Networks. In Proceedings of the 24th Annual joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005).pp: 2513-2524. Doi: 10.1109/INFCOM.2005.1498536
10. Konorski, J., 2002. Multiple Accesses in Ad-Hoc Wireless LANs with Noncooperative Stations. NETWORKING 2002. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications": Second International IFIP-TC6 Networking Conference, Pisa, Italy, May 19-24, 2002. Proceedings. pp: 1141. http://www.springerlink.com/content/6d981hq4j3cewkqf/
11. Fall, K. and K. Varadhan, 2003. NS Notes and Documentation. UC Berkeley, LBL, USC/ISI, Xerox PARC, number of pages (380), USA. http://www.ecse.rpi.edu/Homepages/shivkuma/teaching/fall2002/ns-2/ns_doc.pdf