Published in 2008 IEEE Wireless Communications and Networking Conference (DOI 10.1109/WCNC.2008.244). ©2008 IEEE

Impact of Contention Window Cheating on Single-hop IEEE 802.11e MANETs

Szymon Szott, Marek Natkaniec, Roberto Canonico, and Andrzej R. Pach, *Members, IEEE*

***Abstract*— This paper presents a work in progress which deals with the important and unresolved problem of node misbehavior. A realistic approach is used to determine the impact of contention window manipulation on IEEE 802.11e ad-hoc networks. It is explained why such networks are more prone to misbehavior. Novel results pertaining to the 802.11e standard are presented. Simulation analysis is done for several scenarios with a distinction made for uplink and downlink traffic. It is shown that a misbehaving node can jeopardize network performance, therefore, countermeasures to this problem need to be developed.**

***Index Terms*— Ad-hoc networks, IEEE 802.11e, misbehavior, QoS, contention window**

I. Introduction

Mobile ad-hoc networks (MANETs) are based on the principle of cooperation, with each node acting as both terminal and router. The performance of the network depends on how well the participants of the network collaborate with each other. The threat of misbehavior arises when nodes (or rather: the users controlling them) decide to maximize their own benefit rather than work together as a group. These gains can be measured for example in terms of throughput or battery life. The detection and mitigation of such behavior is important for the functioning of the ad-hoc network.

Currently, the IEEE 802.11 family of standards is most often being used to deploy MANETs. However, the MAC layer provided by these standards was designed for cooperation. Nodes contend for the medium using a distributed mechanism, which assumes that all participants behave properly. As will be shown, one of the simplest ways to misbehave within this mechanism is to modify the Contention Window (CW) selection algorithm.

The IEEE 802.11e standard [4] was developed to provide Quality of Service (QoS) provisioning at the MAC layer. This is achieved through a new distributed channel access mechanism: Enhanced Distributed Channel Access (EDCA)[1]. It separates traffic into four access categories (AC) of different priority. Each category is differentiated by its own set of parameters (in particular those related to the contention window). The 802.11e standard allows for the easy modification of these AC parameters. In practice this can be easily done using a WLAN card based on the Atheros chipset and the madwifi driver [9]. A malicious user may want to exploit this feature for his own benefits. Since the IEEE standard contains no incentive mechanisms for the users to behave properly, the degree of misbehavior does not have to be subtle. Therefore, the misbehaving user does not have to be careful in the scale of his cheating (e.g., he can set the contention window to the lowest allowable value).

This paper aims to answer the following questions related to contention window misbehavior in 802.11e mobile ad-hoc networks: Is this type of misbehavior easy and beneficial to perform? What is its impact on QoS provisioning? What behavior can we expect from a malicious user? Are the user's gains dependent in terms of transport protocol used and network size? Are the misbehaving user's gains the same for both uplink and downlink traffic?

The rest of the paper is organized as follows. A description of the IEEE 802.11 standard and the contention window mechanism can be found in Section II. Section III presents and discusses the state of the art. The simulation scenarios are described in Section IV and their results – in Section V. Section VI concludes the paper and describes the future work.

II. The IEEE 802.11 Standard

The IEEE 802.11 standard [3] defines a distributed access method for wireless networks – DCF (Distributed Coordination Function). This is the basic access method in ad-hoc mode. It is based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). This access method is enhanced with Virtual CS (Virtual Carrier Sense) and NAV (Net Allocation Vector).

The 802.11 MAC protocol distinguishes three important time periods: SIFS, PIFS, DIFS (Short-, PCF-, and DCF- Inter Frame Space) which have lengths corresponding to the following rule: DIFS>PIFS>SIFS. When stations sense that the medium is free, they begin to measure these periods in order to estimate when they can begin their own transmission. The protocol also identifies three priorities of transmission, according to these periods.

The contention window algorithm works as follows. Each node, ready to transmit, senses the medium to determine whether it is idle. If so, it begins to transmit. Otherwise, since the channel is busy, the node waits for the current transmission

This work has been carried out under the EU-funded NoE CONTENT project no. 038423.

Szymon Szott, Marek Natkaniec and Andrzej R. Pach are with the Department of Telecommunications of AGH University of Science and Technology (emails: {szott, natkanie, pach}@kt.agh.edu.pl).

Roberto Canonico is with the Consorzio Interuniversitario Nazionale per l'Informatica - University of Napoli (email: roberto.canonico@unina.it).

[1] IEEE 802.11e also defines another access method: HCF Controlled Channel Access (HCCA). However, this method is suited only for infrastructure networks and therefore beyond the scope of this paper.

to finish and then waits until the medium has been free for one DIFS period. Afterwards, it randomly chooses a backoff value from the range [0, CW]. The chosen value denotes the time slot in which the node will begin its transmission. This decreases the probability that two nodes will transmit simultaneously and thus cause a collision. The countdown of the backoff value is paused when the channel is busy. When the backoff reaches zero, the node may transmit. At the beginning, the parameter CW is equal to a predefined value CWmin. After each collision, CW is doubled until it reaches another predefined value – CWmax. A successful transmission resets CW to the value of CWmin.

The IEEE 802.11e standard [4] introduces EDCA as the new distributed channel access mechanism. Traffic is divided into four access categories to provide appropriate QoS. These categories are, from the highest priority: Voice (Vo), Video (Vi), Background (BK), and Best effort (BE). Each category has its own set of parameters: AIFS (Arbitration InterFrame Space), TXOP (Transmission Opportunity), and, in particular, CWmin and CWmax (Table I). These parameters are responsible for traffic differentiation.

TABLE I
VALUES OF CW PARAMETERS IN 802.11E

AC	CWmin	CWmax
Voice	7	15
Video	15	31
Background	31	1023
Best effort	31	1023

The medium contention rules for EDCA are similar to 802.11 DCF. The difference in channel access prioritization is shown in Fig. 1 and Fig. 2. Each frame arriving at the MAC layer is mapped, according to its priority, to an appropriate AC. There are four transmission queues, one for each AC. AIFS[AC] is the parameter which replaces the DIFS of DCF. An internal collision resolution mechanism (*virtual collision*) is used to determine which frame can be sent. A physical collision can still occur, when two or more nodes start their transmissions simultaneously.

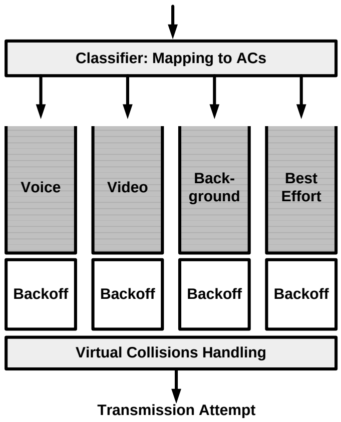

Fig. 1 Mapping to access categories [4]

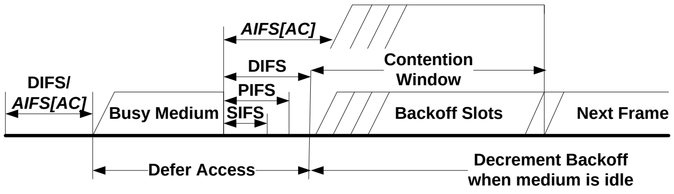

Fig. 2 Channel access prioritization [4]

III. STATE OF THE ART

One of the first papers dealing with the problem of contention window misbehavior was [7] (later extended in [8]). The authors take into account several misbehavior strategies, such as selecting a smaller backoff (from the range [0, CW/4]), having a fixed backoff (1 slot) or not doubling the CW. It was the first paper to report degraded throughput in 802.11 infrastructure networks. The authors proposed an algorithm to solve this problem, under the assumption that the receiver (802.11 Access Point) is well-behaved. In their approach, it is the receiver, not the sender which chooses the random backoff value. This value is transferred to the sender in either a CTS or ACK frame. Misbehavior occurs when the sender deviates from that backoff. The penalty assigned by the receiver is a higher backoff value in subsequent transmissions. The problem with this approach, other than requiring changes to the 802.11 standard, is that it is unsuitable for ad-hoc networks, where the receiver cannot be trusted. Hidden nodes also cause a problem in terms of determining the correct backoff.

Several works in the field were written by Baras et al: [1], [2], and [11]. In [2], an algorithm (named ERA-802.11) for ensuring randomness in ad-hoc networks is proposed. It is based on the negotiation of CW parameters by sender and receiver (inspired by a protocol for flipping coins over the telephone). This assures a truly random backoff. The detection system developed in [5] is used to monitor nodes. In the case of misbehavior, a report is sent to an external reputation management system. ERA-802.11 introduces extra messages so it is not compatible with the 802.11 standard.

The problem of trying to detect CW cheating is how to correctly observe the chosen backoff of another node. Observations are hindered by such factors as: interference from other transmissions, unsynchronized clocks, and non-deterministic medium access. It is also necessary to determine when to stop the observation and make a decision. This problem is discussed in [11]. The authors take into account an adaptive attacker and prove that a particular decision rule, the sequential probability ratio test (SPRT), is the optimal approach to minimizing the number of needed observations. Similar work was done in [13].

ICMAC [1] is a TDMA-based MAC protocol robust to contention window misbehavior. Through a game theoretic approach and the use of the Vickrey auction mechanism, the authors have managed to provide incentives for the nodes to cooperate. However, the TDMA nature of this protocol makes it more complicated to use in ad-hoc environments.

Paper [12] presents DOMINO, an advanced software application designed to protect hotspots from greedy users. It monitors traffic, collects traces and analyzes them to find anomalies. DOMINO can detect many types of malicious and greedy behavior, including backoff manipulation techniques. Anomaly detection is based on throughput (instead of observed backoff), which the authors acknowledge is not an optimal detection metric. The application can be seamlessly integrated with APs and it complies with standards. However, it cannot be used in ad-hoc networks.

Theoretical research pertaining to backoff attacks in ad-hoc networks has been published by Konorski (e.g., in [6]). A game theoretic approach is used to provide incentives for proper cooperation. A strategy is proposed which provides fair and efficient bandwidth use.

To summarize, research efforts have so far been mostly focused on detecting nodes cheating on backoff in 802.11 infrastructure scenarios. Ad-hoc networks pose a challenge because they are distributed and have no centralized authority. Thus, there have not been many papers discussing contention window cheating in MANETs. Furthermore, the 802.11e QoS extension allows for the easy modification of MAC parameters, as stated in the previous section. Therefore, the subsequent sections address these issues.

IV. Simulation Scenarios

The purpose of the simulations is to determine the impact of CW cheating on ad-hoc network performance. The potential benefits of a misbehaving node are measured for UDP/TCP traffic in both uplink and downlink directions.

The simulation analysis was performed in the ns2 simulator with the TKN EDCA model [14] to allow the use of the 802.11e standard. All stations were within hearing range of each other. Table II presents the various simulation parameters used.

TABLE II
Simulation Parameters

Parameter	Value
MAC Protocol	802.11b + 802.11e
Data rate	11 Mb/s
Basic rate	1 Mb/s
Routing protocol	None
Transport protocol	UDP and TCP
Node distribution	Random
Traffic generator	CBR
Frame size	1000 B
Frame exchange	DATA-ACK

Two different network topologies were considered, for the uplink and downlink scenarios. In the uplink scenario, the number of homogenous nodes in the ad-hoc network was set to 5, 25, and 100 to represent small, average and large network sizes, respectively. The per-station offered load changed from 64kb/s to 8Mb/s. The node distribution was random and the traffic pattern – circular (with each node sending and receiving exactly one traffic stream). An example topology, for 5 nodes, can be seen in Fig. 3.

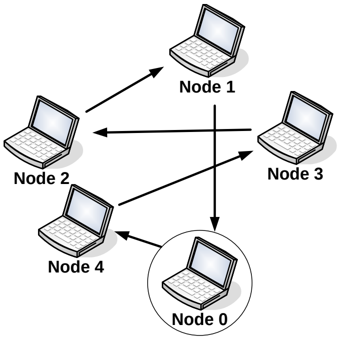

Fig. 3 Network topology (uplink scenario)

Within each uplink scenario, there was one misbehaving node (e.g., the encircled node in Fig. 3). Out of the four traffic classes of 802.11e, the background priority was used by all nodes. The well behaving (*good*) nodes had unaltered contention window parameters: CWmin = 31, CWmax=1023. The misbehaving (*bad*) node had these parameters significantly decreased: CWmin = 1, CWmax = 5. It seems realistic that the misbehaving node would choose such low (or lower) parameters to maximize its gain. The effect of choosing other CWmax values is studied further on.

In the downlink scenario, a different network topology was considered (Fig. 4). There was one misbehaving node (encircled in the figure) and three well-behaving nodes, all within hearing range of each other. Measuring UDP traffic is pointless because the misbehaving node has no means of influencing it in the downlink direction. For TCP, however, the *bad* node sends TCP-ACK packets so it has some influence on the rate of the received data. Therefore, two TCP flows, with an offered load of 8Mbit/s each, were used to put the network in a state of saturation. The downlink throughput was measured with the misbehavior either on or off.

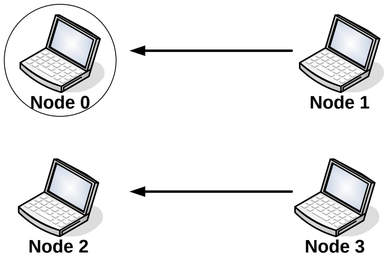

Fig. 4 Network topology (downlink scenario)

V. Results

The results of the uplink simulations are presented in the following figures. The plots present the curves, where the error of each simulation point for a 95% confidence interval does not exceed 2% (this is too small for graphical representation). The throughput of nodes as a function of the offered data rate is presented in Fig. 5, Fig. 6, and Fig. 7 for 5, 25 and 100 total nodes in the network, respectively. These figures illustrate the throughput of the misbehaving node compared to the average throughput of the well-behaving nodes and the average throughput in a case where there are no misbehaving nodes present. Fig. 8 presents the average frame delay of the misbehaving and well-behaving nodes in the small network scenario. In the case of no misbehaving nodes present, the results match the delay of *good* nodes. The delay was similar for larger simulated networks, therefore only this figure is being presented.

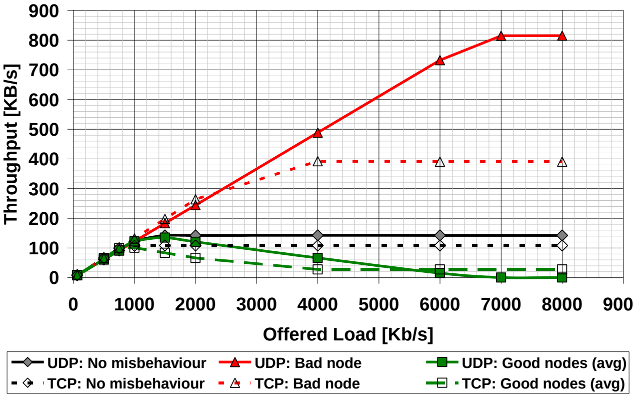

Fig. 5 Throughput vs. offered load (total no. of nodes: 5)

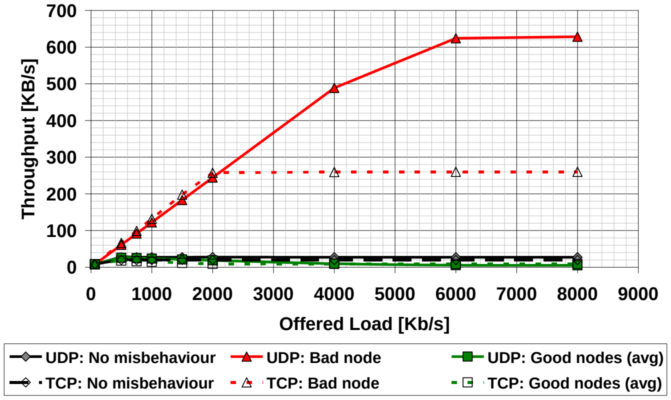

Fig. 6 Throughput vs. offered load (total no. of nodes: 25)

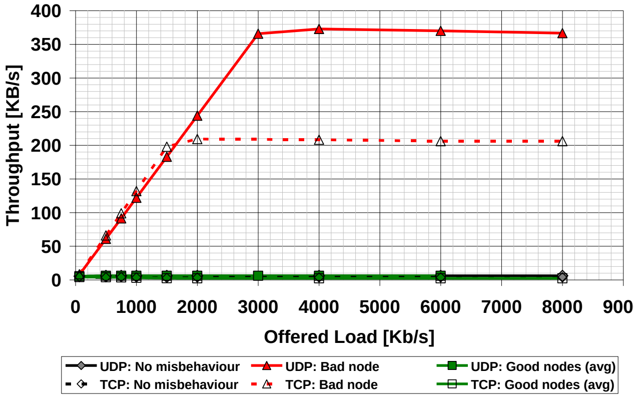

Fig. 7 Throughput vs. offered load (total no. of nodes: 100)

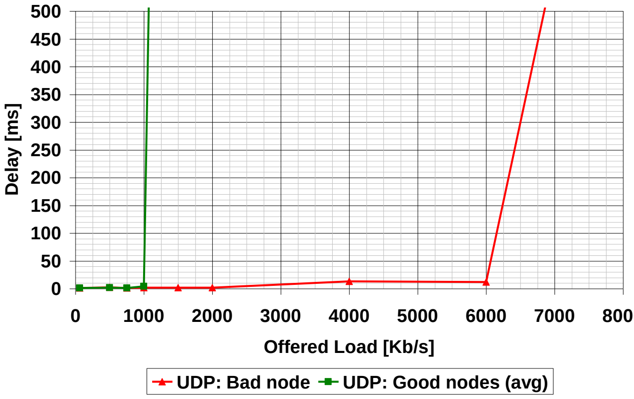

Fig. 8 Average frame delay vs. offered load (total no. of nodes: 5)

The main conclusion from these figures is that the misbehaving node can easily dominate the network in terms of throughput and delay. This occurs once the network reaches congestion (i.e., the network would have been saturated if it consisted only of well-behaving nodes). Until that point the *bad* node's presence is not harmful (Fig. 5). After reaching congestion, the *bad* node increases its throughput at the cost of the *good* nodes until saturation is achieved, in which the *bad* node has much more throughput than the average *good* node. The type of transport protocol used has no influence on this behavior, although throughput is, of course, generally lower for TCP than UDP. This is related to the TCP congestion control mechanisms and the dependence on the TCP ACK packets, which are sent to the *bad* node by a well-behaved receiver. The total number of nodes in the network only limits the maximum throughput of the misbehaving node, otherwise, the behavior is similar.

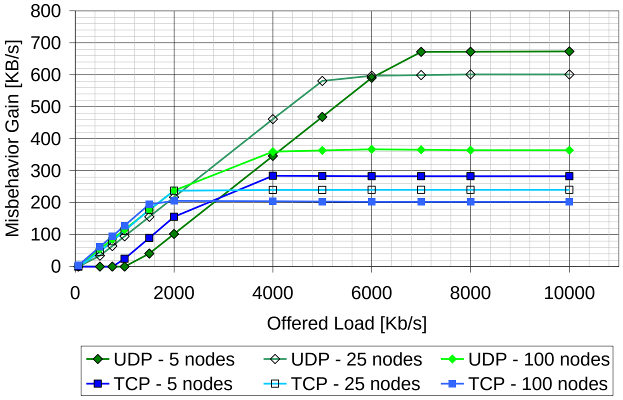

Fig. 9 Throughput gain of misbehaving node

The throughput gain of the misbehaving node in absolute values is presented in Fig. 9. This gain is calculated as the difference between *bad* node throughput and average throughput in a scenario with no misbehavior. The points where the network saturates differs for transport protocol (TCP, UDP) and network size (5, 25, 100 nodes). Nonetheless, misbehavior is profitable – the cheating node always experiences an increase in uplink throughput.

In the presented results, the misbehaving node used the following CW parameters: CWmin = 1, CWmax = 5. The choice of CWmax can of course be challenged. Further simulations were performed to determine the impact of the choice of CWmax. Table III presents the results in the form of the misbehaving node's throughput.

TABLE III
IMPACT OF CWMAX ON BAD NODE THROUGHPUT (IN KB/S)

Nodes	CWmax		
	1	5	31
5	763	755	754
25	676	623	604
100	507	367	265

Increasing the CWmax parameter (to 31) only makes a difference in the large network scenario. It can be assumed that the misbehaving node will want to choose the lowest possible CWmax to maximize its gain. In order to test this limit, the value of CWmax = 1 was also simulated. It further increased the misbehaving node's throughput, although again, this increase was mostly visible for the largest network size. As mentioned before, there is no incentive in the 802.11 standard for the user to use only a subtle form of cheating.

Another important and interesting question concerns the impact of misbehavior on higher priority traffic. Can a node, misbehaving with the use of the parameters of a lower priority Access Category (e.g., BK), take away throughput from a higher AC (e.g., Vo)? To answer this question, a modified version of the previous 5 node scenario (Fig. 3) was simulated. The four *good* nodes were sending traffic of the highest priority – Voice. The misbehaving node continued to use the Background priority. Two situations were simulated, with the *bad* node's misbehavior turned off and on. The achieved throughput, with respect to the offered load, is shown in Fig. 10. In the first situation (represented by the unbroken lines), the *good* (Vo) nodes get all the throughput, while the throughput of the *bad* (BK) node is reduced. This is in line with how the EDCA mechanism and the assignment of Access Categories are expected to work. The dashed lines in the figure represent the case when the *bad* node modifies its CW parameters as in the previous scenarios (i.e., CWmin = 1, CWmax = 5). It can now obtain a significantly higher throughput then before, even higher than the Vo priority nodes. The difference between this scenario and the previous one is that the misbehaving node is not able to dominate the channel in the presence of Vo priority nodes (at least with contention window manipulation), as it was possible in the presence of other BK priority nodes. It can be inferred that, despite the fact that the Voice priority is the highest, it does not matter which AC the misbehaving node will manipulate – it is always able to benefit it terms of throughput. This kind of network behavior can further influence the decision of a potentially malicious user to take advantage of the benefits of misbehavior.

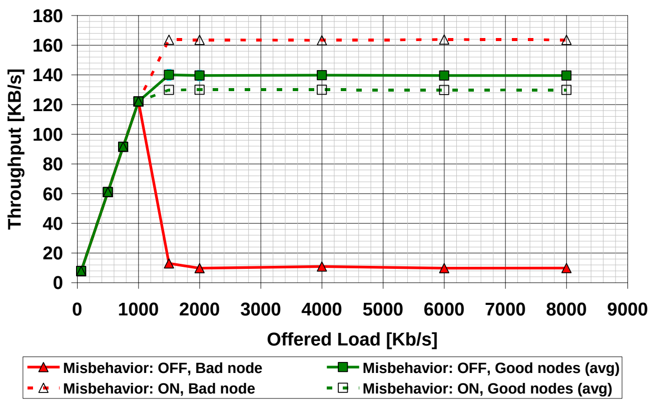

Fig. 10 Throughput vs. offered load for BK vs. Vo priority scenario

In the downlink scenario (Fig. 4) only TCP traffic was simulated. The misbehaving node could only influence the sending of TCP-ACK packets, by changing the CW values as in the previous scenario (to CWmin = 1, CWmax = 5). Fig. 11 presents the throughput results for misbehavior turned on and off.

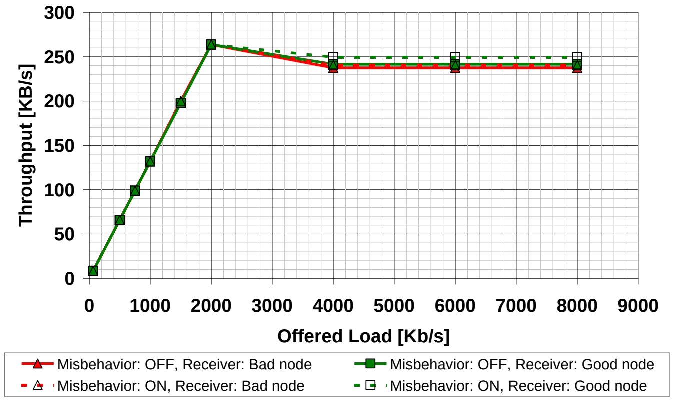

Fig. 11 Downlink throughput

The result is that with misbehavior turned on, the throughput of the *bad* node increases by an extremely small amount. The rapid sending of TCP-ACK packets increases the number of collisions in the channel but does not yield any substantial increase in the rate of the sender. This is a situation in which misbehavior does not bring beneficial results.

VI. CONCLUSIONS

This paper has presented the impact of contention window misbehavior on single-hop ad-hoc networks. Throughput, delay and fairness were considered for TCP and UDP. Two scenarios, for uplink and downlink traffic, were simulated. The assumed misbehavior model was a rational one: the malicious user would perform simple actions to obtain significant gains.

The main conclusion is that CW misbehavior leads to severe unfairness in the uplink direction. The misbehaving node can dominate uplink traffic in terms of both throughput and delay, therefore receiving substantial benefits from its actions. Misbehavior is always profitable, however, the increase in throughput is higher for UDP than TCP and more significant for smaller network sizes. The domination of the misbehaving node jeopardizes the whole ad-hoc network since other nodes receive so little throughput. Such behavior also creates exposed nodes which can be a severe problem for multihop networks.

It has been observed that the increase in throughput of a misbehaving node occurs only when the network is in saturation. Therefore, any future analysis should be limited to such scenarios. In non-congested networks, a node's misbehavior, though theoretically observable, has no influences on its neighbors and is therefore harmless.

The IEEE 802.11e standard is very prone to misbehavior – it allows easy modification of MAC layer parameters and does not provide any incentives to behave properly. A misbehaving user can choose the lowest possible CWmin and CWmax values to achieve the best performance. This paper has also shown that 802.11e fails to provide Quality of Service in the face of contention window cheating. Misbehavior allows a user's lower priority traffic to outperform the higher priority traffic of others.

The analyzed downlink scenario showed that the misbehaving node cannot significantly influence the rate of the sender, even with TCP traffic. It has also demonstrated that in some cases acts of misbehavior may not be advantageous. This is an important observation. The aim of a malicious user may be to increase the download throughput (e.g. of an FTP transfer), however, no considerable gain can be achieved using contention window cheating.

Future work will take an even more realistic approach. Studies will focus on complex real life scenarios and applications (e.g., p2p applications in multihop ad-hoc networks). The impact of the number of misbehaving nodes in these scenarios will be taken into account. The most likely forms of misbehavior also need to be determined. Simple, straightforward and advantageous actions which can be performed by any casual user, not just an expert hacker, need to be considered.

An architecture to counteract the influence of misbehavior on 802.11e networks will be developed. Its aim will be to detect the most plausible types of bad node behavior and respond through refusing to cooperate with such nodes. This will be incorporated with previous research regarding misbehavior in the networking layer [5].

The 802.11e standard plays an important role in future research. What are the benefits of manipulating the other parameters introduced by this standard? What are the limits of setting these values that still enable network operation? Is it possible to detect misbehavior with 4 different traffic categories? Can detection mechanisms be tailored to 802.11e requirements? Resolving these questions will also be the aim of further studies.

REFERENCES

[1] N. BenAmmar, J. S. Baras, "Incentive compatible medium access control in wireless networks", Proceeding From the 2006 Workshop on Game theory For Communications and Networks (GameNets '06), Pisa, Italy, October 14 - 14, 2006.

[2] Cardenas, A. A., Radosavac, S., and Baras, J. S, "Detection and Prevention of MAC Layer Misbehavior for Ad Hoc Networks", Technical Report, 2004.

[3] IEEE 802.11 Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specification, New York, IEEE Inc. (1999.

[4] IEEE 802.11e-2005, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

[5] M. Grega, Sz. Szott, P. Pacyna „Collaborative Networking with Trust and Misbehavior – A File Sharing Case", First Workshop on Operator-assisted (Wireless Mesh) Community Networks, 18-19 September 2006, Berlin, Germany.

[6] J. Konorski, "A Game-Theoretic Study of CSMA/CA Under a Backoff Attack," IEEE/ACM Transactions on Networking, vol.14, no.6, pp.1167-1178, Dec. 2006.

[7] P. Kyasanur, N.H. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," dsn, p. 173, 2003 International Conference on Dependable Systems and Networks (DSN'03), 2003

[8] P. Kyasanur, N.H. Vaidya, "Selfish MAC Layer Misbehavior in Wireless networks", IEEE Transactions on Mobile Computing, Volume 4, Number 5, September/October 2005.

[9] MADWiFi – Multiband Atheros Driver for WiFi, http://madwifi.org

[10] M. Raya, I. Aad, J.P. Hubaux, A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots", IEEE Transactions on Mobile Computing, Dec. 2006.

[11] S. Radosavac, J. S. Baras, I. Koutsopoulos, "A Framework for MAC Protocol Misbehavior Detection in Wireless Networks", In Proc. 4th ACM workshop on Wireless security (WiSe), Cologne, Germany, September 2005.

[12] M. Raya, J. Hubaux, I. Aad, "DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots", Proceedings of the 2nd international Conference on Mobile Systems, Applications, and Services (MobiSys '04), Boston, MA, USA, June 06 - 09, 2004.

[13] Y. Rong, S.-K. Lee, H.-A. Choi, "Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis," INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings , April 2006.

[14] S. Wiethölter, M. Emmelmann, C. Hoene, A. Wolisz "TKN EDCA Model for ns-2", Technical Report TKN-06-003, Telecommunication Networks Group, Technische Universität Berlin, June 2006.