Cheating on the CW and RTS/CTS Mechanisms in Single-hop IEEE 802.11e Networks

Szymon Szott, Marek Natkaniec, Andrzej R. Pach

AGH University of Science and Technology,
Department of Telecommunications,
Kraków, Poland
{szott, natkanie, pach}@kt.agh.edu.pl

Abstract. This paper presents a work in progress which deals with the problem of node misbehaviour in ad-hoc networks. A realistic approach is used to determine the impact of contention window manipulation and RTS/CTS cheating. It is explained why IEEE 802.11e ad-hoc networks are more prone to misbehaviour. The paper presents simulation results related to the mentioned types of misbehaviour. The analysis is performed for several distinct scenarios, which yields novel results. It is shown under which conditions a misbehaving node can gain a significant advantage over well-behaving nodes. The limitations of the IEEE 802.11e standard in providing QoS in the presence of misbehaving nodes is also presented.

Keywords: Ad-hoc networks, IEEE 802.11e, misbehaviour.

1 Introduction

With the increasing popularity of wireless connectivity in mobile devices (laptops, PDAs, cell phones, etc.) there is a need for interconnecting these devices in a spontaneous manner. Mobile ad-hoc networks (MANETs) are networks built without infrastructure in which every node acts as both terminal and router. Thus, they rely on the cooperation of nodes to ensure the proper functioning of the network. A problem arises if a node decides not to cooperate with others. We call such actions *misbehaviour*. A node may decide to misbehave in order to gain certain measurable profits (such as higher throughput, increased battery life). Misbehaviour is always done at the cost of the well-behaving nodes in the network. Therefore, it would be beneficial if such actions were, if not made impossible, then at least discouraged.

The problem of node misbehaviour is strengthened by the fact that the current WLAN standards (the IEEE 802.11 family) do not contain any incentives for nodes to behave accordingly. The 802.11 standards are all based on the notion that each node will strictly adhere to them. However, new wireless drivers [8] enable easy modification of MAC layer parameters. Section 2 describes the 802.11 standard (in particular the QoS extension – 802.11e) and shows to what forms of misbehaviour the standard is prone to.

The focus of this paper is put on two types of misbehaviour in ad-hoc networks. One of them is contention window (CW) cheating. This means modifying the parameters introduced in the 802.11 standard (CW_{min} and CW_{max}), which are responsible for channel access. This, and other different aspects of misbehaviour in MANETs, has already been addressed in the literature (Section 3). However, the proposed solutions do not take many aspects into account. One particular aspect is the RTS/CTS mechanism (normally used to avoid the hidden node problem) and its influence on network performance in the presence of misbehaving nodes. This is related to the second type of misbehaviour discussed in this paper – cheating on the RTS/CTS mechanism. A node may decide on not using this mechanism, even though other nodes in the network do.

In this paper we show the results from several simulation scenarios (Sections 4 and 5). We try to answer the following questions: How does CW cheating impact network performance (throughput, delay, and fairness) when RTS/CTS is used? Is this affected by the network size? Is cheating on the RTS/CTS mechanism beneficial for the misbehaving user? Should it be used alone or together with CW cheating? How do these two types of misbehaviour impact the QoS provisioning mechanisms of 802.11e? The authors of the paper prove that a rational misbehaving node will choose the lowest possible CW parameters as they are the most beneficial. The most innovative contribution of this paper is the study of RTS/CTS cheating. To the authors' best knowledge, this has not been done before.

2 Misbehaviour in the 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).

In the context of DCF, the 802.11 MAC protocol distinguishes two important time periods: SIFS and DIFS (Short- and DCF- Inter Frame Space), the latter is longer. The lengths of both of these times are defined in the standard. 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 choice of the time period depends on the frame type.

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 to finish and then waits until the medium is 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 CW_{min}. After each collision, CW is doubled until it

reaches another predefined value – CW_{max}. A successful transmission resets CW to the value of CW_{min}.

The IEEE 802.11e standard [4] introduces EDCA (Enhanced Distributed Channel Access) as the new distributed channel access mechanism. Traffic is divided into four access categories (AC) to provide appropriate QoS. These categories are, from the highest priority: *Voice* (Vo), *Video* (Vi), *Best effort* (BE), and *Background* (BK). Each category has its own set of access parameters: AIFS (Arbitration InterFrame Space), TXOP (Transmission Opportunity), and, in particular, CW_{min} and CW_{max} (Table 1). These parameters are responsible for traffic differentiation.

Table 1. Values of CW parameters in 802.11e

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

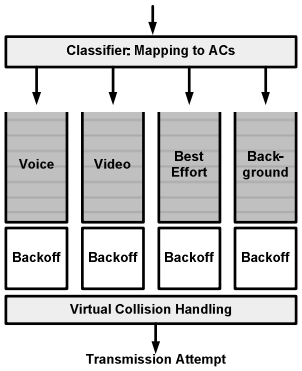

Fig. 1. Mapping to access categories [4]

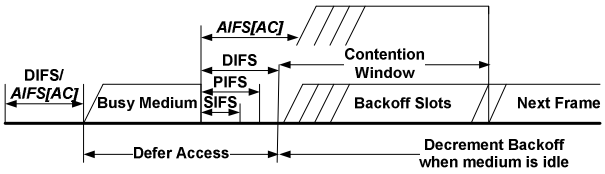

Fig. 2. Channel access prioritization [4]

In 802.11 the data exchange is made by the default simple DATA-ACK. This means that the sender sends a DATA frame and the receiver acknowledges it with an ACK frame. However, this leads to the hidden node problem. To counter this problem, the data exchange can be switched to RTS-CTS-DATA-ACK. The RTS/CTS mechanism uses two small frames (Request/Clear to Send) sent prior to the actual data exchange to inform neighbouring nodes about planned transmissions. This consumes bandwidth, but is necessary to avoid collisions caused by hidden nodes.

The IEEE 802.11 family of standards contain no incentive for nodes to adhere to the specified parameter values. Since new drivers allow manipulating these parameters it is possible that users will want to cheat to maximize their network performance. Based on the described characteristics of 802.11, several types of misbehaviour can be considered. In this paper we concentrate on two of those: cheating on the contention window parameters and the RTS/CTS mechanism. Both these mechanisms result in a decrease in channel access time. The former is done by choosing lower CW values and the latter by refusing to send the RTS/CTS frames.

3 State-of-the-art

One of the first papers dealing with the problem of contention window misbehaviour was [6] (later extended in [7]). The authors take into account several misbehaviour 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. Misbehaviour 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 [9]. In [1], 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 [6] is used to monitor nodes. In the case of misbehaviour, 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 [9]. 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 [11].

Paper [10] 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 behaviour, 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 access points and it complies with standards. However, it cannot be directly used in ad-hoc networks.

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 that many papers discussing contention window cheating in MANETs. In papers [12] and [13] the authors show how modifying the CW values can degrade the performance of an 802.11e ad-hoc network. However, to the authors' knowledge, no papers have considered cheating on the RTS/CTS mechanism. Therefore, the subsequent sections address this issue.

4 Simulation Scenarios

The purpose of the simulation study was to determine how misbehaviour impacts ad-hoc network performance. The actions taken into consideration were manipulating CW parameters and cheating on the RTS/CTS mechanism.

The simulation analysis was performed with the use of the ns2 simulator with a modified version of the TKN EDCA model [14]. This model implements the 802.11e standard in ns2. The modification of the TKN EDCA model involved correcting the RTS/CTS implementation. The following scenario was considered. 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. All stations were within hearing range of each other (i.e., it was a single-hop network). The per-station offered load changed from 64 kb/s to 8 Mb/s.

Table 2. Simulation parameters

Parameter	Value
WLAN Standards	802.11b + 802.11e
Data rate	11 Mb/s
Routing protocol	None
Transport protocol	UDP
Node distribution	Random
Traffic generator	CBR
Packet size	1000 B
Packet exchange	DATA-ACK and RTS-CTS-DATA-ACK

Table 2 presents the various simulation parameters used. 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.

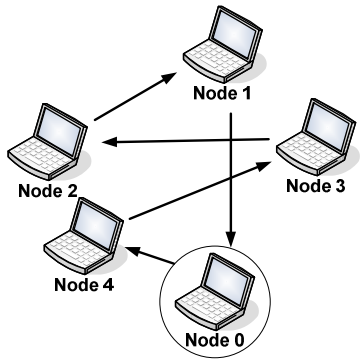

Fig. 3. Network topology

In each scenario, there was one misbehaving node (e.g., the encircled node in Fig. 3). All nodes used the *Best effort* priority to send their traffic. The well behaving (*good*) nodes had unaltered contention window parameters: CW_{min} = 31, CW_{max}=1023. The misbehaving (*bad*) node had these parameters significantly decreased: CW_{min} = 1, CW_{max} = 5. It seems realistic that the misbehaving node would choose such low (or even lower) parameters to maximize its gain. The effect of choosing other CW values and their impact on the use of the RTS/CTS mechanism is studied further on.

5 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).

Fig. 4 presents the simulation results for the small network size (5 nodes). The figure shows the achieved uplink throughput as a function of offered load. The throughput is given for the well-behaving *good* nodes (on average) and for the *bad* node which cheats on the CW. The difference in the throughput of the *good* nodes was insignificant, that is why only the average is shown. In the first case RTS/CTS is off and in the second it is on. In the next case misbehaviour is turned off and RTS/CTS is either on or off. Finally, in the last case, the misbehaving node cheats both on CW and the RTS/CTS mechanism.

The black dashed lines are the reference values and represent the situation in which there is no misbehaviour. Turning on RTS/CTS lowers the saturation throughput. The solid lines represent the situation in which one node misbehaves (cheats on the CW) with RTS/CTS turned off. The misbehaving node dominates the network (this has been shown in [12]). If RTS/CTS is turned on in such a network the throughput, of course, decreases: for the misbehaving node by 30 % and for the good nodes by 40 %.

Another case has been considered – when the misbehaving node decides not to use RTS/CTS despite the fact that the other nodes are using this mode of transmission. The gain is obvious – the misbehaving node's throughput almost reaches the throughput it had when RTS/CTS was not used in the network. This is obviously at the cost of the good nodes' throughput. Therefore, there is a strong incentive for the misbehaving node to turn off RTS/CTS whenever possible.

Similar results regarding obtained throughput occur for medium and large network sizes (Fig. 5 and Fig. 6). The difference is in the throughput achieved by the misbehaving node when the network is saturated because it decreases with network size.

There are two characteristic points in the figures which present throughput. The first occurs once the network reaches congestion. In other words, it is the point where if the network consisted only of well-behaving nodes it would become saturated. Until that point the *bad* node's presence is not harmful. After reaching the congestion point, the *bad* node increases its throughput at the cost of the *good* nodes. This occurs until the second characteristic point is reached. After this, the network is in saturation and the *bad* node has much more throughput than the average *good* node. These two characteristic points can be perhaps most clearly seen in Fig. 4. The first one appears for an offered load a bit higher than 1 Mbit/s, the second one – at approximately 7 Mbit/s. The conclusion is that analysis of misbehaviour should be limited to congestion scenarios. In non-congested networks the misbehaving node does not impact network performance.

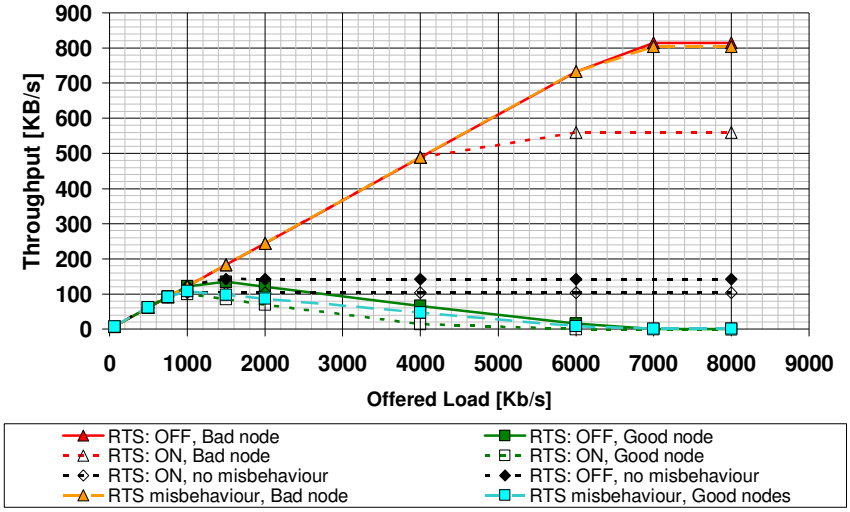

Fig. 4. Throughput vs. offered load (total no. of nodes: 5)

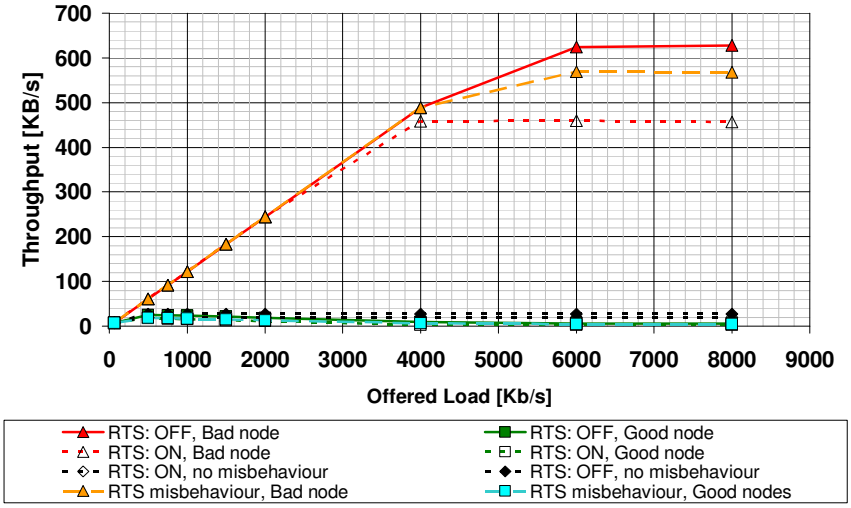

Fig. 5. Throughput vs. offered load (total no. of nodes: 25)

Fig. 6. Throughput vs. offered load (total no. of nodes: 100)

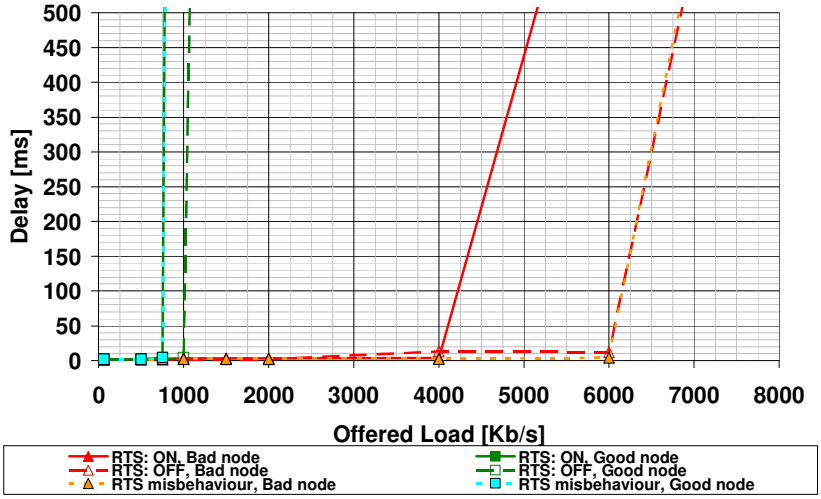

Fig. 7. Packet delay vs. offered load (total no. of nodes: 5)

presents the average frame delay of the misbehaving and well-behaving nodes in the small network scenario. The delay of the *good* nodes suffers greatly in the presence of misbehaviour. It quickly rises very sharply in all cases. The delay of the *bad* node is at an acceptable level for much higher offered loads. With the RTS/CTS mechanism turned on, the delay is low until 4 Mbit/s. If it is turned off (intentionally

or maliciously), it is at a low level until 6 Mbit/s. These observations confirm the conclusions presented above: cheating on the RTS/CTS mechanism "restores" the achieved delay to the value as when the network was not using RTS/CTS. Furthermore, it can be once again noted that in non-congested networks the misbehaving node does not impact network performance (in this case: delay). The measured delay was similar for larger simulated networks, therefore only this figure is being presented.

Two types of cheating have been discussed: manipulating the CW parameters and disabling RTS/CTS in a network which uses this mechanism. The following question arises: is the misbehaviour gain different when these actions are performed alone and together? The answer can be seen in Fig. 8, which shows the throughput gain of the misbehaving node in absolute values. In this case, simulations were performed for a network of 5 nodes (the rest of the simulation parameters remained unchanged) in which RTS/CTS was always enabled. Three cases were considered: the misbehaving node used either CW cheating, RTS/CTS cheating or a combination of both. The achieved throughput was compared with the average node throughput in a network with no misbehaviour. The result is that cheating only on the RTS/CTS mechanism does not give almost any benefits. This is obvious because when there are no hidden stations, the RTS/CTS mechanism only introduces a delay in the medium access. However, if this is combined with CW cheating the gain is much larger than when cheating only on the CW mechanism. There is a synergy between low contention window parameters and refusing to use RTS/CTS. When a node accesses the channel more often (through low CW parameters) the gain from not using RTS/CTS is greater.

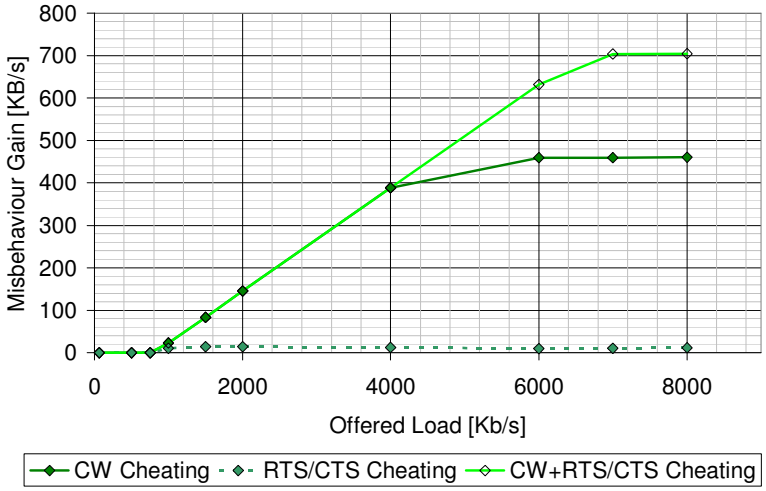

Fig. 8. Misbehaviour gain for different forms of cheating

In the previously mentioned simulations, the CW parameters of the misbehaving node were set to $CW_{min} = 1$ and $CW_{max} = 5$. In order to determine the exact impact of the CW values the following simulation study was performed. The network of 5 nodes (Fig. 3) was in saturation – all nodes were sending UDP traffic of an offered load of 7 Mbit/s. The RTS/CTS mechanism was either off or on. The misbehaving node varied it CW parameters ($CW_{min} = CW_{max}$) from 1 to 100 (Fig. 9). The highest throughput it achieved was for the smallest CW parameters and for RTS/CTS turned off. The *bad* node's throughput decreases in an exponential manner with the increase of the contention window size. The point where the *bad* node's throughput is approximately equal to the average throughput of the *good* nodes occurs for $CW_{min} = CW_{max} = 40$. Since the 802.11 standard does not include any incentives for cooperation, a misbehaving user is free to chose the most profitable CW parameters (i.e., equal to 1).

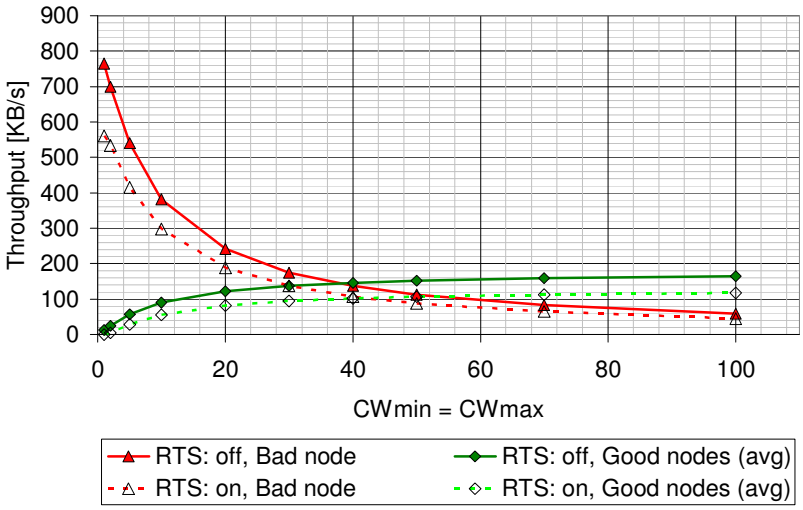

Fig. 9. Throughput comparison for different CW parameters

When dealing with the 802.11e standard it is important to determine the impact of misbehaving in one AC on the performance of a higher priority AC. Simulations were performed, likewise, for a 5 node scenario. The RTS/CTS mechanism was turned on. The well-behaving nodes were using *Voice* priority to send their traffic ($CW_{min} = 7$, $CW_{max} = 15$). The misbehaving node continued to use *Best effort* traffic (with misbehaviour parameters $CW_{min} = 1$ and $CW_{max} = 5$). The results are presented in Fig. 10. In the first case, with no misbehaviour, the achieved throughput rates are in line with the 802.11e standard. When the *bad* node cheated on the CW, it was able to dramatically increase its throughput at the cost of the *good* nodes. Surprisingly, when

the *bad* node cheated on both the CW and RTS/CTS mechanisms, an increase in throughput was observed for all nodes (even the *good* ones). This result can only be explained by the fact that the RTS/CTS mechanism introduces overhead which consumes a small portion of bandwidth. Since one node (the *bad* one) did not use RTS/CTS frames, the total available throughput in the network increased. Therefore, even the *good* nodes could use a small share of this newly available throughput to slightly increase their performance. Had the network consisted of more nodes, the increase in throughput of the well-behaving nodes would be even less significant. If the network was multihop and hidden nodes were present, the gain would depend on how the stations (especially the hidden ones) were placed. In particular it can be assumed, based on [5], that if the misbehaving node was a hidden one in a simple star topology, it would benefit neither from CW manipulation, nor from RTS/CTS cheating.

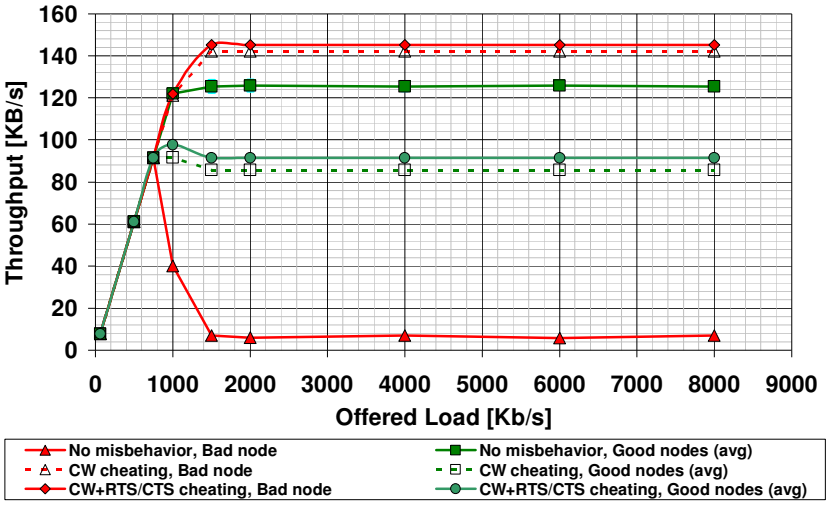

Fig. 10. Throughput vs. offered load for BE vs. Vo priority scenario

6 Conclusions

This paper presented the impact that cheating on the contention window and RTS/CTS mechanism has on single-hop ad-hoc networks. Several simulation scenarios were analyzed. Throughput, delay and fairness were considered for networks of different sizes. A rational misbehaviour model was assumed, i.e., the malicious user would perform simple actions to obtain significant gains.

The first conclusion is that the use of modified CW parameters allows a misbehaving node to jeopardize network performance. The throughput and delay of such a node is significantly better than well-behaving nodes. This occurs regardless of network size and whether the RTS/CTS mechanism is used.

Secondly, a node can cheat on the RTS/CTS mechanism, i.e., refuse to turn in on, even though the whole network is using it. It has been shown that while such behaviour does not provide gains, it is especially beneficial when joined with CW misbehaviour. When used together, these two types of misbehaviour can give greater advantages than when used alone.

Furthermore, a simulation analysis was performed for different CW values of the *bad* node. Assuming that the misbehaving user is rational, and taking into consideration the fact that 802.11 has no mechanisms to encourage proper behaviour, it is obvious that the lowest possible CW values should be chosen.

In non-congested networks, a node's misbehaviour, though theoretically observable, has no influences on its neighbours and is therefore harmless. Therefore, future studies should be focused on congested networks. In real-world ad-hoc networks saturation can be a common situation because of multimedia and peer-to-peer applications.

Finally, it was shown that 802.11e fails to provide QoS in the face of CW and RTS/CTS cheating. A misbehaving node can easily manipulate MAC layer parameters and thus gain an advantage over other nodes. Low priority traffic can be assigned such parameters, with which it can outperform high priority traffic.

Future work will take an even more realistic approach. Studies will focus on multihop ad-hoc networks, which suffer from the hidden node problem. Cheating on other EDCA parameters (AIFS, TXOP) will be taken into account. Furthermore, more complex traffic patterns and networks with more misbehaving nodes will be considered. It is important that misbehaviour is simple, straightforward and advantageous so that it can be performed by any casual user, not just an expert hacker. An analytical model will be derived to support the findings.

Acknowledgments. This work has been carried out under the Polish Ministry of Science and Higher Education grant no. N51739133.

References

1. BenAmmar, N., Baras, J.S.: Incentive compatible medium access control in wireless networks. In: Proceedings 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., Baras, J.S. Detection and Prevention of MAC Layer Misbehavior for Ad Hoc Networks. In: 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. Kosek, K., Natkaniec, M., Vollero L., Pach, A. R.: An Analysis of Star Topology IEEE 802.11e Networks in the Presence of Hidden Nodes, Proc. The International Conference on Information Networking 2008, ICOIN'08, Korea, January 2008.

6. Kyasanur, P., Vaidya, N.H.: Detection and Handling of MAC Layer Misbehavior in Wireless Networks. In: International Conference on Dependable Systems and Networks (DSN'03), p. 173, 2003.
7. Kyasanur, P., Vaidya, N.H.: Selfish MAC Layer Misbehavior in Wireless networks. In: IEEE Transactions on Mobile Computing, Volume 4, Number 5, September/October 2005.
8. MADWiFi – Multiband Atheros Driver for WiFi, http://madwifi.org
9. Radosavac, S., Baras, J.S., Koutsopoulos, I.: A Framework for MAC Protocol Misbehavior Detection in Wireless Networks. In: Proc. 4th ACM workshop on Wireless security (WiSe), Cologne, Germany, September 2005.
10. Raya, M., Hubaux, J., Aad I.: 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.
11. Rong, Y., Lee, S.-K., Choi, H.-A.: Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis. In: Proceedings of INFOCOM 2006, 25th IEEE International Conference on Computer Communications., April 2006.
12. Szott, S., Natkaniec, M., Canonico R., Pach, A.R.: Impact of Contention Window Cheating on Single-hop IEEE 802.11e MANETs. IEEE Wireless Communications and Networking Conference (WCNC 2008), Las Vegas, NV, USA, March 31 – April 4, 2008.
13. Szott, S., Natkaniec, M., Canonico, R., Pach, A. R.: Misbehaviour Analysis of 802.11 Mobile Ad-Hoc Networks – Contention Window Cheating, Med Hoc Net 2007, Corfu, Greece, 12–15.06.2007.
14. Wiethölter, S., Emmelmann, M., Hoene, C., Wolisz, A.: TKN EDCA Model for ns-2. In: Technical Report TKN-06-003, Telecommunication Networks Group, Technische Universität Berlin, June 2006.