Misbehavior Constraint MAC Protocol (MC-MAC) for Wireless Networks

Yupeng Ma, Yonggang Li(✉), Zhizhong Zhang, and Haixing Li

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
lyg@cqupt.edu.cn, {ma-yupeng, li_haixing}@foxmail.com

Abstract. The IEEE 802.11 protocol assumes that all wireless network nodes will abide by the protocol and cooperate well with it. However, in order to obtaining more channel resources or destroying network performance, some selfish nodes will be in misbehaviors when they are in the certain condition wireless contention-sharing channels, such as, Backoff Value Manipulation is a kind of misbehavior. And for this kind of misbehavior, this paper proposes a Misbehavior Constraint MAC protocol (MC-MAC), which can detect and penalize the backoff value manipulation, and it includes a new backoff value generating function with penalty function and a reputation model. Simulation experiments shows that the MC-MAC protocol has a significant inhibitory effect on misbehavior and can improve system throughput.

Keywords: IEEE 802.11 · Medium access control · Misbehavior constraint · Backoff value manipulation

1 Introduction

The IEEE 802.11 Medium Access Control (MAC) protocol, which at the basis of the Distributed Coordination Function (DCF) mechanism, is the most commonly used MAC protocols in current wireless network. When nodes access to wireless channel resources, they must follow the fairness and trust in the certain wireless sharing channels of a distribution network condition. However, there are some nodes will be in misbehavior that do not comply with the wireless network protocol rules. In addition, due to the great programmability of the network adapter (mobile base station), it is much easier for bad nodes to change the parameters of MAC protocols and achieve selfish or malicious purposes.

Nowadays many researches are focusing on MAC layer misbehaviors. The research in [1,2] analyzes the greedy receivers misbehavior. And this misbehavior

This work is supported by the Defence Advance Research Foundation of China under Grants 61401310105 and the Chongqing Research Program of Basic Research Frontier Technology (No. cstc2017jcyjA1246).

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 656–664, 2019.
https://doi.org/10.1007/978-3-030-06161-6_64

is mainly reflected in the traffic that received by selfish nodes is much larger than sending. In research [2], the writer determines the scope of influence of greedy receiving nodes and quantifies the harm of greedy receiver misbehavior by using simulations and tests. The result is that the greedy receiving nodes will cause the nodes which affected by them to receive none traffic.

RTS/CTS (Request to Send/Clear to Send) DOS attack is also a kind of misbehaviors. The principle of RTS/CTS DOS attack is that making competing nodes set a longer Network Allocation Vector (NAV) by tampering with the duration field of the RTS/CTS control frame. The research [3,4] is on simulation analysis of this misbehavior. They found that as long as the NAV duration field is set to the maximum value and the rate of attacking nodes reach 30 frames per second, the normal node cannot access the channel.

Backoff value manipulation [5] as a common misbehavior, to obtain more channel resources, it mainly accesses the channel earlier by selecting a smaller backoff value. This misbehavior will not only reduce system performance, but also can lead to denial of service attacks [6] and result in good nodes cannot communicate properly. The research [7] classifies the backoff value manipulation as continuous misbehavior and intermittent misbehavior. It respectively evaluates and quantifies the harm to the network. After simulation analysis, they found that intermittent misbehavior will easily evade misbehavior detection, but when the size of the network becomes larger, this type of misbehavior will cause little harm to the network. But no matter how the size of the network changes, the continuous misbehavior can cause serious damage to the network.

There are many studies on misbehavior detection [8–10], but only few paper have studied how to suppress misbehavior [11]. Based on the dangerous and continuous of the misbehavior of backoff value manipulation, this paper proposes a MC-MAC protocol at the basis of CSMA/CA protocol. The MC-MAC protocol can detect and penalize the backoff value manipulation, and it specifically includes a new backoff value generating function with a penalty function and a reputation model.

The rest of this paper is organized as follows. We present the details of MC-MAC protocol in Sect. 2. The protocol implementation details and simulation results are discussed in Sect. 3. Finally, Sect. 4 draws the conclusion.

2 Proposed Misbehavior Constraint MAC Protocol(MC-MAC)

In the IEEE 802.11 DCF mechanism, when the channel is busy, a node should randomly select a backoff time in the range of [0, CW] (Contention Window) if wants to send data, and wait until the backoff time goes back to zero before sending the control packet RTS. And it can win the channel if its random backoff time is shorter than the others. The misbehavior of reducing the backoff time is that the selfish node can access the channel earlier with a shorter backoff time than the normal node and preempt resources, then affect the throughput of the normal node and the entire system. In order to limiting the misbehavior nodes in wireless

network, it is necessary to detect the misbehavior of the node, and then punish the selfish node to ensure the fairness of the communication environment. The detection mechanism of MAC-MAC protocol is implemented by modifying the message exchange mechanism of CSMA/CA. Next, calculating the Trust Value (TV) based on the detected performance of the node. Then the penalty level is graded according to the Trust Value (TV) of the node. Finally, the receiver calculates the penalty backoff value based on the $penaltybackoffgeneratingfunction$ proposed in this paper for the sender, and it will be used as the nodes to calculate the penalty backoff value at next time. Next, we will introduce MC-MAC in three parts.

2.1 Detection Mechanism Based on CSMA/CA

MC-MAC protocol detection mechanism is completed by modifying the CSMA/CA message exchange mechanism. The proposed modification can ensure that receiver R can assign a backoff value to sender S through RTS packet and Acknowledgement (ACK) packet. Therefore, R could verify whether the actual backoff time of S deviates from the backoff time allocated by R. This detection mechanism needs to modify the packet headers of the RTS, CTS and ACK packets. And the proposed modifications make the communication between the nodes more transparent. Figure 1 illustrates the message exchange mechanism of MC-MAC protocol and the related packet header changes.

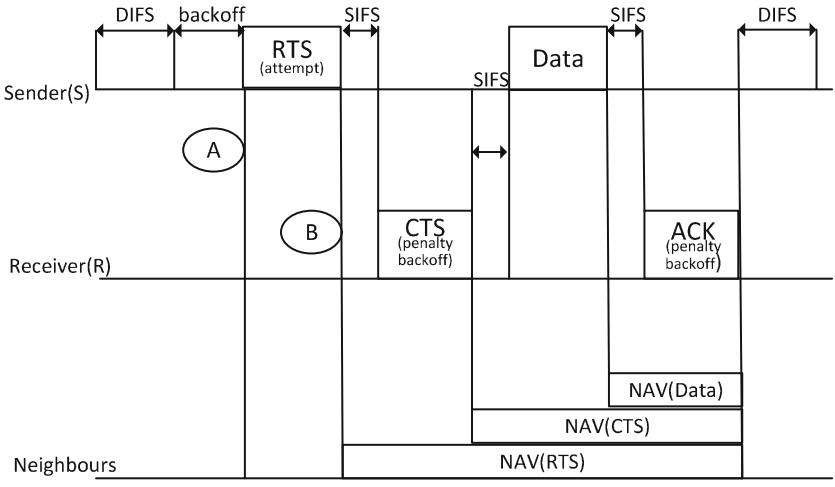

Fig. 1. Detection mechanism based on CSMA/CA

(1) The Sender S generate a backoff value according to the penalty backoff generation function (The following will introduce) during the first communication, but all subsequent transmission S should use the backoff value ($B_{exp} = penaltybackoff$) assigned by the receiver R. In point A of Fig. 1, S

sends an RTS packet to the R, and the number of retransmissions (*attempt*) is added to the packet header of the RTS.
(2) At point B, R receives the RTS packet, then extracts the number of retransmissions (*attempt*), and uses a monitoring function to detect the actual waiting backoff time B_{act}. The actual waiting backoff time B_{act} is equal to the interval which the receiver sends an ACK and receives the next RTS from the same sender.
(3) Receiver R calculates backoff value ($penaltybackoff$) for next transmission according to B_{act}, B_{exp} and $attempt$. Then add $penaltybackoff$ to the packet header of CTS and ACK return to S.

2.2 Trust Value and Statues

The MC-MAC introduces the Trust Value (TV) in order to score the performance of the nodes and then grades nodes according to the score. Changes in the communications environment may affect the protocol's judgment about node performance. However, the grading process is dynamically changing. So the judgment about the performance of a node depends on the multiple communications of the node. Therefore it turns out that grading improves the fault tolerance of the protocol. Equations (1), (2) demonstrate how to calculate the Trust Value (TV) by the receiver (R).

Firstly, the misbehavior factor (Mf) is obtained by Eq. (1). The Mf represents the ratio between receiver reported deviation $B_{exp}\alpha - B_{act}$ to the receivers expected backoff value B_{exp}. The parameter α can be adjusted according to the channel conditions to reduce the error of the judgment. However, when the smaller α is used, the protocol will miss some misbehavior. Therefore, this paper choose a reasonably large α for simulation. Equation (2) shows how to calculate the TV. The initial value of the TV is 100% for each node. Then update the TV according to each node's performance when each communication is completed. Table 1 shows the four grades of penalty level (PL), and the parameter PL is introduced. The PL is divided into four levels according to the TV. The protocol will perform corresponding operations on the nodes according to these four levels.

$$Mf = \frac{B_{exp} \times \alpha - B_{act}}{B_{exp}} \tag{1}$$

$$TV = TV - TV \times Mf \tag{2}$$

Table 1. Trust value and statues

Range of Trust Value	Status
$100 \geq T_v \geq 80$	$PL--, \min(PL) = 1$
$79 \geq T_v \geq 60$	$PL = PL + 1$
$59 \geq T_v \geq 40$	$PL = PL + 2$
$39 \geq T_v \geq 0$	Notifying the upper layer protocol

2.3 Penalty Backoff Generation Function

The penalty backoff generation function proposed in this paper can not only double the contention window value like the IEEE 802.11 BEB (Binary Exponential Backoff Algorithm) after a collision, but also generate a punitive backoff value for selfish node. Such a generating function can prevent the selfish node from selecting a smaller backoff value and not doubling the CW value after a collision. Penalty backoff generation function as shown in Eq. (3).

$$penaltybackoff = f(backoff, senderid, y) * 2^{y-1} * CW_{\min} \tag{3}$$

backoff in Eq. (3) is the backoff value previously assigned to sender by receiver, senderid is the identifier of sender. Equation (4) shows that y is equal to the maximum value of the number of retransmissions attempt and penalty level PL. Sender retransmission may occur when there are nodes competing for the channel. So the attempt of sender may be greater than PL. In order to ease the channel conflict the receiver needs to use the attempt number to calculate a new backoff value (penaltybackoff) for the sender. But for the selfish node, the penalty level PL will be bigger than attempt, therefore the receiver will generate a punitive backoff value for the sender. The initial values of attempt and PL are equal to 1. CW_{min} is the node's minimum contention window $CW_{min} = 31$.

$$y = \max(attempt, PL) \tag{4}$$

Function f uses a classical uniformly distributed random number method-linear congruential [12]. It can generate a uniform random number between 0 and 1, And Function f can be ensure that the sender will choose different backoff value after collisions [11]. Function f as shown in Eq. (5).

$$f(backoff, senderid, y) = ((aX + c) \bmod (CW_{\min} + 1))/CW_{\min} \tag{5}$$

where $a = 5$, $c = 2 * y + 1$ and $X = (backoff + senderid) \bmod (CW_{\min} + 1)$.

3 Simulations Result Analysis

Actually NS2 network simulator is used to simulate MC-MAC protocol to evaluate if the MC-MAC protocol can restrain misbehavior. The simulation was processed at Wi-Fi environment. There are 8 senders and one receiver (AP). Simulation configuration as shown in Table 2. The traffic type is a CBR (constant bit rate) and rate 2 Mbps, wireless channel bandwidth is also 2 Mbps, packet size is 512 bytes.

Misbehavior Model. This paper adopts a dangerous continuous misbehavior model, and analyzes it in [6]. The continuous misbehavior model means that the selfish nodes always have a fixed selfish strategy. This model has a parameter which called misbehavior percentage (MP) to indicate the degree of misbehavior. For example, if the MP of a selfish node is 60%, then this node just needs to

wait for 40% of the backoff value B_{exp} allocated by the receiver. As shown in Eq. (6). The larger the MP is, the smaller the actual backoff value of the selfish node is.

$$B_{act} = B_{exp} \times (1 - MP) \tag{6}$$

Table 2. Simulation Parameters

Parameters	Description
Traffic type	CBR
Packet size	512 bytes
Link bandwidth	2 Mbps
Transmission range	250 m
Number of total nodes	9
Number of misbehavior nodes	1
Routing protocol	DSR
Access method	RTS/CTS-DATA-ACK
Misbehavior percentage (MP)	(1%–100%)
Simulation time	60 s

In this section, we will compare the average throughput of good nodes, misbehavior node throughput, and system throughput for the IEEE 802.11 protocol and the proposed protocol, respectively.

3.1 Performance of MC-MAC Protocol Without Misbehavior

First we test the performance of MC-MAC protocol without misbehavior. The purpose of this test is to evaluate the effect of the occasional misjudgment of the MC-MAC. Therefore, we compare the node average throughput for the MC-MAC protocol with the IEEE 802.11 protocol under different network sizes.

In this simulation, we set the number of nodes from 1 to 60. It should be noted that all nodes are good. Other parameters are unchanged according to the settings in Table 2. Figure 2 shows the average throughput of the nodes for the MC-MAC protocol (red) and the IEEE 802.11 protocol (black) when there is no selfish node. It can be seen from the Fig. 2 that the two curves are in the same trend and almost coincide. This shows that in the absence of selfish nodes, the average throughput of the nodes for the MC-MAC protocol is almost same as the average throughput of the nodes for the IEEE 802.11 protocol. It means that there is little misjudgment of the MC-MAC protocol and the proposed protocol will not reduce the throughput of the network.

3.2 Performance of MC-MAC Protocol with Misbehavior

Figure 3 shows the average throughput of good nodes for the MC-MAC protocol (black) and IEEE 802.11 protocol (blue) under different misbehavior percentage

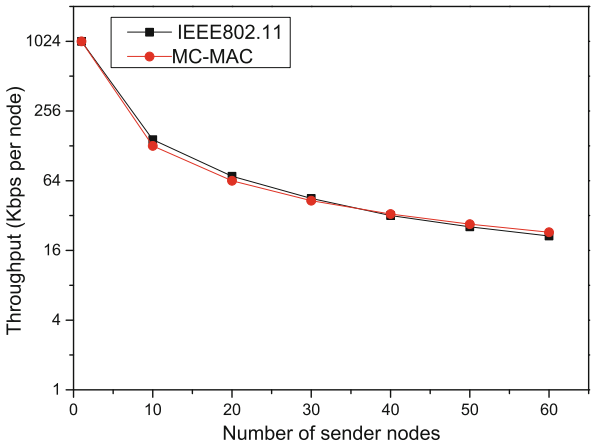

Fig. 2. Throughput of nodes without misbehavior

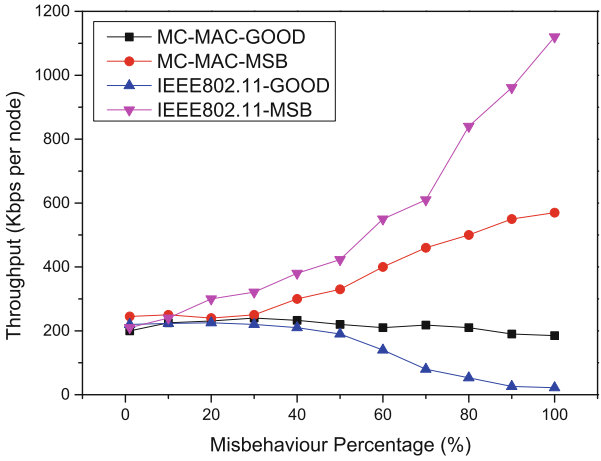

Fig. 3. Throughput of nodes with misbehavior

(MP). The figure also shows the throughput of misbehavior node on the MC-MAC protocol (red) and IEEE 802.11 protocol (pink). According to Fig. 3, when the MP of misbehavior node is from 1% to 100%, the throughput of selfish nodes for both protocols are increase (red and pink), but the MC-MAC protocol (red) is lower. In particular, since the MP reached 60%, the throughput of selfish nodes for the IEEE 802.11 protocol (pink) increased drastically, and the throughput of selfish nodes for the MC-MAC protocol (red) was not that drastic. Also, when the MP of the selfish node increases, the throughput of the good node for the MC-MAC protocol (black) decreases a little, but the average throughput of good nodes for the IEEE802.11 protocol (blue) drops almost to 0. Therefore, it can be concluded that the MC-MAC protocol can keep the average throughput of good

nodes within a normal range in networks when competing misbehavior nodes, and the proposed protocol could reduce the throughput of selfish nodes.

Figure 4 plots the system throughput for the MC-MAC protocol (red) and IEEE 802.11 protocol (black) under different misbehavior percentage MP. From the figure, it can be seen that as the misbehavior percentage MP increases, the system throughput for the IEEE 802.11 protocol (black) decreased greatly, especially after the MP reaches 50%. The MC-MAC protocol (red) has a little change in system throughput as the MP increases. As a result, the MC-MAC protocol is more resilient in wireless networks with misbehavior nodes.

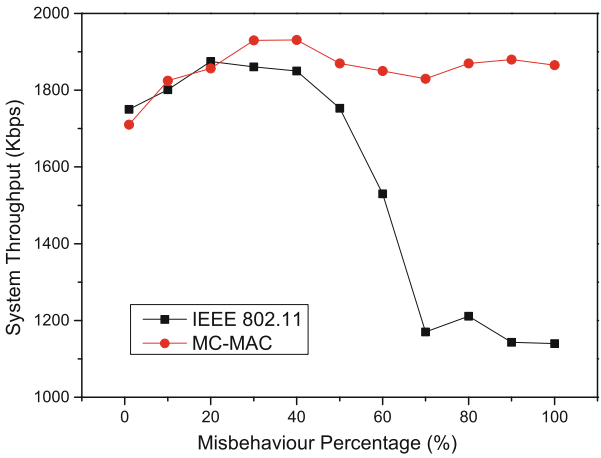

Fig. 4. System throughput

4 Conclusion

This paper proposes the MC-MAC protocol, which is implemented by modifying the IEEE 802.11 MAC protocol. The protocol can suppress the misbehavior that backoff value manipulation, and ensure the fairness and quality of the communication. The Simulation has proved that the protocol can effectively maintain the throughput of good nodes and maintain the throughput of the system in networks when competing misbehavior nodes.

References

1. Diwanji, H., Shah, J.: Effect of MAC layer protocol in building trust and reputation scheme in mobile ad hoc network. In: 2013 Nirma University International Conference on Engineering (NUiCONE), p. 1C3 (2013)
2. Han M.K. Qiu, L.: Greedy receivers in IEEE 802.11 hotspots: impacts and detection. IEEE Trans. Dependable Secur. Comput. **7**(4), 410–423 (2010)

3. Nagarjun, P., Kumar, V., Kumar, C., Ravi, A.: Simulation and analysis of RTS/CTS DoS attack variants in 802.11 networks. In: 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, pp. 258–263 (2013)
4. Alocious, C., Xiao, H., Christianson, B.: Analysis of dos attacks at mac layer in mobile adhoc networks. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International, pp. 811–816 (2015)
5. Kyasanur, P., Vaidya, N.H.: Detection and handling of MAC layer misbehavior in wireless networks. In: Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN 03), pp. 173–182 (2003)
6. Szott, S., Natkaniec, M., Canonico, R., Pach, A.R.: Impact of contention window cheating on single-hop. In: IEEE 802.11e MANETs, Proceedings of the IEEE Wireless Communication and Networking Conference on (WCNC 08), pp. 1356–1361 (2008)
7. Lu, Z., Wang, W., Wang, C.: Modeling and evaluation of backoff misbehaving nodes in CSMA/CA networks. IEEE Trans. Mob. Comput. **11**(8), 1331–1344 (2012)
8. Patras, P., et al.: Policing 802.11 MAC misbehaviours. IEEE Trans. Mob. Comput. **15**(7), 1728–1742 (2013)
9. Zhang, Y., Lazos, L.: Countering selfish misbehavior in multi-channel MAC protocols. In: Proceedings - IEEE INFOCOM12.11, pp. 2787–2795 (2013)
10. Cao, X., et al.: A two-step selfish misbehavior detector for IEEE 802.11-based Ad Hoc networks. In: IEEE Global Communications Conference on IEEE, pp. 1–6 (2015)
11. Kyasanur, P., Vaidya, N.H.: Selfish MAC layer misbehavior in wireless networks. IEEE Trans. Mob. Comput. **4**(5), 502–516 (2005)
12. Knuth, D.E.: The Art of Computer Programming, chapter 3, vol. 2, 3rd edn., pp. 10–17. Addison-Wesley, Boston (2000)