Detection And Prevention Of Greedy Behavior In Ad Hoc Networks

Lin Chen
Université Paris-Sud 11
116 PUBLICATIONS 771 CITATIONS
SEE PROFILE

Jean Leneutre
Télécom ParisTech
40 PUBLICATIONS 302 CITATIONS
SEE PROFILE

Detection And Prevention Of Greedy Behavior In Ad Hoc Networks

Lin Chen, Khaled Aslan Almoubayed, Jean Leneutre
Department of Computer Science and Networking
École Nationale Supérieure des Télécommunications
37-39, Rue Dareau, 75014, Paris, France
{Lin.Chen, aslanalm, Jean.Leneutre}@enst.fr

***Abstract*—IEEE 802.11 DCF (Distributed Coordination Function) is the most popular MAC layer protocol of ad hoc networks. It is designed for networks where all the network participants obey the protocol rules. However, this assumption is not valid in open networks such as ad hoc networks with the presence of greedy/selfish nodes. In fact, by simply manipulating the backoff values, greedy nodes can cause a drastically reduced allocation of bandwidth to well-behaved nodes. Current solutions address the problem by modifying IEEE 802.11 DCF and providing verifiable backoff values at receiver side. However, they are not resistant to sender-receiver collusion. In this paper, we address the problem by proposing the selfishness resistant MAC protocol for ad hoc networks. Compared with existing solutions, our approach can counter sender-receiver collusion. Moreover, it is lightweight and simple to integrate to existing IEEE 802.11 DCF with almost no additional traffic overload.**

Keywords: Ad hoc networks, IEEE 802.11 DCF, Greedy/selfish behavior, Network security

I. Introduction

The IEEE 802.11 DCF (Distributed Coordination Function), based on the Carrier Sense Medium Access with Collision Avoidance (CSMA/CA) mechanism, is the most popular MAC layer protocol of ad hoc networks. It is designed for networks where all the network participants obey the protocol rules. However, the assumptions is not valid in open networks such as WLANs in airports and ad hoc networks in hostile environments.

Moreover, nowadays, the network adapters are becoming more and more programmable. As a result, a user can modify the behavior of his wireless interface very easily when the implementation of MAC protocol in software rather than hardware or firmware in network access cards, e.g., for some network cards, to cheat on the *contention window* (CW) values CW_{min} and CW_{max} values, all one has to do is simply to write to the correspondent register stocking these values the desired values using the following command:

```
writel(value, register_address);
```

Even network interface cards implementing most MAC layer functions in hardware and firmware usually provide an expanded set of functionalities which can be exploited to circumvent the limitations imposed by the firmware. In the worst case scenario a vendor might create network interface cards violating the MAC protocol to create an improved performance of its products.

As result, IEEE 802.11 DCF is very vulnerable to various attacks and even slight changes of one protocol parameter in one or a set of nodes can have devastating effects on the overall network performance which may further lead to denial of service (DoS).

II. IEEE 802.11 MAC Layer Misbehavior

MAC layer misbehavior in ad-hoc networks can be classified into two categories: greedy/selfish misbehavior and malicious misbehavior. Selfish nodes typically misbehave to improve their own performance, e.g., operating on smaller CW value. Malicious nodes aim primarily at disrupting the normal operation of the network, e.g., jamming the communication channel.

In our work, we focus on the IEEE 802.11 MAC layer greedy behavior in ad hoc networks. A greedy node is a node that deliberately misuses the IEEE 802.11 DCF to gain bandwidth at the expense of other nodes. The benefits of this misuse are the following:

- The greedy nodes achieve significant bandwidth gains as they directly deal with the wireless medium.
- The greedy misbehavior is hidden and independent from upper layers and hence cannot be detected by any mechanism designed for those layers.

Among the various greedy behavior, the most profitable one is the manipulation of backoff value, i.e., operating on smaller CW value. It is also the most challenging task to detect the backoff manipulation. Due to the randomness introduced in the choice of the back-off, it is difficult to decide if a node has chosen small back-off values by chance or if the small back-off values are part of a misbehavior strategy.

III. Related Work

The current literature offers two major categories of approaches to address this problem: the approaches based on the modification of the IEEE 802.11 DCF and the approaches without making any changes to the IEEE 802.11 DCF. In this paper, we focus on the first category of approach.

One example of the first category of approaches is proposed in [1]: instead of the sender selecting random backoff values to initialize the backoff counter, the receiver assigns a random backoff value B_{exp} and sends it in the CTS and ACK frames to the sender. The sender uses this assigned backoff value in the next transmission to the receiver. So a receiver can identify the sender's deviating from the protocol by observing the number of idle slots between consecutive transmissions from the sender. If this observed number of idle slots is less than a specified fraction α of the assigned backoff, then the sender is considered to deviate from the protocol. For a detected node, a penalty for the next assigned backoff is selected based on the deviation.

[2] proposes another protocol called ERA-802.11. The basic idea of this protocol follows the protocol for flipping coins over the telephone by Blum [3]. The goal is that the sender and the receiver agree through a public discussion on a random value. The main property of the protocol is that an honest party will always be sure that the agreed value is truly random.

The above existent schemes is not resistant to colluding nodes. For example, in the solution proposed in [2], when sender and receiver conspire by selecting their backoff a priori, they can deny the network access to neighboring nodes. Thus these solutions may work well in wireless networks with infrastructure, but they are not adapted for infrastructure-less environments such as ad hoc networks where no a priori trust is available.

Motivated by the above analysis, we propose the selfishness resistant MAC protocol for ad hoc networks with the following desirable properties:

- It can counter sender-receiver collusion;
- It does not require sender-receiver negotiation or dialogue;
- It is lightweight and simple to integrate to existing IEEE 802.11 DCF;
- There is almost no additional traffic overload.

IV. Proposed Solution

In IEEE 802.11 protocol, a sender transmits a RTS after waiting for a randomly selected number of slots in the range $[0, CW]$. Consequently, the time interval between consecutive transmissions by the sender can be any value within the above range. Hence, a receiver that observes the time interval between consecutive transmissions from the sender cannot distinguish a well-behaved sender that legitimately selected a small random backoff, from a misbehaving sender that maliciously selected a non random small backoff. It may be possible to detect sender misbehavior by observing the behavior of senders over a large sequence of transmissions, but this may introduce a large delay in detecting misbehavior. In addition, it may not be feasible to monitor the behavior of senders over a large sequence of transmissions when the node mobility is high. Hence, we propose modifications to the IEEE 802.11 standard that enables a monitoring node to identify sender misbehavior within a small observation interval.

Our proposed solution is designed to handle greedy MAC layer misbehavior in nodes using IEEE 802.11 DCF mode. The goal of this solution is to simplify greedy misbehavior detection. It will mainly address the backoff manipulation attacks. This solution is based upon a slight modification of the IEEE 802.11 MAC protocol with RTS/CTS dialogue. We assume that the MAC address of the network participants are non spoofable.

A. Approach Overview

In each MAC data frame we add a new field SN (sequence number, 45bits), which is initialized by a random value, then it is incremented for each successive frame transmission; and f (contention window multiplication factor $f \in 0, 1, 2, 3, 4, 5$, 3bits) field, which is initialized to 0.

We suggest the following scheme to calculate the backoff value for each node:

- Initialization: $f = 0$, $SN = rand()$
- $Backoff = Hash(SN||MAC)\ mod\ 2^f \times CW_0$
- If no collision occurs, $SN = SN + 1$, If collision occurs, $f = f + 1$ and $SN = SN + 1$

We used a random value to initialize the SN field in order to give the sender different backoff values sequence on each startup. We increment the SN after each transmission even if a collision is produced in order to avoid consecutive collisions caused by two (or eventually more than two) senders which chose the same backoff value, and by adding the MAC address information we are certain that the next calculated backoff values are different.

We chose a hash function because it is a one-way function, so it would be impossible for a sender to choose a small backoff values then calculate the corresponding SN. This guarantees to the receiver that the sender did not fabricate its own backoff value. And in order to be verifiable at the receiver side (or at a monitoring node).

Using this modification the receiving node (or any other monitoring node) can easily verify that the sending node has used a valid backoff value, and thereby respected the protocol. This solution assumes that the MAC addresses are non spoofable, this problem can be addressed at upper layers. The advantages of this solution are:

- Prompt detection of backoff manipulation
- Prompt detection of DIFS/SIFS manipulation
- Prompt detection of CW manipulation
- Colluding nodes cannot take advantage of this mechanism
- No trusted receiver and/or sender is assumed

B. Detection Scheme

The goals of a misbehaving node range from exploitation of available network resources for its own benefit up to network disruption. The solution to the problem is the timely and reliable detection of such misbehavior instances, which would eventually lead to network defense and response mechanisms and isolation of the misbehaving node. However, two difficulties arise: the random nature of the IEEE 802.11 protocol, and the nature of the wireless medium with its inherent volatility. Therefore, it is not easy to distinguish between node misbehavior and occasional protocol malfunction due to wireless link impairment.

The detection scheme that we propose is dependent and based upon the backoff scheme presented earlier and it has three main verification tests:

1) Backoff verification
2) Sequence number verification
3) Exponential backoff verification

The scheme goes as follows: Each monitoring node have a verification table (VT), each line contains: the MAC address of the sending node, the SN, and the time of the last frame that was received from this node. This table is updated periodically so the entries that are older than a given threshold τ are deleted. The following table shows an example of this table.

MAC address	SN	Time
`00-17-F2-4D-EC-86`	1349	105.35
`00-14-51-77-FA-13`	198	132.15
`00-80-5A-39-71-FE`	46	127.9
⋮	⋮	⋮

TABLE I
VERIFICATION TABLE

In addition to the VT the monitoring station measures the channel idle time (CIT). Upon receiving a frame, the monitoring node calculates the backoff value of this frame using the information provided in the header (MAC address, SN and f) and compares it to the CIT. If the CIT was smaller than the calculated backoff (plus the DIFS); the monitoring node raise an alert. If the CIT was greater than the backoff; the monitoring node proceeds by checking the VT; If there is no entry for this nodes MAC address in the VT; the monitoring station will add a new line to the table with the nodes MAC address, the SN of the frame, and the time of the reception. If there is an older entry for this node; the monitoring station verify that the new SN is greater than the one in its table; if not it will raise an alert.

If the new SN is greater than the old one; the monitoring station calculates the time differences, Δt, and thus deduces the maximum number of frames n that could have been sent within this Δt, then it verifies than the new SN is no more than the old one by n.

It maybe possible for the sender to provide incorrect f value; to overcome this problem, the receiver can sense the channel to identify high collision intervals. During these intervals, the receiver can analyze the traffic to identify any sender S achieving larger number of successful transmissions than other hosts, or having smaller average f value than other hosts. If such a sender S exists, the receiver can intentionally drop frames (RTS frames, to not significantly affect the throughput of S) from S, and verify that S increments f, when it retransmits the frame. Even a single failure by S to increment f is an immediate proof of misbehavior and an alert is raised. Figure 1 shows the flowchart of the detection algorithm.

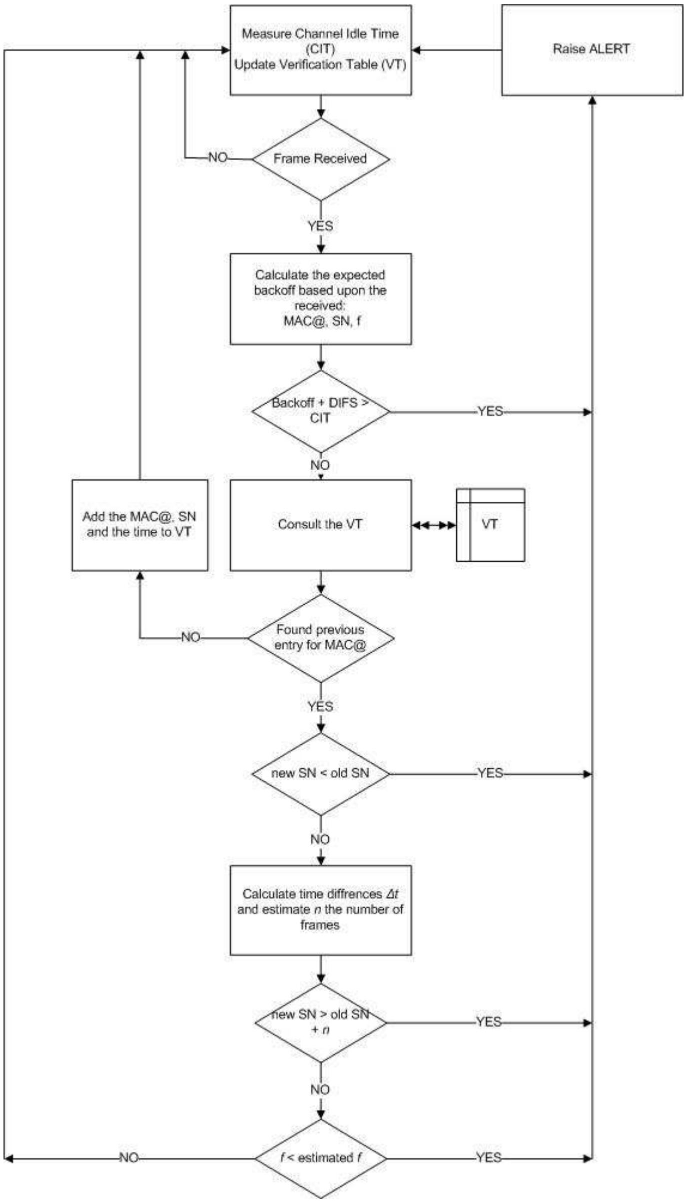
Measure Channel Idle Time (CIT)
Update Verification Table (VT)
Raise ALERT
NO
Frame Received
YES
Calculate the expected backoff based upon the received: MAC@, SN, f
Backoff + DIFS > CIT
YES
NO
Add the MAC@, SN and the time to VT
Consult the VT
VT
NO
Found previous entry for MAC@
YES
new SN < old SN
YES
NO
Calculate time diffrences Δt and estimate n the number of frames
new SN > old SN + n
YES
NO
NO
f < estimated f
YES

Fig. 1. Detection Algorithm

V. PERFORMANCE EVALUATION

A. Fairness Tests

One important property that should be guaranteed is the randomness of the backoff value. In this section we present the results obtained by carrying randomness tests on our backoff calculation scheme to test whether the backoff values calculated by Hash operation is uniformly distributed in the range $[0, CW]$. We compute the entropy [4], [5] of the backoff values in our approach. Table II shows the obtained results for 1000 samples. The entropy should be as close as possible to $log_2\,(CW)$ for a random sequence. We can see from the result that the backoff values in our approach shows satisfactory randomness with both MD5 and SHA-1 Hash functions.

CW	MD5	SHA1
32	4.97	4.98
64	5.93	5.96
128	6.90	6.92
256	7.79	7.82
512	8.55	8.59
1024	9.12	9.13

TABLE II
CALCULATED ENTROPY OF BACKOFF VALUES IN OUR APPROACH

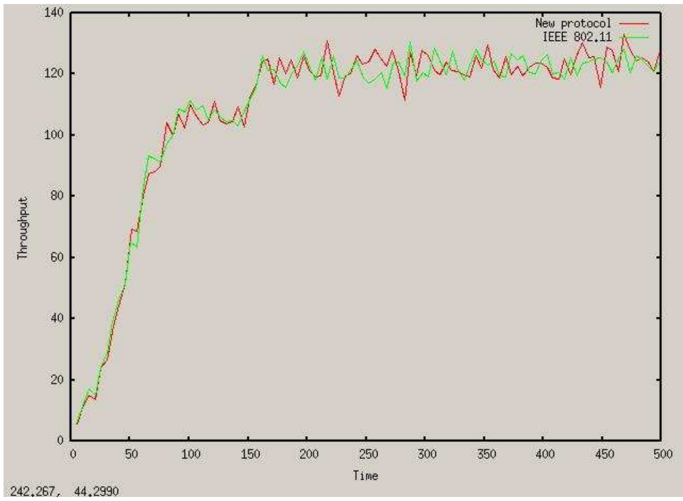

Fig. 2. Throughput

B. Performance Of Our Approach In Non-hostile Environments

In this section we will present the performance tests results we obtained running simulations in a non-hostile environment, with respect to the IEEE 802.11 performance.

1) Simulation Environment: The simulation is conducted on the Network Simulator (ns-2) [6], [7]. Table III lists the ns-2 parameters in our simulation.

Parameter	**Value**
Topology	500m × 500m
Node movement	random waypoint model
Max movement speed	5m/s
Radio range	250m
Sending capacity	2Mbps
Nodes count	30
Total number of flows	15
Packet size	1024 byte
Data rate	2 packets/s
Send buffer	64 packet
Traffic model	CBR
Routing protocol	OLSR
Execution time	500s

TABLE III
NS-2 SIMULATION ENVIRONMENT

2) Simulation Results: We will focus on the throughput, delay, and jitter which are the main characteristics of network performance, as shown in Figures 2, 3, 4.

The simulation results shows that our proposed solution has the practically same performance as the IEEE 802.11 standard in non-hostile environments. We are currently evaluating the performance of our approach in hostile environments with the presence of greedy nodes.

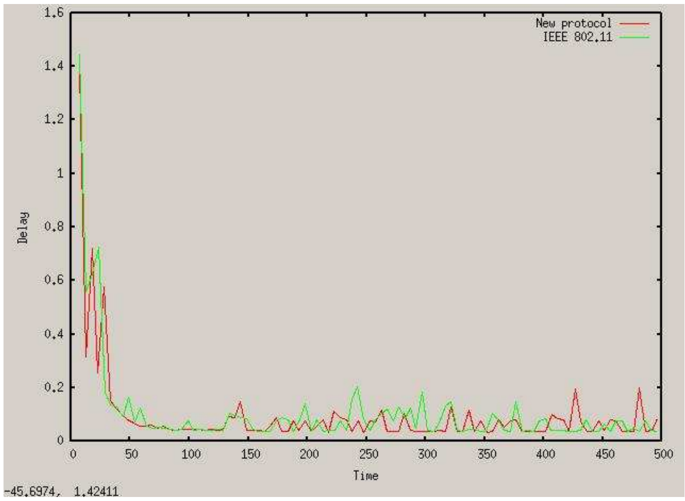

Fig. 3. Delay

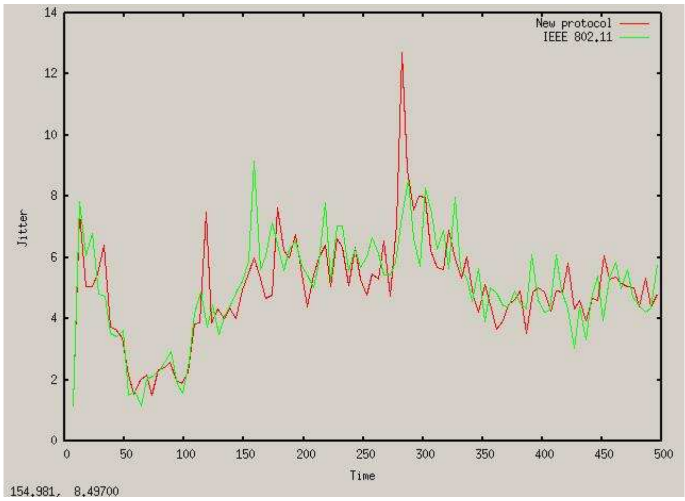

Fig. 4. Jitter

VI. DISCUSSION

The proposed solution as mentioned above, handles the backoff manipulation attacks, this eventually leads to the detection of smaller DIFS attacks. The oversize NAV attack is detected easily by a monitoring node, while the frames scrambling attack is more or less related to DoS attacks. We wish to state here that we do not address the problem of MAC address spoofing. Prior work on the design of certificate authorities for effectively using public key infrastructure in ad hoc networks [8] is assumed to sufficiently thwart attempts to spoof MAC addresses.

Our scheme introduces a new potential attack, where an attacker might try to have an off-line list of SN values and the corresponding backkoff values. So he might choose to fix a backoff threshold that he does not want to exceed and he tries to skip the SN values which generate these big backoff values. In the detection scheme we added a new verification test related to this problem, which is the sequence number verification test, this test ensures that the SN value is not incremented in a exaggerated manner, thus reducing the probability that such attack would succeed on the long term.

VII. Conclusion And Future Work

Misbehavior at the MAC layer by changing the backoff mechanism can lead to performance degradation and even denial of service attacks in ad hoc networks. Handling MAC layer misbehavior is an important requirement in ensuring a reasonable throughput share for well-behaved nodes in the presence of misbehaving nodes.

In this paper, we presented modifications to IEEE 802.11 DCF that simplifies misbehavior detection by proposing a novel scheme for calculating the backoff values. Compared with existing solutions, our approach can counter sender-receiver collusion. Moreover, it is lightweight and simple to integrate to existing IEEE 802.11 DCF with almost no additional traffic overload.

As our future work, we plan to study the performance of the proposed scheme in hostile environments. Then we plan to address the issue of how to react after detecting a misbehaving node. How bad is the performance degradation for the rest of the network? What is the best punishment strategy?

References

[1] P. Kyasanur and N. H. Vaidya, “Selfish MAC layer misbehavior in wireless networks,” *IEEE Trans. Mob. Comput*, vol. 4, no. 5, pp. 502–516, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TMC.2005.71

[2] A. A. Cárdenas, S. Radosavac, and J. S. Baras, “Detection and prevention of MAC layer misbehavior in ad hoc networks,” in *Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks, SASN, Washington, DC, USA*. ACM, 2004, pp. 17–22. [Online]. Available: http://doi.acm.org/10.1145/1029102.1029107

[3] M. Blum, “Coin flipping by telephone - A protocol for solving impossible problems,” in *COMPCON*. IEEE Computer Society, 1982, pp. 133–137.

[4] R. W. Hamming, *Coding and Information Theory*. Prentice-Hall, 1986.

[5] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEETIT: IEEE Transactions on Information Theory*, vol. 23, 1977.

[6] *Network Simulator ns-2, http://www.isi.edu/nsnam/ns/*, NS-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[7] F. Schmidt-Eisenlohr, J. Letamendia-Murua, M. Torrent-Moreno, and H. Hartenstein, “Bug fixes on the IEEE 802.11 DCF module of the network simulator ns-2.28,” Institute of Telematics, University of Karlsruhe, Germany, Tech. Rep., Jan. 2006.

[8] S. Yi and R. Kravets, “Moca: Mobile certificate authority for wireless ad hoc networks.” [Online]. Available: citeseer.ist.psu.edu/676460.html