A Fuzzy Logic based Scheme to Detect Adaptive Cheaters in Wireless LAN

Soufiene Djahel and Farid Naït-Abdesselam
LIFL – UMR CNRS USTL 8022 – IRCICA
University of Lille, France
{soufiene.djahel, farid.nait-abdesselam}@lifl.fr

Abstract—The most commonly used medium access mechanism in WLAN is based on the CSMA/CA protocol. This protocol schedules properly the access to the medium among all competing nodes. However, in a hostile environment, such as wireless local area networks (WLANs), selfish or greedy behaving nodes may prefer to decline the proper use of the protocol’s rules in order to increase their bandwidth shares at the expense of well behaving nodes. In this paper, we focus on one such misbehavior and in particular on the adaptive greedy misbehavior of a node in the context of wireless local area network environment. In such environment, wireless nodes compete to gain access to the medium and communicate directly with an access point (AP). In this case, a greedy node may violate the common rules in order to earn extra bandwidth upon its neighbors. In order to avoid its detection, this node may adopt intelligent different techniques and switch dynamically between each of them. To counter such a misbehavior, we propose the use of a fuzzy logic technique in a new detection scheme. This scheme, implemented in the access point, monitors the behavior of associated wireless nodes and reports any deviation from the proper use of the CSMA/CA protocol. The simulation results of the proposed scheme show its robustness and ability to detect and identify quickly most of the deviations of an adaptive cheater.

Keywords – Wireless Local Area Networks, MAC Layer misbehavior, Adaptive Cheater, Fuzzy Logic.

I. INTRODUCTION

The increase in computation power, the compactness of size, incorporation of mobility and ease of connectivity from anywhere are amongst the major factors that resulted in tremendous growth of handheld devices in recent years. From cordless phones to cellular networks and from WiFi to sensors, the wireless medium has become the preferred backbone of today’s deployed networks. The widespread deployment of wireless hotspots at many public places has increased the wireless users’ motivations to misbehave in order to get a longer communication with the hotspot. Therefore, this misbehavior, which can be easily launched by any wireless node, leads to unfairness in bandwidth sharing among the wireless nodes attached to the same hotspot and consequently the network performance collapses sharply.

Since IEEE 802.11 MAC protocol, as described in [4], [9], is commonly used by wireless nodes to access the medium, any misbehavior at this level may jeopardize the network performance. The serious damage caused by MAC layer misbehavior has received a considerable research attention leading to an in-depth investigation and analysis of the root causes of this misbehavior [5], [6] and [11]. As a result of this investigation, some pioneering contributions have been proposed in the literature to cope with this problem such as [7], [8] and [10]. These works identify several types of MAC layer misbehavior, and propose countermeasures to detect and prevent such misuses. However, their solutions are based on the assumption that the misbehaving node has no knowledge about the way the detection scheme works. Therefore these solutions are unable to struggle a smart cheater which might be aware of the functioning detection schemes. Such cheater exploits its knowledge to escape from the detection system.

In this paper, we study the adaptive cheater misbehavior and explain how easy this can be performed in IEEE 802.11 MAC protocol. We then present our solution, dubbed FLSAC, which exploits the strength of fuzzy logic theory to detect and identify a cheater node. To the best of our knowledge, such adaptive greedy misbehavior has not been investigated until so far, and FLSAC represents the first work which deals with and provides countermeasures.

The rest of the paper is organized as follows. In section II, we give an overview of the MAC layer vulnerabilities. Next, we address the components and the detailed description of the proposed FLSAC scheme in section III. In section IV, we report our simulations and discuss the obtained results. Finally, section V concludes the paper and gives some future investigations.

II. MAC LAYER VULNERABILITIES

A misbehaving node may disobey the MAC protocol rules to gain more bandwidth over regularly behaving honest nodes. To do so, it should change the MAC layer parameters. A node is able to change these parameters only in networks access cards that run the MAC protocol in software instead of hardware or firmware. In this case, the misbehaving node can easily implement the following misbehavior strategies:

- Selects its backoff values from different distributions, for example the backoff period is randomly picked out from the interval \([0, k \times CW_{\text{min}}]\) where \(0 \leq k \leq 1\). Note that if \(k = 1\) then the cheater behaves correctly however it doesn’t not double its CW after a collision is occurred. Moreover, it can adopt different retransmission strategies upon experiencing unsuccessful transmission.
- Scrambles CTS or ACK frames of other nodes in order to increase their contention windows.
- When the channel is sensed to be idle, it transmits before the required DIFS time slots elapse, i.e. the misbehaving node waits for a shorter period called S-DIFS (Short-DIFS). This misbehavior technique is significant only if the cheater node’s backoff was already elapsed before it defers its transmission. Otherwise the result is similar to the case of backoff manipulation.
- Increases the value of the duration field in RTS or DATA packets, such that the receiving nodes updates their NAV according to the received duration. As a consequence, if the misbehaving node has more packets to send, it gets more chance to access the medium as it starts decreasing its backoff before its neighbors. Otherwise a DoS attack is occurred.

A more sophisticated cheater, which has some knowledge about the deployed detection system, can easily switch frequently between these several strategies to avoid detection. Moreover, it is worth noting that it is not compulsory for the cheater to know the actual parameters of the detection system. A rational switch rate between the different misbehavior strategies, without large deviation from the...
standard (for each technique), allows it to acquire more bandwidth than the well behaving nodes, without being detected by the actual systems.

III. THE PROPOSED FLSAC SCHEM

A. Main Idea

Our solution attempts to extend the DOMINO scheme to have a first line defense against the adaptive cheaters. The main idea behind this solution is to carry out a global estimation of the observed deviation from the legitimate protocol operation of a given wireless node. Here, the global estimation means that instead of testing each misbehavior technique alone, we carry out a global test which encloses all the techniques together. To do so, we apply a fuzzy technique [3] which is proven to be suitable in such cases. The advantage of using fuzzy system is to eliminate the decision making ambiguity regarding the behavior of an adaptive cheater which never reaches a threshold of one misbehavior technique to gain more bandwidth than the other nodes.

B. Fuzzy Controller Description

Now we provide a detailed description of our scheme, introducing some notions of fuzzy logic such as fuzzy sets and fuzzy inference in order to help readers unfamiliar with this topic for better understanding. For an exhaustive presentation of fuzzy logic theory, the reader can refer to the abundant literature in this topic such as [2].

In the following, we describe how FLSAC works to detect the adaptive cheater. The role of each component shown in Figure 1 is defined as follows:

1) Inputs: We have designed our fuzzy system to support four inputs where three of them represent the traces collected by the AP for each monitoring period T the AP computes the average of the observed backoff values of a station S_i. Then, it compares it to its own average backoff B_AP to distinguish whether S_i is a cheater or not. However a smart cheater can easily trick DOMINO by choosing N - m times a small cw value and for m (m ≥ 1) times large cw value such that:

\[ \alpha \times B_{AP} \leq \frac{\sum_{i=1}^{N-m} B_i + \sum_{j=(N-m)+1}^{N} B_j}{N} \leq \beta \times B_{AP} \] (1)

Even if the detection system parameters, such as T and B_AP, are not easy to guess, a cheater can escape from the detection system by using a sequence of backoff values 0 and cw alternatively. In any ways, the cheater is still accessing the medium more than the other nodes while keeping its average backoff below the B_AP × α. (α being a parameter configured according to the desired false detection ratio).

For this reason, we estimate in our scheme the deviation (BDEV) of a node from the standard backoff algorithm as follows:

\[ BDEV = \frac{\sum_{i=1}^{N} (B_{AP} - B_i)}{N} \] (2)

where B_i is the i^{th} observed backoff value of a node. Notice that if (B_{AP} - B_i) < 0 then this difference is considered as 0. Therefore, in our scheme we consider the deviation for each backoff value rather than the average of all values.

Example:

Let us suppose that during a monitoring period the AP observes the following backoff values of a node S_i: 4, 5, 0, 18, 0, 30, 6 while the mean backoff of the AP is 8. In this case, the DOMINO scheme concludes that this node is well behaving since

\[ \frac{4 + 5 + 18 + 30 + 6}{7} = 9 > 8 \]

whereas our scheme calculates the following deviation according to the Equation 2:

\[ BDEV = \frac{4 + 3 + 8 + 0 + 8 + 0 + 2}{8} = 44\% \]

This BDEV value will be processed later by the fuzzy controller together with the other inputs to assess the global deviation of the node.

b) RTR (Retransmission rate): This parameter is used to detect the node which scrambles other’s frames to increase their contention windows. It is calculated as explained below:

\[ RTR = \frac{num - rtx(S_i)}{AVG_{tx}(num - rtx(S_j))} \] (3)

where num - rtx(S_i) is the retransmission number of the node i and AVG_{tx}(num - rtx(S_j)) is the average number of the other nodes’ retransmissions.

c) S-DIFS (Frames sent after short DIFS): This parameter is used to count the number of times a node accesses the channel without waiting for the required DIFS period, either after its own successful transmission or whenever its NAV period is elapsed. The distinction of this misbehavior from BDEV one is a hard task since it is not easy to determine the exact time spent for DIFS and the one consumed for backoff. Therefore, the accurate measurement of this time is feasible only in the following cases:

- After a successful transmission, the channel is found idle, i.e. the random backoff selection step is skipped.
- The backoff value selected by the cheater is 0.
- The backoff time of the cheater is already expired before setting its NAV.

2) Fuzzification: This step consists of replacing the input values by corresponding fuzzy parameters. To evaluate the deviation of each input, three fuzzy sets are defined: Low (L), Medium (M) and High (H). Formally, a fuzzy set F in a universe U can be defined by the following membership function:

\[ \beta_F : U \rightarrow [0,1] \] (4)

such that for each u ∈ U, its degree of membership to F is given by \( \beta_F(u) \). In our fuzzy controller, we use a trapezoidal method as a membership function due to its simplicity [3].

![Figure 1: The main components of FLSAC](image)
3) Rules-based Decision: This step aims to use the rules established by the expert together with the knowledge acquired from the knowledge base to classify the node’s behavior in one of the following classes: Normal (N), Lowly Suspected (LS), Highly Suspected (HS) and Cheater (C). The knowledge base defines the relationship between the crisp inputs/outputs and their fuzzy representation understood by the system.

The degree of truth for a predicate in the form \( P \) is given by \( \beta_P = \beta(x) \). The traditional logic operators such as \( \land \) (AND) and \( \lor \) (OR) are redefined in order to produce the truth value of the final statement as follows:

\[
\begin{align*}
\beta_P \land \beta_P' &\equiv \min(\beta_P, \beta_P') \\
\beta_P \lor \beta_P' &\equiv \max(\beta_P, \beta_P') \tag{5}
\end{align*}
\]

The rules are formulated as IF-THEN directives where the condition part is built using the membership of each input to every fuzzy set, and the conclusion is the corresponding classification of the node behavior. For example, if we consider the following rule:

**IF** BDEV is H and RTR is M and S-DIFS is H **THEN** the node is a Cheater

for which we have the following membership functions:

- BDEV (0.3, 0.5, **0.2**), RTR (0.4, **0.3**, 0.3) and S-DIFS (0, 0.2, 0.8).

By applying the above rule the result will be "the node is 20% Cheater" because the minimum of (0.2, 0.3, 0.8) is 0.2. The rules that fill our rule base are depicted in the tables I and II while the rules shown in table III are constructed based on the last monitoring period decision combined with the output of the table II.

These rules are inferred following an in depth analysis of the correct behavior of nodes in DCF mode. Notice that we have given significant weight to BDEV and RTR misbehavior since they have more impact of gaining access to the medium as compared to S-DIFS misbehavior. Besides, they are more harmful in terms of the incurred performance loss, i.e., the RTR misbehavior allows the cheater to access the medium easily by decreasing the spatial reuse because even if the cheater is not in the saturated case, the other nodes have to count down a larger backoff value before acquiring the channel.

4) Defuzzification: In this phase we use the fuzzified rules to calculate the final decision which provides an appropriate crisp value to be used as an output. A number of defuzzification strategies exist. One of the most commonly used techniques is CoG (Center of Gravity), in which a crisp output \( \text{Output}_{\text{crisp}} \) is chosen using the center of area of each fuzzy set. The CoG technique is given by the following formula:

\[
\text{Output}_{\text{crisp}} = \frac{\sum_{i=1}^{4} d_i \beta_i}{\sum_{i=1}^{4} \beta_i} = \frac{d_N \beta_N + d_{LS} \beta_{LS} + d_{HS} \beta_{HS} + d_C \beta_C}{\beta_N + \beta_{LS} + \beta_{HS} + \beta_C} \tag{6}
\]

where \( d_i \) is the the membership function’s center of area corresponding to each class \( i \) of node behavior, and \( \beta_i \) is the membership level of a node behavior to the class \( i \).

The operations of FLSAC are summarized by the Algorithm 1.

IV. SIMULATIONS

This section reports the simulation results by using the OPNET 14.0 network simulator [1]. The simulation scenarios and settings are summarized in table IV.

1In fuzzy logic, the term crisp is used to indicate variables having exact values, as opposed to the term fuzzy, which indicates a qualitative rather than quantitative method of representation.

2The decision about the node behavior at the end of the previous monitoring period.

---

Table I: Fuzzy rules of the formula

\[(RES_1 = (BDEV \land RTR))\]

<table>
<thead>
<tr>
<th>BDEV</th>
<th>RES_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>M</td>
<td>C</td>
</tr>
<tr>
<td>H</td>
<td>N</td>
</tr>
</tbody>
</table>

Table II: Fuzzy rules of the formula

\[(RES_2 = (BDEV \land RTR) \land S \land DIFS)\]

<table>
<thead>
<tr>
<th>S-DIFS</th>
<th>RES_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>LS</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>H</td>
<td>HS</td>
</tr>
</tbody>
</table>

Table III: The final fuzzy decision of FLSAC

\[(F - DEC = (RES_2 \land L - DEC))\]

---

Algorithm 1 FLSAC

1: if \( (BDEV_i \geq \text{thr}_1) \lor (RTR_i \geq \text{thr}_2) \lor (S-DIFS_i \geq \text{thr}_3) \) then
2: The node \( i \) is declared as cheater;
3: else
4: if \( (FLSAC_{dev_i} \land RTR_i \land S-DIFS_i) \equiv C \) then
5: The node \( i \) is declared as cheater;
6: else
7: if \( (FLSAC_{dev_i} \land RTR_i \land S-DIFS_i) \equiv HS \) then
8: if \( (FLSAC_{dev_i} \land RTR_i \land S-DIFS_i) \equiv HS \) then
9: The node \( i \) is declared as cheater;
10: end if
11: end if
12: end if
13: end if
14: \( FLSAC_{dev_i} \) refers to the decision of the current monitoring period and \( FLSAC_{dev_{i-1}} \) refers to the decision of the previous monitoring period.

**thr_1, thr_2 and thr_3** are the thresholds adopted by FLSAC to accuse a given node as a cheater or not.

A. Simulation Environment

For the simulation environment we consider a WLAN that consists of an AP and 8 wireless nodes which are within transmission range of each other. The wireless nodes (including the cheater) are source of CBR traffic (512 bytes/packet and 200 packets/s) towards the common sink (AP) which is also a CBR source to a randomly chosen wireless node. In our simulation we implemented the MAC layer misbehavior techniques, discussed in section II along with the adaptive cheating misbehavior. The results are averaged over 20 simulations, with 120 seconds each. To outline the impact of the different MAC layer misbehavior strategies, we configure the simulation as follows: the cheater first launches each cheating strategy separately and then carries out an adaptive attack by switching
to implement the adaptive cheater behavior, we generate 3 random numbers within the interval $[0, \text{thresh}_i \times MC]$ at the beginning of each monitoring period. These numbers represent the number of times the cheater can deviate from the protocol, in each strategy $i$ (Scramble CTS, BDEV and S-DIFS), without being detected by DOMINO.

As depicted in the Figure 2(d), the cheater gains a considerable bandwidth compared to well behaving nodes. The higher the misbehavior coefficient is, the higher the bandwidth the node gains. However, this bandwidth is lesser than the one earned by a full cheater launching a single strategy solely.

In the sequel, we evaluate the efficiency of our scheme in terms of detection ratio and speed. To do so, we increase the number of wireless nodes to 29 and the number of cheaters to 10. Moreover, we generate three scenarios through which we vary the number of cheaters to 3, 7 and 10 in order to figure out the impact of the number of cheaters on the performance of FLSAC.

The Figure 3(a) reveals that the detection ratio is proportional to the misbehavior coefficient of the cheater and it varies according to the number of adaptive cheaters in the network. When the MC is low (less than 0.5 for the scenarios 2 and 3, and less than 0.4 for the scenario 1) the cheaters can escape from the detection system because their deviation is not large enough to accuse them as cheaters. However, their gain in bandwidth, in this case, is moderate as illustrated previously in the Figure 2(d). When the cheaters increase their MC parameters in order to gain more bandwidth, the detection ratio increases accordingly until it reaches the highest value in the scenario 1. As we can see, the lower the number of cheaters, the higher is the detection ratio. This is due to the fact that when the number of cheaters increases the number of collisions increases excessively and hence it will be hard to the AP running FLSAC to collect enough samples to decide about the nodes’ behavior.

To observe the response latency (detection speed) of FLSAC, we set all the cheaters start misbehaving simultaneously, and we record the detection window number in which the cheater was detected. The MC as well is set to the highest value. As shown in Figure 3(b), the cheater nodes in the scenario 1 were detected earlier than the nodes in scenario 2 and 3. For example, the first cheater in scenario 1 was detected in $DW = 1$, while the first cheater in scenario 2 was detected in $DW = 3$ and the one in scenario 3 in $DW = 6$. The reason of this latency on detecting the cheaters in scenario 3 is the large number of collisions caused by the cheaters preventing the AP from monitoring the BDEV and S-DIFS. Moreover, the number of retransmission of cheaters as well as the well behaving nodes will appear close to each other since the cheater nodes also experience the collision of their own frames.

V. CONCLUSION

This paper proposes a new scheme to cope with the adaptive greedy behaviors in wireless local area networks. Our scheme can be regarded as an extension to the so-called DOMINO scheme, aiming to detect greedy nodes, which might escape from this latter by combining several techniques alternatively and switching intelligently among them.

FLSAC is based on a fuzzy logic controller that merges the observations of three different metrics to conclude whether a node is greedy or not. According to the simulation results, FLSAC can reduce significantly the negative impact of the adaptive greedy behavior. Moreover, FLSAC is lightweight in terms of response speed. As a future work, we are interested in characterizing the greedy behavior in mobile ad hoc networks and developing adequate solution to counter it.

REFERENCES


**Table IV: Simulation settings**

<table>
<thead>
<tr>
<th>Simulation parameters</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>1500m × 1000m</td>
</tr>
<tr>
<td># Wireless nodes</td>
<td>8 (light load) and 29 (heavy load)</td>
</tr>
<tr>
<td>MAC protocol</td>
<td>IEEE 802.11b</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250 m</td>
</tr>
<tr>
<td># Greedy nodes</td>
<td>1, 3, 7 and 10 (4 scenarios)</td>
</tr>
<tr>
<td>FLSAC</td>
<td>running on AP (Access Point)</td>
</tr>
<tr>
<td>Switching scheme</td>
<td>Random</td>
</tr>
<tr>
<td>Traffic type</td>
<td>CBR</td>
</tr>
<tr>
<td>Data rate</td>
<td>11mbps</td>
</tr>
<tr>
<td>CBR packets size</td>
<td>512 bytes</td>
</tr>
<tr>
<td>Monitoring period</td>
<td>10 s</td>
</tr>
<tr>
<td># Simulation epochs</td>
<td>20</td>
</tr>
</tbody>
</table>

We use the mean of throughput of the other 7 nodes as there is a common bandwidth fair share for each of them.
(a) Impact of backoff manipulation on Throughput

(b) Impact of DIFS value reduction on Throughput

(c) Impact of the proportion of scrambled CTS packets on Throughput

(d) Impact of the adaptive cheater on Throughput

Figure 2: Impact of MAC layer misbehavior on throughput in WLAN

(a) Detection ratio versus misbehavior coefficient

(b) Detection speed of FLSAC

Figure 3: Assessment of FLSAC’s robustness