# A Bayesian Statistical Model to Alleviate Greediness in Wireless Mesh Networks

Soufiene Djahel, Youcef Begriche and Farid Naït-Abdesselam

$^{\mp}$LIFL – UMR CNRS USTL 8022 – IRCICA
University of Lille, France
{soufiene.djahel, Youcef.Begriche, naf }@ieee.org

***Abstract*— Wireless mesh Networks (WMNs) are a prominent paradigm of wireless communication that have been widely used in many applications. The growing popularity of such networks opened the door to a profusion of attacks that may target their core functioning leading to a harmful impact on their performance. Hence, the need of robust and fast detection of those attacks became a major prerequisite in order to guarantee an efficient and fair share of network resources among nodes. One of the well known devastating attacks is MAC layer misbehavior which may lead to severe collapse of network performance. In this study, we focus on such misbehavior and in particular on the adaptive greedy behavior of a node in wireless mesh network environment. In such environment, wireless nodes compete to gain access to the medium in order to communicate with a mesh router (MR). In this case, a greedy node may violate the MAC protocol rules to earn extra bandwidth share upon its neighbors. To evade from detection, the cheater node may use more than one technique and switch dynamically between each of them. To counter such misuse, we propose to extend our previous solution, dubbed FLSAC, through the use of a Bayesian statistical model. This new scheme is implemented in conjunction with FLSAC at the mesh router/gateway to monitor the behavior of the attached wireless nodes and detect any deviation from the proper protocol rules. The simulation results reveal that this new solution outperforms both of DOMINO and FLSAC in terms of detection rate and accuracy.**

***Keywords* – Wireless Mesh Networks, MAC Layer misbehavior, Adaptive Cheater, Bayesian Statistical Model, FLSAC.**

## I. Introduction

In recent years, Wireless Mesh Networks (WMNs) have emerged as a novel and prominent paradigm of wireless communication. A mesh network is made of both wireless and wired nodes forming a mesh topology, as shown in Figure 1. WMNs can be seen as a three levels network where the nodes belonging to each level have a specific role to accomplish which is generally different from the task of other levels' nodes. At the highest level, we find the gateways which are usually equipped with multiple interfaces (wired and wireless) and serve as internet access points for the mesh nodes (mesh clients). These gateways can be either stationary (e.g. rooftop) or mobile (e.g, airplane, buses/subway). At the middle level, a large number of mesh routers (MRs) is needed in order to provide reliable service. Each router has at least one wireless interface and acts as a repeater to transmit data from nearby routers/clients to peers that are too far away to reach. Finally, the mesh clients are situated at the lowest level; these clients are the only sources/destinations for data traffic flows in the network. The connection to the mesh network is provided through wireless routers (or directly through the gateways).

Since IEEE 802.11 MAC protocol, as described in [3], 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 its root causes [4], [5]. As a result of this investigation, a bunch of solutions have been proposed in the literature to cope with this problem such as the works done in [6], [7] and [8]. These works have identified several types of MAC layer misbehavior, and proposed 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 face a smart cheater which might be aware of the functioning of the deployed detection scheme. Such cheater exploits its knowledge to escape from being detected.

In this paper, we conduct an in depth analysis of the adaptive cheater [9] misbehavior in IEEE 802.11 MAC protocol. In such misbehavior, the cheater node prefers to frequently switch between several cheating strategies rather than applying one technique; thereby it avoids detection and makes the task of the observer/monitor node harder. To tackle this problem, we first explain how easy this can be performed in IEEE 802.11 MAC protocol. Then, we present our Bayesian statistical model that aims to detect these misbehaving nodes through a probabilistic computation and ensure a lower false accusation and misdetection rate. To this end, we have opted for Bayes theorem to develop our cheating probability. This probability is calculated based on an estimation of a set of MAC parameters that might be modified by the cheater node to gain fast access to the wireless medium. Later on, we have integrated this model with FLSAC scheme in order to enhance its detection rate and accuracy.

The rest of the paper is organized as follows. In section **??**, we present the literature, followed by a detailed description of the proposed solution in section II. Then, the simulation results are reported and discussed in section III. Finally, section IV concludes the paper.

## II. The proposed solution

In what follows, we present our solution and provide detailed proof of its correctness.

### A. Motivations

The main reason that incites us to investigate the adaptive cheater behavior in WMNs is the devastating consequences that may be induced from this misbehavior due to the architecture and particular characteristics of these networks. Since the mesh routers are connected to each other through wireless links then any mischief of any client attached to them will affect both of packets delivery towards clients, and the forwarding of their packets towards far away clients or internet.

Let us now suppose that the carrier sensing range ($R_{cs}$) of a mesh client is slightly larger than its transmission range, which is considered as the best case regarding the propagation of the greedy

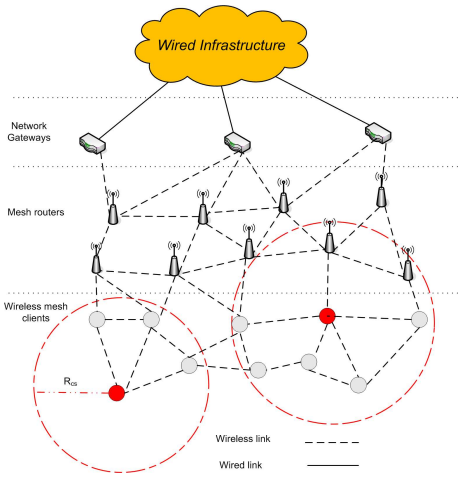



Figure 1: Wireless Mesh Networks model

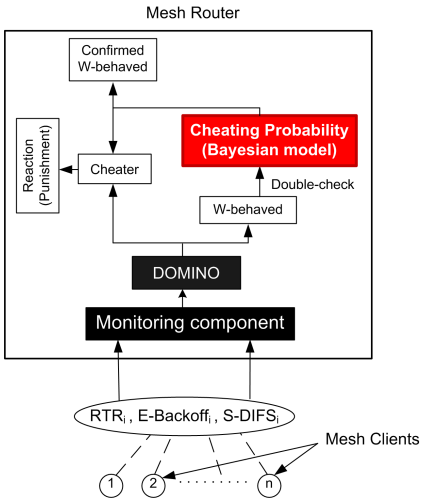



Figure 2: The operations of our scheme

behavior impact in the network. As shown in Figure. 1, when a misbehaving client (the red mesh client) violates the MAC protocol rules, all the wireless links whose at least one of their vertices (either client or mesh router) is within the $R_{cs}$ of this misbehaving client are paralyzed. Consequently, no communication is allowed over them, as long as the misbehaving client is still gaining the competition to access the medium using illegitimate ways.

As compared to MANETs, the impact of MAC layer misbehavior is more damaging in WMNs. This is due to the fact that the lower mobility of mesh routers extends the duration of their incapability (i.e. the sharp decrease of their acquired throughput) of delivering (forwarding) the frames to clients (neighboring routers), respectively, because the medium is being monopolized by the cheater node. However, in MANETs the high mobility of nodes may be useful to escape from the cheater range and thus minimizing the induced impairment.

### B. key idea

The main idea of this work can be described as follows. At the end of each monitoring window $W$, we test the hypothesis that a given mesh client (M-client) is an adaptive cheater even though DOMINO has classified it as well behaved, as depicted in Figure. 2. To do so, we compute the likelihood that a M-client is cheating given the evidence E. This evidence is consisted of the collected statistics by the MR regarding the different parameters, cited below, that characterize the MAC protocol. The resulted probability is then compared to a threshold value $\alpha$, dynamically updated by the MR, in order to classify this M-client as cheater or honest. Notice that $\alpha$ is updated according to the variation of mesh clients' density around the MR and the collision rate.

### C. Model description

This model is based on a fundamental Bayesian principle that allows us to make the correct decision regarding the M-clients' behavior. The behavior of each M-client is observed by the MR through the estimation of the following parameters that determine the winner of the medium access contention between neighboring M-clients.

- the observed idle slots between consecutive transmissions from the same M-client.
- the observed idle slots between two transmissions from the same M-client interspersed by other transmissions.
- the observed retransmissions rate of each M-client: this metric is calculated through the comparison of the number of failed transmissions of a given M-client with the average of that of the other M-clients attached to the same MR. In practice this information can be extracted from the Retry field of the MAC frame.
- the observed deviation of a M-client from the required DIFS duration, i.e. the difference between the DIFS value and the real period (S-DIFS: short DIFS in case of cheater M-client) that the cheater M-client has waited before decrementing its backoff.

In the rest of the paper, we refer to these parameters as $p_1$, $p_2$, $p_3$ and $p_4$, respectively. Notice also that the parameters $p_1$ and $p_2$ allow us to estimate the backoff of a given mesh client (E-Backoff), whereas $p_3$ defines the retransmission rate (RTR) of the M-client.

Another parameter may also be considered since it represents a way to disobey the MAC protocol. This parameter is the difference between the advertised value of the duration filed of RTS or DATA frames and the effective transmission time of the ongoing transmission. The cheating method that exploits this field is the NAV oversize technique. In this technique, the sender of RTS frame amplifies the value of the duration field in order to increase the deferment delay of the receivers M-clients; thereby a DoS attack or bandwidth under utilization may be resulted. Notice that this technique is rarely applied by the cheaters since it can be easily detected by their neighbors, compared to the previous techniques. Therefore, we neglect this parameter in our work.

*1) **Computation of the cheating probability**:* Based on the values of the previous parameters we use Bayes theorem to calculate the cheating probability as described below.

If A and B are two events, the Bayes'formula :

$$\mathcal{P}(A/B) = \frac{\mathcal{P}(B/A).\mathcal{P}(A)}{\mathcal{P}(B)} \tag{1}$$

gives the conditional probability of A knowing B. This formula can be applied to our problem as follows:
If a certain M-client is represented by the vector $(p_1, \ldots, p_m)$, then the probability that this M-client is an adaptive cheater is expressed as

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\mathcal{P}(che/p_1) \cdot \ldots \cdot \mathcal{P}(che/p_m)}{\mathcal{P}(che/p_1) \cdot \ldots \cdot \mathcal{P}(che/p_m) + R_1 \cdot R_2} \tag{2}$$

where

$$R_1 = (\frac{\mathcal{P}(che)}{\mathcal{P}(hon)})^{(m-1)}$$

and

$$R_2 = (1 - \mathcal{P}(che/p_1)) \cdot \ldots \cdot (1 - \mathcal{P}(che/p_m))$$

Notice that $p_i$ indicates that the M-client has been observed cheating by manipulating the MAC parameter of index $i$.

We did an earlier classification of the acquired observations of the parameter $p_1$ and $p_2$, as described below. For each observed value of these two parameters of a given M-client, the MR checks if it is obeying the Backoff rules or not through a simple comparison with the previous collected values of $p_i$ within the same monitoring window, as described in the following.

If $obs_j(p_i)$ - $\frac{\sum_{k=1}^{J-1} obs_k(p_i}{(J-1)} > \delta$ then the counter of the suspected observations ($cpt_1$ ($p_i$) is incremented, otherwise the counter of legitimate ones ($cpt_2(p_i)$ is increased. We assume that at the end of a given monitoring window, the MR has collected $N$ observations of all the parameters of a M-client. Thus, we calculate a prior probability that this M-client is misbehaving ($\mathcal{P}$ (susp)) and a prior probability that it is obeying the rules ($\mathcal{P}$ (leg)) as follows.

$$\mathcal{P}(susp) = \frac{\frac{\sum_{i=1}^{3} cpt_1(p_i)}{3}}{N} \tag{3}$$

$$\mathcal{P}(leg) = \frac{cpt_2}{N} \tag{4}$$

Notice that we didn't take into account $p_3$ and $p_4$ for calculating those prior probability because

***Proof of correctness of Eq. 2:*** We now prove the correctness of our formula, given in Eq. 2, for any integer $m$ although in our solution we use that only for $m$ = 4.
According to the basic Bayes formula, we have

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\mathcal{P}(p_1, \ldots, p_m/che) \cdot \mathcal{P}(che)}{\mathcal{P}(p_1, \ldots, p_m/che)(\mathcal{P}(che) + C)} \tag{5}$$

where

$$C = \mathcal{P}(p_1, \ldots, p_m/hon)\mathcal{P}(hon)$$

Assuming that the set of DCF parameters is an independent set, then we get the following formula

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\mathcal{P}(p_1/che) \cdot \ldots \cdot \mathcal{P}(p_m/che) \cdot \mathcal{P}(che)}{C_1 + C_2} \tag{6}$$

such that

$$\begin{aligned} C_1 &= \mathcal{P}(p_1/che) \cdot \ldots \cdot \mathcal{P}(p_m/che) \cdot \mathcal{P}(che) \\ C_2 &= \mathcal{P}(p_1/hon) \cdot \ldots \cdot \mathcal{P}(p_m/hon) \cdot \mathcal{P}(hon) \end{aligned}$$

Let us now consider the following rules issued from Bayes theorem

$$\mathcal{P}(p_i/che) = \frac{\mathcal{P}(che/p_i) \cdot \mathcal{P}(p_i)}{\mathcal{P}(che)}$$
$$\mathcal{P}(p_i/hon) = \frac{\mathcal{P}(hon/p_i) \cdot \mathcal{P}(p_i)}{\mathcal{P}(hon)}$$

By applying those rules on the Eq. 6, we obtain

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\frac{\mathcal{P}(che/p_1) \cdot \mathcal{P}(p_1)}{\mathcal{P}(che)} \cdot \ldots \cdot \frac{\mathcal{P}(che/p_m) \cdot \mathcal{P}(p_m)}{\mathcal{P}(che)}}{d_1 \cdot \ldots \cdot d_m \cdot \mathcal{P}(che) + \overline{d_1} \cdot \ldots \cdot \overline{d_m} \cdot \mathcal{P}(hon)} \tag{7}$$

with

$$d_i = \frac{\mathcal{P}(che/p_i) \cdot \mathcal{P}(p_i)}{\mathcal{P}(che)} \;\; and \;\; \overline{d_i} = \frac{\mathcal{P}(hon/p_i) \cdot \mathcal{P}(p_i)}{\mathcal{P}(hon)}$$

As a result, we obtain the following formula

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\prod_{i=1}^{i=m} \mathcal{P}(p_i)[\mathcal{P}(che/p_1) \cdot \ldots \mathcal{P}(che/p_m)]}{[\mathcal{P}(che)]^{(m-1)} \cdot \prod_{i=1}^{i=n} \mathcal{P}(p_i)[E + \overline{E}]} \tag{8}$$

where

$$E = \frac{\mathcal{P}(che/p_1) \cdot \mathcal{P}(che/p_m)}{\mathcal{P}(che)^{(m-1)}}$$

*and*

$$\overline{E} = \frac{\mathcal{P}(hon/p_1) \cdot \mathcal{P}(hon/p_m)}{\mathcal{P}(hon)^{(m-1)}}$$

So, after applying a set of simplifications we obtain the following result

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\mathcal{P}(che/p_1) \cdot \ldots \cdot \mathcal{P}(che/p_m)}{\mathcal{P}(che/p_1) \cdot \ldots \cdot \mathcal{P}(che/p_m) + F} \tag{9}$$

such that

$$F = [R_1 \cdot \mathcal{P}(hon/p_1) \cdot \ldots \cdot \mathcal{P}(hon/p_m)]$$

Since $che$ and $hon$ are two complementary events, and

$$\mathcal{P}(\overline{A}/B) = 1 - \mathcal{P}(A/B)$$

for any two events A and B, thus by applying this rule on the Eq. 9 we obtain the same formula of the Eq. 2. Therefore, the correctness of our solution is proven.

*2) **Classification criteria:*** A M-client represented by the set of parameters: $p_1$, $p_2$, …, $p_m$ is classified as a cheater when:

$$\frac{\mathcal{P}(che/p_1 \cdot \ldots \cdot p_m)}{1 - \mathcal{P}(che/p_1, \ldots, p_m)} > \lambda \tag{10}$$

therefore the selection criteria is equivalent to $\mathcal{P}(che/p_1, \ldots, p_m) > \alpha$ with $\alpha = \frac{\lambda}{1+\lambda}$

*3) **Filter evaluation methodology**:* A filter performance is based on two parameters which are its accuracy (Acc), as defined in [2], and the error (Err = 1 - Acc), respectively defined as

$$Acc = \frac{N_{che\rightarrow che} + N_{hon\rightarrow hon}}{N} \quad (11)$$

$$Err = \frac{N_{che\rightarrow hon} + N_{hon\rightarrow che}}{N} \quad (12)$$

where $N_{x\rightarrow y}$ denotes the number of M-clients of class $x$ which are erroneously classified in class $y$, and $N_{x\rightarrow x}$ represents the number of M-clients of class $x$ which are correctly classified.

Referring to the above notation, we define below the weighted accuracy and error that take into consideration the weight assigned to classification failures.

$$Wacc = \frac{\lambda N_{che\rightarrow che} + N_{hon\rightarrow hon}}{\lambda N_{che} + N_{hon}} \quad (13)$$

$$Werr = \frac{\lambda N_{che\rightarrow hon} + N_{hon\rightarrow che}}{\lambda N_{che} + N_{hon}} \quad (14)$$

where $N_{che}$ and $N_{hon}$ refer to the number of cheaters and honest M-clients, respectively.

To assess the performance of this filter we compare it to a non filtered network where every M-client is considered as honest, thus granting network access to every cheater M-client. According to the definition of $Wacc$ and $Werr$, the referential weighted error and weighted accuracy (respectively noted $Wacc^b$ and $Werr^b$) are calculated as follows:

$$Wacc^b = \frac{\lambda N_{che}}{\lambda N_{che} + N_{hon}} \quad (15)$$

$$Werr^b = \frac{N_{hon}}{\lambda N_{che} + N_{hon}} \quad (16)$$

These values allow the performance of the filter to be compared to that of the baseline, hence the Total Cost Ratio (TCR) is defined as:

$$TCR = \frac{Werr^b}{Werr} = \frac{N_{hon}}{\lambda N_{che\rightarrow hon} + N_{hon\rightarrow che}} \quad (17)$$

Notice that high TCR values reflect a good filter while values being smaller than 1 indicate that the reference filter outperforms the evaluated one. Finally, we calculate the ratio of the honest M-clients correctly classified by the filter (i.e, Node Recall (NR)), and the precision of this filter when it classifies the M-clients as honest (i.e. Node Precision (NP)). These two metrics are defined as

$$NR = \frac{N_{hon\rightarrow hon}}{N_{hon}} \quad (18)$$

$$NP = \frac{N_{hon\rightarrow hon}}{N_{hon\rightarrow hon} + N_{che\rightarrow hon}} \quad (19)$$

The impact of NR and NP on the network performances is highly dependent on the filter context. Similarly to the parameter $\lambda$, the weights of NR and NP are influenced by the action taken based on filter decisions. Therefore, NR and NP values are not relevant in a context independent comparison of filters. A better solution to compare the efficiency of two filters is to rely on the TCR.

**Algorithm 1** Interaction between FLSAC and our bayesian model

```
At the end of the monitoring window W_i;
if (FLSAC_decision == (Normal ∨ L-susp)) then
    status(M-client_id)= honest;
else
    if (FLSAC_decision == H-susp) then
        β = (Σ_{j=1}^{k} FLSAC_output(Normal∨L−susp)) / (2·l);
        α= FLSAC_output (current window W_i) + β;
        if (Bayes_prob(cheater)>= α) then
            status(M-client_id) = cheater; //confirmed cheater
        else
            Defer the decision till the end of the subsequent
            monitoring window (W_{i+1});
        end if
    else
        if (FLSAC_decision == cheater) then
            α= FLSAC_output;
            if (Bayes_prob(cheater)≥ α) then
                status(M-client_id) = cheater;
            else
                Defer the decision till the end of the subsequent
                monitoring window (W_{i+1});
            end if
        end if
    end if
end if
```

*4) Integration of the bayesian model with FLSAC:* When we integrate our model with FLSAC, the output of this latter will define the classification criteria $\alpha$ used for identifying the cheater M-clients. Hence, for better understanding of the interaction between these two schemes we distinguish three different cases as follows:

- FLSAC's output reveals that the M-client is classified as either normally behaving or lowly suspected (L-susp). In this case the bayesian cheating probability is ignored.
- if FLSAC's decision is highly suspected (H-susp) then the value of $\alpha$ is calculated as the addition of the current FLSAC's output and the half of the average of the last $k$ outputs, where the decision was either normal or lowly suspected.
- if FLSAC judges that the M-client is cheater then $\alpha$ is assigned the value of FLSAC's output.

Algorithm 1 provide a pseudo code that explains how a MR combines both of FLSAC and the Bayesian model to make the final decision regarding the M-client's behavior.

| Parameters | Values |
|---|---|
| Area | 2000m · 2000m |
| Physical layer | Direct sequence |
| No. of cheaters | 4..8 |
| Transmission range of clients | 250m |
| Transmission range of MRs | 400 m |
| Topology | Random |
| | 20 MRs |
| | 20 clients per MR |
| Traffic type | CBR |
| Data rate | 5.5 mbps |
| CBR packets size | 500 bytes |
| Simulation time | 300 seconds |
| No. of simulation epochs | 10 |
| Network simulator | OPNET 14.0 [1] |

Table I: Simulation settings

## III. Simulation settings and results

In this section, we present and interpret the obtained results that quantify the performance of our proposed solution as compared to DOMINO and FLSAC, in terms of detection rate and accuracy. Simulations are performed using the network simulator OPNET 14.0 which we have extended by adding new functions, required for our solution, to the MAC layer (*wlan-mac* process model). The simulation settings and configuration parameters of each M-client and MR are summarized in table I.

We have conducted experiments using CBR traffic where a source M-client sends a CBR stream to a distant M-client that is attached to a different MR. In our configuration we vary the percentage of the cheater M-clients attached to the same MR from 20% to 40%. Notice that the switching scheme used by the cheater M-clients to alternate between the cheating techniques is similar to that used in [9]. We also vary the classification criteria $\alpha$ of our Bayesian model (B_Model) from 0.55 to 0.85. A summary of the different scenarios used in our simulation is presented in table II.

Figure 3 plots and compares the detection rates of DOMINO, FLSAC, B_Model and FLSAC+B_Model [1](FB_Model). Except DOMINO which, as expected, fails totally to detect the adaptive cheaters in all scenarios and with varying misbehavior coefficient (MC), the other schemes achieve an acceptable detection rate. This failure of DOMINO is justified by the fact that it assesses the observed deviation of each MAC layer parameter independently from the other parameters. Thus, since the adaptive cheaters, implemented in our simulation, don't deviate so much by manipulating one MAC parameter but they get benefits from the combined deviation of several cheating techniques, they succeed to escape from DOMINO.

In general, we observe that the detection rate increases as the MC increases till it reaches its highest values when the MC value gets closer to 1. In this latter case, the cheater's deviation is high and therefore easy to be distinguished from the normal behavior by both the fuzzy controller of FLSAC and the Bayesian probability of B_Model, especially when the number of cheaters is quite low. We also remark that the raise of the cheater's percentage negatively affects all the schemes, in particular FLSAC, since the multiple collisions provoked by the cheaters prevent the MR from either collecting enough samples of observations or correctly estimating the values of certain parameters.

[1]The notation FLSAC+B_Model refers to the scheme resulted from the integration of B_Model with FLSAC as described in section II-C.4.

| Scenario | Percentage of cheater M-clients per MR | $\alpha$ |
|---|---|---|
| 1 | 20 % | 0.55 |
| 2 | 30 % | 0.65 |
| 3 | 35 % | 0.75 |
| 4 | 40 % | 0.85 |

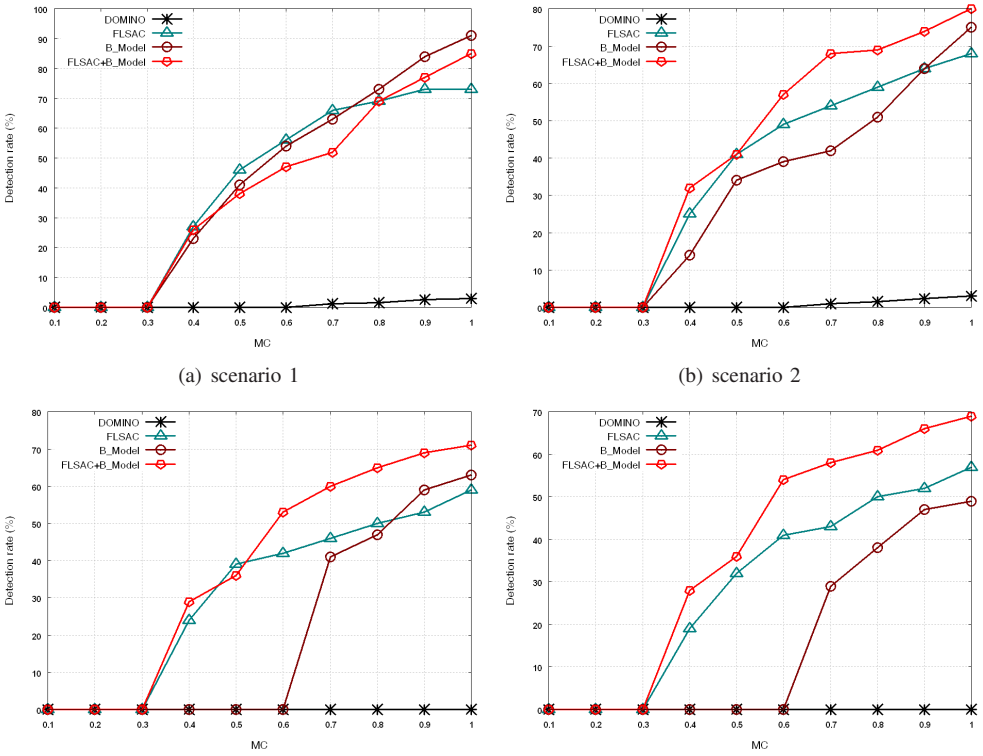
Table II: Scenarios setting

From the curves plotted in Figure 3(a), we observe that the B_Model outperforms the other schemes as the value of $\alpha$ is small enough to let the MR easily recognize the cheater M-clients. So, it ensures that most of the cheater M-clients are detected even if they are slightly deviating from the standard, at the expense of some wrong accusation of well behaved M-clients.

The Figures 3(b), 3(c) and 3(d) divulge that the increase of $\alpha$ leads to a sharp decrease of B_Model's detection rate since the large value of $\alpha$ allows only detection of a small portion of the deviating M-clients, whereas the rest of the deviating ones are wrongly classified as well behaved. Additionally, these figures show that the $FB\_Model$ significantly outperforms the two other schemes. This is due to the following reasons: (i) the use of FLSAC's output, which is dynamically updated, as a classification criteria of B_Model allows it to detect more cheaters as compared to the case where we use a fixed value of $\alpha$, (ii) the new defined classification criteria for the M-clients that have been classified as H-susp by FLSAC ensures their detection if they are misbehaving, hence those M-clients cannot escape from FB_Model as they have done with FLSAC (see Algorithm 1).

The histogram plotted in Figure 4 highlights the accuracy of the decisions taken by the three schemes. Notice that the graphed values have been calculated based on the filter presented in section II-C.3. As we can see from this histogram, the higher the value of $\alpha$ is the lower accuracy of B_Model because when $\alpha$ rises the interval [0.5, $\alpha$] gets larger. Thus, the cheater M-clients whose the cheating probability belongs to this interval will be wrongly classified as legitimate and consequently the detection accuracy drops sharply to less than 60% in scenario 4. Compared to FLSAC and B_Model, the FB_Model shows the highest accuracy in all scenarios, whereas FLSAC outperforms B_Model, particularly in scenarios 3 and 4 where the gap between them is important. This supremacy of BF_Model over the other schemes is due to the same reasons explained in the previous paragraph. Notice that we neglect the detection accuracy of DOMINO since this latter presents detection rate of around 2% in the best case, thus it is insignificant to calculate its accuracy.

To conclude, the results presented above confirm that our




(a) scenario 1 (b) scenario 2

(c) scenario 3 (d) scenario 4

Figure 3: Detection rate in different scenarios with varying MC values

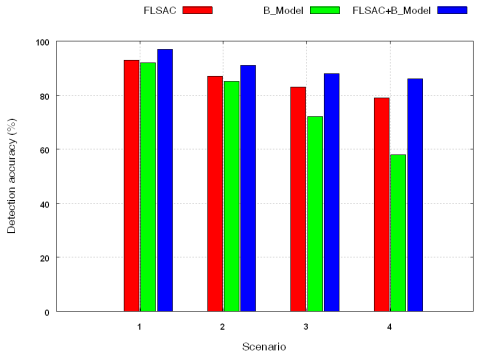



Figure 4: Detection accuracy in different scenarios

choice of combining FLSAC and B_Model was a right decision, since this hybrid solution shows a high detection rate and accuracy in a mesh network dominated by a large number of cheaters (till 40% of the M-clients are cheaters in our simulations).

## IV. Conclusion

In this work, we have presented a new scheme based on a Bayesian statistical model in order to detect the cheater M-clients that apply a bunch of misbehavior techniques and switch intelligently among them to evade detection. This scheme is implemented at the mesh router which is the responsible for collecting information regarding certain parameters used by the M-clients to transmit their packets. Based on these information our scheme calculates the cheating probability and then combines it with FLSAC's output to make the final decision regarding a given mesh client's behavior. According to the simulation results, our scheme can significantly reduce the negative impact of the cheater clients even in a network dominated by cheaters. Therefore this proves that Bayes theory stills an efficient model that can be exploited to defend against other misbehavior in WMNs. As a future work, we are interested to tackle the jamming attack and develop robust solution to cope with it.

## References

[1] OPNET Technologies, "OPNET Modeler", http://www.opnet.com/.

[2] M. sahami, M. Dumais, S. Heckerman and E. Horvitz, "A Bayesian Approach to Filtering Junk E-mail", *In Proc. of the Workshop of learning for text categorization -AAAI*, Madison Winsconsin, 1998.

[3] IEEE 802.11 wireless LAN media access control (MAC) and physical layer (PHY) specifications, 1999.

[4] V. Gupta, S. Krishnamurthy and M. Faloutsos, "Denial of service attacks at the MAC layer in wireless ad hoc networks", *In Proc. of Military Communication Conference MILCOM 02*, Anaheim, CA, Oct. 2002.

[5] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions", *In Proc. of the* $12^{th}$ *USENIX Security Symposium*, Washington, DC, USA, Aug. 2003.

[6] A. A. Cardenas, S. Radosavac and J. S. Baras, "Detection and Prevention of MAC layer misbehavior in Ad Hoc Networks", *In ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington DC, USA, Oct. 25, 2004.

[7] P. Kyasanur, and N. H. Vaidya, "Selfish MAC Layer misbehavior in Wireless Networks". *IEEE TRANSACTIONS ON MOBILE COMPUTING*, Vol. 4, No. 5, p 502-516, Sept./Oct. 2005.

[8] M. Raya, I. Aad, J. P. Hubaux and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots". *IEEE TRANSACTIONS ON MOBILE COMPUTING*, Vol. 5, No. 12, p 1691-1705, Dec. 2006.

[9] S. Djahel and F. Naït-Abdesselam, " FLSAC: A New Scheme to Defend Against Greedy Behavior in Wireless Mesh Networks", *International Journal of Communication Systems (IJCS), Wiley InterScience Publisher*, Vol. 22, No. 10, pp. 1245-1266, jun. 2009.