FLSAC: A New Scheme to Defend Against Greedy Behavior in Wireless Mesh Networks

Soufiene Djahel*and Farid Naït-Abdesselam

LIFL – UMR USTL CNRS 8022 – IRCICA
University of Lille, France

SUMMARY

The most commonly used medium access mechanism in wireless mesh networks is based on the CSMA/CA protocol. This protocol schedules properly the access to the medium among all the competing nodes. However, in a hostile environment, such as wireless mesh networks (WMNs), selfish or greedy behaving nodes may prefer to decline the proper use of the protocol rules in order to increase their bandwidth shares at the expense of the well behaving nodes. In this paper, we focus on such misbehavior and in particular on the adaptive greedy misbehavior of a node in the context of wireless mesh network environment. In such environment, wireless nodes compete to gain access to the medium and communicate with a mesh router (MR). In this case, a greedy node may violate the protocol rules in order to earn extra bandwidth share upon its neighbors. In order to avoid its detection, this node may adopt different techniques and switch dynamically between each of them. To counter such misbehavior, we propose to use a fuzzy logic based detection scheme. This scheme, dubbed FLSAC, is implemented in the mesh router/gateway to monitor the behavior of the attached wireless nodes and report any deviation from the proper use of the protocol. The simulation results of the proposed FLSAC scheme show robustness and its ability to detect and identify quickly any adaptive cheater.

KEY WORDS: *Wireless Mesh Networks, MAC Layer Misbehavior, Adaptive Cheater, Fuzzy Logic.*

1. 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 newest model being introduced is the Wireless Mesh Network (WMN), in which nodes called Mesh Clients (MC), within the transmission range of each others, can

*Correspondence to: Soufiene Djahel, IRCICA, Parc scientifique de la haute borne, 59650, Villeneuve d'Ascq, France.
E-mail: soufiene.djahel@lifl.fr

communicate directly over the wireless link, while those that are far apart use other nodes called Mesh Routers (MR) as relays. The properties of WMNs, such as shared wireless medium, open network architecture and stringent resource constraints for mesh clients make the secure communication a hard task to achieve. Owing to these characteristics, WMNs are vulnerable to several types of security attacks at different layers. Especially due to leveraging security flaws in the IEEE 802.11 MAC protocol [17] and the salient features of WMNs, MAC-layer attacks are easy to target.

Since IEEE 802.11 MAC protocol 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 considerable research attention leading to an in depth investigation and analysis of the root causes of this misbehavior as shown in [11], [10] and [22]. As a result of this investigation, some pioneering contributions have been proposed in the literature to cope with this problem such as [12], [15] and [21]. 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 of 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 to detect and identify the 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 2, we give a brief description of WMNs and their features along with an overview of the commonly used MAC protocol and its vulnerabilities. Next we summarize the literature in section 3. Section 4 addresses the components and the detailed description of the proposed FLSAC scheme. In section 5, we report our simulations and discuss the obtained results. Finally, section 6 concludes the paper and gives some future investigations.

2. PRELIMINARIES ON WMNs AND IEEE 802.11 MAC PROTOCOL

In this section, we exhibit some features of WMNs and give a brief overview of MAC protocol rules in order to reveal the possible threats at this level which might affect the performance of WMNs.

2.1. Wireless Mesh Networks Overview

WMN is a new 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. The gateways are usually equipped with multiple interfaces (wired and wireless) and serves as Internet access points to client nodes (mesh nodes (MNs)). These gateways can be either stationary (e.g. rooftop) or mobile (e.g, airplane, buses/subway). In a mesh network, a large number of mesh routers (MRs) is needed in order to provide a 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. The mesh 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).

We outline below the most representing features of WMN [14];

- WMN provides support for ad hoc networking with capability of self-forming, self-healing and self-organization, along with significant enhancements in the network performance and ease of deployment.
- As opposed to ad hoc network, WMN provides larger coverage, connectivity and robustness due to redundancy.
- WMN can integrate with several types of networks, including Internet, WiMAX, WSN, etc., using gateways technologies.
- Mesh connectivity significantly enhances network performance, such as fault tolerance, load balancing, throughput and protocol efficiency.

As a consequence of the above mentioned characteristics, WMN is widely used in many applications. For this reason, the security issue in WMN needs to be addressed for identifying the potential threats that can target such technology. In this work, we analyze the communication between mesh clients and mesh routers to discover the possible misbehaviors. Especially, we focus on MAC layer vulnerabilities as any violation of the protocol at this level affects all the upper layers and leads to performance collapse.

2.2. IEEE 802.11 MAC protocol overview

The 802.11 standard [6] specifies a common medium access control (MAC) layer, which provides a variety of functions that support the operation of wireless access. In general, the MAC layer manages and maintains communication between 802.11 stations (radio network cards and access points) by coordinating access to a shared radio channel and utilizing protocols that enhance communication over a wireless medium. Often viewed as the "brain" of the network, the 802.11 MAC layer uses a dedicated physical layer, such as 802.11b or 802.11a, to perform the tasks of carrier sensing, transmission, and receiving of 802.11 frames.

The IEEE 802.11 MAC protocol supports two types of access methods. The basic access method is the distributed coordination function (DCF), which is a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. It is designed to support best effort traffic, like internet data, that does not require any service guarantees. Additionally, IEEE 802.11 also incorporates an optional access method in which access point performs the polling to determine which station has the right to transmit resulting in a contention free communication. This method is known as the point coordination function (PCF) and is generally used in scenarios where service guarantees are needed. As stated earlier, PCF is an optional access mechanism which can be used only in the presence of an access point, concurrently with DCF.

Since the emphasis in this paper is on the WMNs, therefore DCF will be the primary access method. In the DCF mode, a station shall ensure that the medium is idle before attempting to transmit. It selects a random backoff value less than or equal to the current contention window (CW) size, and decreases the backoff timer by one at each time slot when the medium is idle. A station may wait for DIFS (DCF Inter Frame Space) time slot after a successful transmission or EIFS (Extended Inter frame Space) period when collision occurs. If the medium is determined to be busy, the station freezes its backoff timer and sets its NAV (Network Allocation Vector)

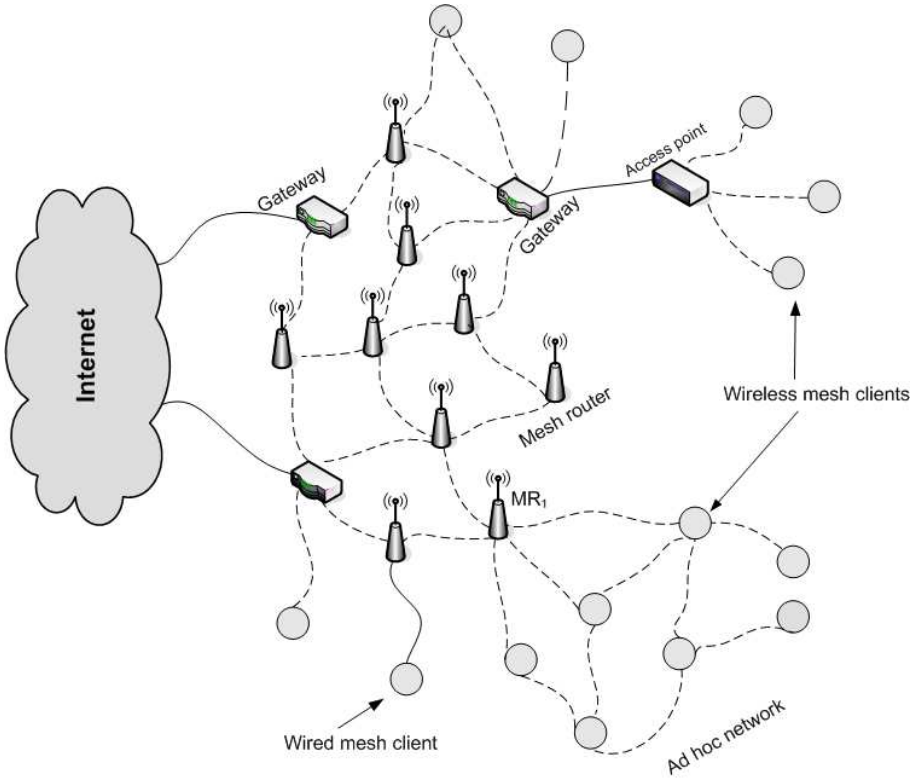


Figure 1. Wireless Mesh Network Architecture

to the expected duration of transmission indicated in the received frame. Transmission shall start whenever the backoff timer reaches zero.

When a CTS or an ACK of a packet is not received in a certain time, the sender assumes a transmission failure. A transmission failure, like collision, leads the station to invoke the backoff procedure by selecting a random number in the interval [0, CW]. After each successful transmission, the size of CW is initialized to CWmin. But in case of each unsuccessful attempt the CW is doubled until it reaches CWmax and remains the same till it is reset. Then, it is reset either when the packet is delivered successfully or maximum retry limit is reached. In case of latter, the retransmission shall stop, CW will be reset to CWmin and the packet is discarded.

2.3. Adaptive cheater in DCF mode

A misbehaving node may disobey the MAC protocol rules to gain more bandwidth over the regularly behaving honest nodes. To do so, it should change the MAC layer parameters. A node is able to change these parameters only in network access cards that run the MAC protocol in software instead of hardware or firmware. In this case, the misbehaving user 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, $k \times CWmin$] 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 when experiencing unsuccessful transmission. Additionally, it may access the medium without passing through the backoff procedure or always wait for a constant short period.
- 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 as depicted in Figure 2(b); otherwise the result is similar to the case when the cheater chooses a small backoff as illustrated in Figure 2(a).
- 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 value as illustrated in Figure 3. 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.
- Increases its bandwidth share by launching a cross-layer attack targeting the routing protocols to decrease the number of contending nodes around it. The cheater can carry out this attack by denying response to the incoming RTS frames without any alteration of the other MAC parameters. Therefore, it increases its chance to gain more access to the medium.

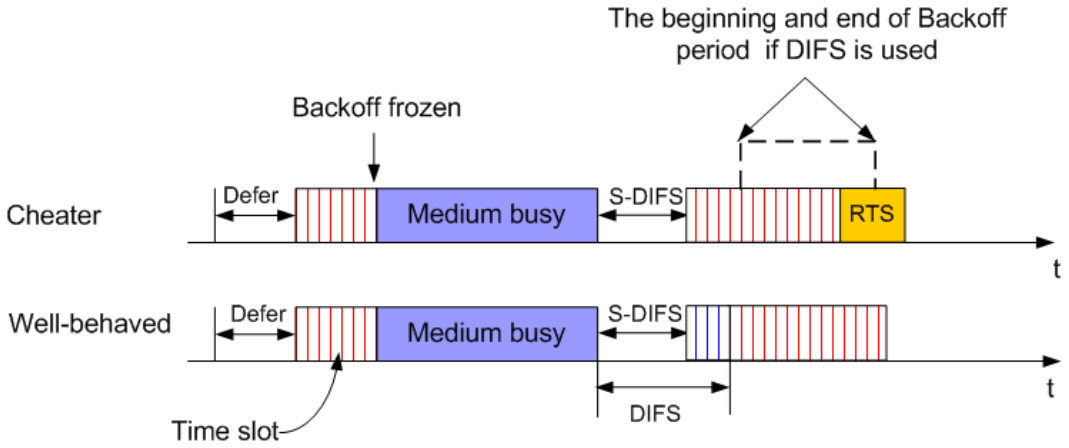
A more sophisticated cheater, which has some knowledge about the deployed detection system, can easily switch frequently between these several strategies to avoid detection. Moreover, it is worth noting that it is not compulsory for the cheater to know the actual parameters of the detection system. A 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.

3. RELATED WORK

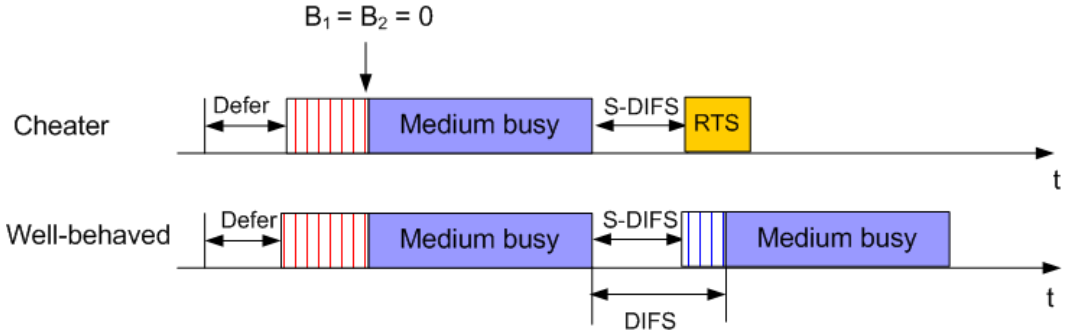
Many solutions have been proposed so far to deal with the greedy behavior in MAC layer. These solutions can be grouped into three main categories: backoff algorithm modification based schemes, monitoring based schemes and game theory based schemes. The first category aims to design a new MAC protocol which overcomes the vulnerabilities of the standard backoff algorithm to greedy behavior. On the other hand, the second one seeks to implement an additional component at the MAC layer to detect greedy nodes without modifying the standard backoff algorithm. The third category exploits the strength of game theory to develop more robust solutions.

3.1. Backoff algorithm modification based schemes

In order to handle the problem of MAC layer misbehavior, [15] proposes to modify the BEB (Binary Exponential Backoff) algorithm specified in IEEE 802.11 protocol [6], in order to



(a) Cheater node gains access to the medium ahead of schedule because it starts decreasing its backoff before the well-behaved nodes



(b) Cheater node accesses the medium early, since it waits a period lesser than the well-behaved nodes when the medium is sensed idle

Figure 2. The S-DIFS misbehavior consequences

facilitate the detection of cheater nodes. The main assumption of this approach is that the receiver node is trustworthy, and it is responsible for generating and assigning the backoff value to be used by the sender for its next transmission. The backoff value is sent in the CTS and ACK frames of the actual transmission. By comparing the observed value of backoff and the assigned one, the receiver is able to detect any misbehavior at the sender side. A penalty is added to the next backoff value whenever the sender deviates from the assigned one. If the total deviation of the sender throughout the last N observations exceeds a predefined threshold, the node is termed as misbehaving and the higher layers are informed accordingly. This solution is efficient however the following issues make it practically infeasible: the receiver may misbehave by assigning small values to some nodes (colluding nodes), and assigning large values to other nodes in order to decrease their throughput. Moreover, if the sender node generates TCP traffic with inter frame delay, the observed backoff value at the receiver side will appear greater than the assigned one, and then leave the misbehaved node undetected.

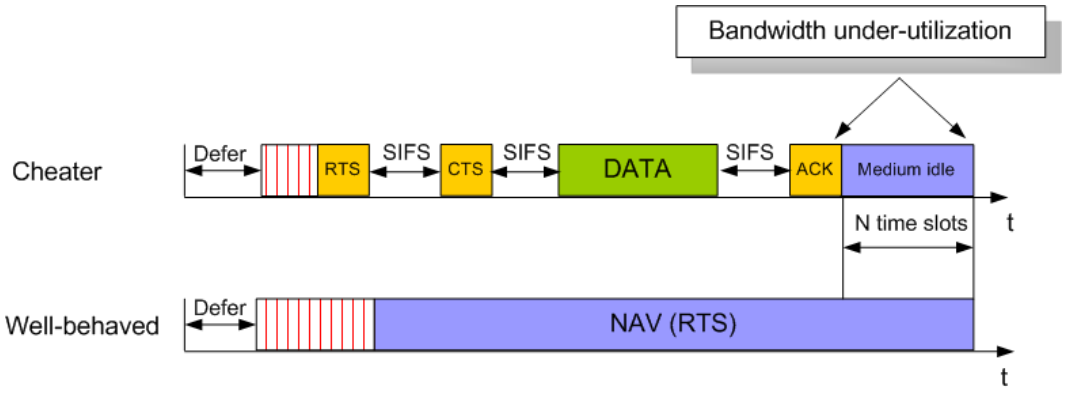


Figure 3. The NAV duration increase misbehavior consequence

In order to ensure the randomness of the backoff value, a new scheme is proposed in [12], in which at least one of the communicating nodes (sender or receiver) is assumed to be honest and a reputation system such as CONFIDANT [9] is deployed at the MAC layer. The main idea is to let both the sender and receiver agree on a random value of backoff through a public discussion using a commitment scheme inspired from the protocol applying flipping coins over the telephone, which was introduced in [2]. This scheme circumvents the misbehavior of one part of the communication, however it is still unable to detect the collusion among sender and receiver.

The called PRB (Predictable Random backoff) algorithm is proposed in [19] aiming to guarantee per frame monitoring to provide fast detection of misbehaving nodes. PRB is based on slight modification of the standard backoff algorithm by forcing each node to choose a backoff value from the interval $[CW_{lb}, CW]$ instead of $[0, CW]$, where CW_{lb} is calculated based on the previous backoff value and CW is a function of CW_{min} and the number of failed transmission. In this way, a receiver node can detect any deviation from the sender since the backoff value is predictable. This solution is faster than the previous ones however its main drawbacks are:

- The backoff value observed by the receiver may be different from the one generated by the sender due to hidden terminal phenomenon and interference. Hence per frame detection may increase the probability of misdiagnosis.
- Since in PRB each node selects its backoff from a smaller interval than in the standard backoff algorithm, the collision probability increases, leading to higher packet delay. Moreover, PRB leads to low channel utilization by the contending nodes.

3.2. Monitoring based schemes

The authors of [21] have presented a modular system, dubbed DOMINO, that does not require any modification to the standard MAC protocol. This system is implemented at the AP (access point) which is assumed to be trusted. It consists of a set of components ensuring complementary tasks. The first task is to monitor the behavior of wireless nodes around the AP for a certain period of time in order to collect traffic traces of each node. As a second

task, these traces are passed through a set of tests to measure the deviation of each node from the expected regular behavior. Each of these tests correspond to a specific misbehavior technique (e.g. backoff manipulation, S-DIFS misbehavior and scramble CTS frames). The output of these tests is analyzed by the decision component to infer whether a given node is well behaved or cheater. A node is considered as cheater if its corresponding deviation counter exceeds a predefined threshold for at least one test. The network administrator is then informed about the detected cheaters in order to punish them adequately.

DOMINO is a simple and efficient solution compatible with the existing infrastructure and can be seamlessly integrated with existing IEEE 802.11 MAC protocol security tools to provide ultimate protection. Nevertheless, DOMINO is still vulnerable to adaptive cheater problem described in section 2.3.

The sequential analysis concept introduced by Wald in [1] was widely used by researchers in order to detect security attacks in wireless networks. The work presented in [18] is based on this concept and provides an analytical model for the packets inter-arrival time distribution in saturated networks, representing an extension of Bianchi's stochastic model [7]. Based on this model the authors develop an algorithm to detect the cheating nodes by observing the throughput earned by the well-behaved nodes. These observations are further evaluated through a sequential probability ratio test to identify which node is a cheater. To work well, this scheme assumes the knowledge of the exact value of the greedy factor (i.e., the interval from which the cheater selects its backoff value), however this information is not available in practice. Therefore this scheme can not be implemented in wireless cards.

A framework aiming to cope with backoff rules violations was proposed in [20]. This framework designs a deterministic method allowing nodes to identify any neighbor node disobeying the backoff rules. This method requires that each node uses its MAC address as a seed of the pseudo random number generator used to generate its backoff values. This way, each node is aware of the sequence of backoff values to be used by all of its neighbors. Moreover, a modification to the RTS packet is made by including the following values: the pseudo random sequence of backoff values chosen by the node, its transmission attempts and a message digest of the data packet to be transmitted. Having this information, a node is able to monitor its neighbors and detect any misbehavior attempt. To circumvent the monitoring ambiguity issue and ensure the correct diagnostic, a statistical inference method is adopted wherein a series of tests [8] are applied.

A minimax [3] decision making formulation of the MAC layer misbehavior is proposed in [16]. The advantage of this approach is that it provides detection rules requiring a minimum number of observations to make the right decision regarding the behavior of a given neighbor. The first step in this scheme consists in identifying a class of attacks which leads to the maximum performance loss using sequential tests. Afterwards, a minimax robust sequential detection problem corresponding to the cheating one in the worst case is derived. Finally, the case of an attacker which delays the decision making, as long as possible, in order to prevent being detected is studied too.

3.3. Game theory based schemes

Game theory has been widely applied for investigating and assessing the selfish behavior impact in CSMA/CA, and numerous contributions have been proposed to cope up with. In [13], the authors adopt a dynamic game based scheme to derive the conditions which lead the

		Schemes							
		FLSAC	DOMINO	PRB	[16]	[15]	[20]	[18]	[13]
Features	Detection of adaptive cheater	Yes	No	No	No	No	No	No	No
	Applies the standard BEB	Yes	Yes	No	No	No	No	Yes	No
	Use the standard packet format (RTS/CTS/DATA)	Yes	Yes	Yes	Yes	No	No	Yes	Yes
	Keep the randomness property of BEB	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

Table I. The key difference between FLSAC and the existing schemes

set of cheaters to reach the Pareto optimal Nash equilibrium. Furthermore, they propose a detection scheme for non cooperative cheaters along with an adequate punishment scheme. A cheater node getting higher throughput than the rest of cheaters is deemed as a deviating cheater. Therefore, a selective jamming of its packets is carried out. The main drawback of this approach is that it considers all the participants in the game being cheaters. Hence the obtained throughput is significantly lower than the one achieved in the case where the cheaters are minority, which is the common case in most wireless networks, especially in wireless mesh networks.

As described in this section, most of the previous works either deal with one cheating strategy or detect a cheater applying only one strategy at a time. They generally fail to detect a more sophisticated cheater that combines several strategies together. To address this issue, we have designed a fuzzy logic based scheme, dubbed FLSAC, that better deal with such sophisticated behaviors. The key differences between our solution and the other approaches are exhibited in table I, from which we can see that FLSAC is the sole scheme able to detect misbehaving adaptive cheaters. Furthermore, FLSAC still use the same standard BEB algorithm, preserving its randomness property in backoff values selection, and doesn't introduce any extra field to any of the control or data frames. We can also notice that the schemes proposed in [16], [15], [19], [20] and [18] consider that a cheater node behaves similarly in an Ad Hoc network or within a Wireless LAN, and develop their solutions according to this assumption. We believe that this assumption is not sustainable and that a cheater node behaving greedily in an ad hoc network leads to a collapse of its own traffic performance unless its destination is one of its 1-hop neighbors. Therefore, it is necessary to have a careful characterization of this greedy misbehavior in the context of an Ad Hoc network before suggesting solutions.

4. THE PROPOSED FLSAC SCHEME

The operations of our solution are described in Figure 4. First, the mesh router/gateway monitors the wireless mesh clients connected to it for a period of time. Second, the collected samples of observations about these nodes' behaviors are passed to DOMINO component. If DOMINO classifies a given node as greedy then it will be penalized by the punishment component. Otherwise, FLSAC checks further whether this node is applying adaptive cheating strategy to escape from DOMINO. If so, it gets punished.

4.1. 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 [5] 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.

4.2. 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 [4].

In the following, we describe how FLSAC works to detect the adaptive cheater. The role of each component shown in Figure 5 is defined as follows:

4.2.1. ***Inputs*** We have designed our fuzzy system to support four inputs where three of them represent the traces collected by the MR for each wireless node. These traces represent the possible misbehavior techniques discussed in section 2.3. The output of each monitoring period is used as the fourth input of the system in the next period. The purpose of this is to make the final decision more accurate.

BDEV (Backoff DEViation) In DOMINO, at the end of each monitoring period T the MR computes the average of the observed backoff values of a station S_i, and then compares it to its own average backoff B_{MR} 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 backoff value (B_i) and for m (m ≥ 1) times a large backoff value (B_j) such that :

$$\frac{\sum_{i=1}^{N-m} B_i + \sum_{j=(N-m)+1}^{N} B_j}{N} \geq \alpha \times B_{MR} \quad (1)$$

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

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

$$BDEV = \frac{\frac{\sum_{i=1}^{N}(B_{MR}-B_i)}{N}}{B_{MR}} \tag{2}$$

where B_i is the i^{th} observed backoff value of a node. If (B_{MR} - B_i) < 0 then this difference is considered as 0.

Example :

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

$$\frac{(4+5+18+30+6)}{7} = 9 > 8$$

whereas our scheme calculates the following deviation percentage with respect to the Equation. 2;

$$BDEV = \frac{(\frac{4+3+8+0+8+0+2}{7})}{8} = 44\%$$

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

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

$$RTR = \frac{num - rtx(S_i)}{AVG_{i \neq j}[num - rtx(Sj)]} \tag{3}$$

where $num-rtx(S_i)$ is the retransmission number of the node i and $AVG_{i \neq j}[num-rtx(Sj)]$ is the average number of the other nodes' retransmission attempts.

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:

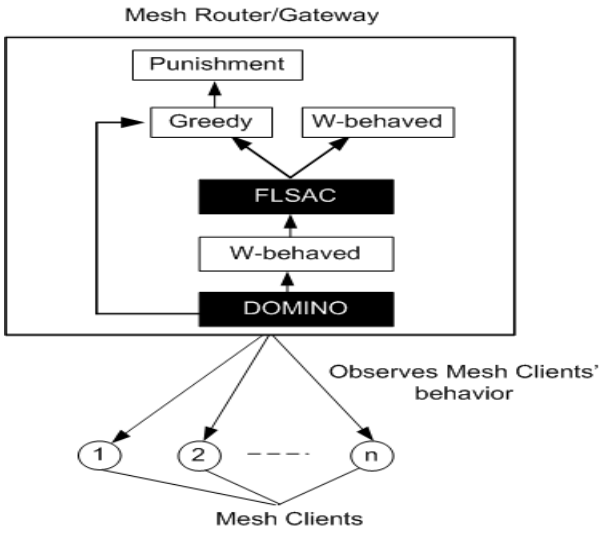


Figure 4. The overall functioning of the proposed solution

- 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.

4.2.2. ***Fuzzification*** This step consists in replacing the input values by the 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:

$$\beta_F : U \rightarrow [0, 1] \tag{4}$$

such that for each $u \in U$, 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 [5].

4.2.3. ***Rule-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.

†In 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.

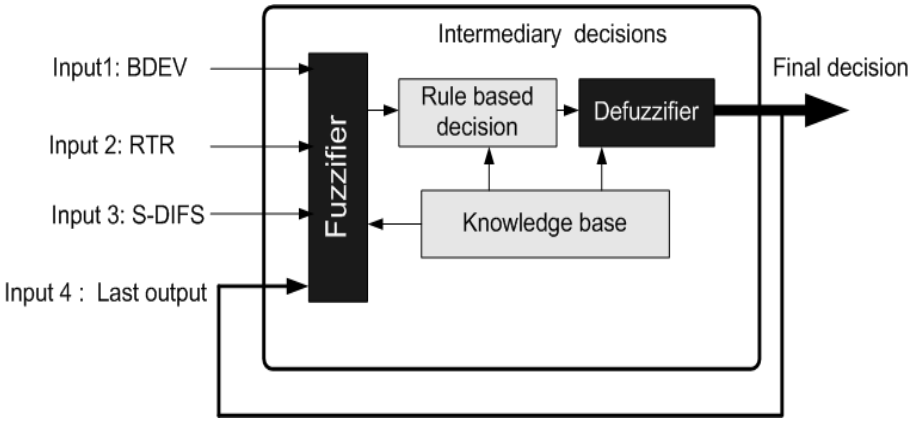


Figure 5. The main components of FLSAC

The degree of truth for a predicate in the form "x is F" is given by $\beta_P = \beta_F(x)$. The traditional logic operators such as $\wedge$ (AND) and $\vee$ (OR) are redefined in order to produce the truth value of the final statement as follows:

$$\begin{aligned} \beta_{P_1} \wedge \beta_{P_2} &\equiv min(\beta_{P_1}, \beta_{P_2}) \\ \beta_{P_1} \vee \beta_{P_2} &\equiv max(\beta_{P_1}, \beta_{P_2}) \end{aligned} \tag{5}$$

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 **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 II and III while the rules shown in table IV are constructed based on the last monitoring period decision combined with the output of the table III.

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.

	BDEV:	L	M	H
RTR:				
L		LS	HS	C
M		N	LS	HS
H		N	N	HS

Table II. Fuzzy rules of the formula : RES_1 = (BDEV $\wedge$ RTR)

	RES_1 :	N	LS	HS	C
S-DIFS:					
L		N	LS	HS	C
M		N	LS	HS	C
H		HS	HS	C	C

Table III. Fuzzy rules of the formula : RES_2 = (BDEV $\wedge$ RTR) $\wedge$ S-DIFS)

	RES_2:	N	LS	HS	C
Last decision:					
N		N	LS	HS	C
LS		N	LS	HS	C
HS		N	LS	C	C

Table IV. The final fuzzy decision of FLSAC : FDEC = (RES_2 $\wedge$ Last decision)

4.2.4. ***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 the center of gravity (CoG), in which a crisp output ($Output_{crisp}$) is chosen using the center of area of each fuzzy set. The CoG output value is given by the following formula:

$$\begin{aligned} Output_{crisp} &= \frac{\sum_{i=1}^{4} d_i \beta_i}{\sum_{i=1}^{4} \beta_i} \\ &= \frac{d_N \beta_N + d_{LS} \beta_{LS} + d_{HS} \beta_{HS} + d_C \beta_C}{\beta_N + \beta_{LS} + \beta_{HS} + \beta_C} \end{aligned} \tag{6}$$

where d_i is the membership function's center of area corresponding to each class i of node behavior, and β_i is the membership level of a node behavior to the class i.

The detailed operations of FLSAC are summarized by the Algorithm 1.

Algorithm 1 FLSAC

if *((*$BDEV_i \geq thr_1$ *) ∨ (* $RTR_i \geq thr_2$ *) ∨ (* $S-DIFS_i \geq thr_3$ *))* **then**
  the node i is declared as cheater;
else
  if *(*$FLSAC_{dev}$ *(* $BDEV_i$, RTR_i, $S-DIFS_i$*)* $\equiv C$*)* **then**
    the node i is declared as cheater;
  else
    if *(*$FLSAC_{dev_j}$ *(* $BDEV_i$, RTR_i, $S-DIFS_i$*)* $\equiv HS$ *)* **then**
      if *(*$FLSAC_{dev_{j-1}}$ *(* $BDEV_i$, RTR_i, $S-DIFS_i$*)* $\equiv HS$*)* **then**
        the node i is declared as cheater;
      end
    end
  end
  /* $FLSAC_{dev_j}$ refers to the decision of the actual monitoring period and $FLSAC_{dev_{j-1}}$ refers to the decision of the previous monitoring period thr_1, thr_2 and thr_3 are the misbehaving thresholds set by FLSAC for the cheating strategies BDEV, RTR and S-DIFS, respectively. */
end

Algorithm 2 Punishment scheme

if *(RTS received)* **then**
  if *(@source* $\notin$ *CL)* **then**
    schedule CTS packet transmission ; /* CL: a list of detected cheater nodes */
  else
    if *(++CPT* $> m$*)* **then**
      schedule CTS packet transmission; /* CPT: is a counter initially set to zero */
      $CPT = CPT \ mod \ N$;
    end
  end
end
if *(DATA packet received)* **then**
  if *(@dest* $\in$ *CL)* **then**
    $d = RAND$ [X_1, X_2];
    Delay the packet delivery toward @dest node by d μs whenever it is scheduled for transmission ;
  end
  /* X_1 and X_2 are integers used to determine the interval from which the MR chooses the amount of delay (in μs) to be applied for the cheater node's packets. */
end

4.3. Punishment scheme and additional issues

In this section we address the reaction of a mesh router running FLSAC after the detection of a greedy node. As a punishment scheme, the MR can deny the response to an RTS sent by the greedy node (For example, doesn't answer to m RTS from the N received ones; $m < N$).

Simulation parameters	Parameter value
Area	1500m × 1000m
# Wireless mesh clients	8 (light load) and 29 (heavy load)
MAC protocol	IEEE 802.11b
Transmission range	250 m
# Greedy nodes	1, 3, 7 and 10 (4 scenarios)
FLSAC	running on MR (Mesh Router)
Switching scheme	Random (see the flowchart in Figure 8)
Traffic type	CBR
Data rate	11mbps
CBR packets size	512 bytes
Monitoring period	10s
# Simulation epochs	20

Table V. Simulation settings

Moreover, the MR can also delay the delivery of packets intended to the greedy node. This reaction will deprive the greedy node from gaining any benefits and decrease the performance of its applications. As a consequence, it forces the greedy node to behave correctly, at least periodically. This way the greedy node can survive longer but with less damage to the network. The reaction of MR node when a cheater node is detected is described in the Algorithm 2. The reason for which the MR reacts in such manner is to try to motivate or force the cheater node to behave correctly since whenever it misbehaves its upload and download throughput is decreased. By upload/download throughput we mean the throughput of the traffic for which the cheater is source/destination respectively.

Another important issue is how to detect a greedy node which does not double its CW after collision? One possible solution is described as follows: If consecutive transmission requests are observed from a suspected node, then the MR denies the response to the last received RTS, forcing the node to double its CW if it is a well-behaved node [6]. If the estimated backoff for its subsequent transmission is smaller than the consecutive backoff (for several times), this node is deemed as greedy and the above punishment scheme is launched.

5. SIMULATION RESULTS

This section reports the simulation results of FLSAC by using the OPNET 14.0 network simulator [23]. The simulation scenarios and settings are summarized in table V.

5.1. Simulation environment

For the simulation environment we consider a WMN similar to the topology shown in Figure 1, in which MR_1 provides connection to 8 wireless clients which are within transmission range of each other. The wireless clients (including the cheater) are sources of CBR traffic

(512 bytes/packet and 200 packets/s). In our simulation we implement the three misbehavior techniques discussed in the previous sections along with the adaptive cheating misbehavior. The results are averaged over 20 simulation times, with 120 seconds each. To outline the impact of the different MAC layer misbehavior strategies, we configure the simulation as follows: the cheater node first launches each cheating strategy individually and then carries out an adaptive attack by switching dynamically between the different strategies. Next, we compare the throughput of the cheater node with the mean [‡] of well behaving nodes' bandwidth fair share. Afterwards, in order to evaluate the efficiency of FLSAC we run it on the MR and measure its efficiency in terms of detection ratio and speed.

5.2. Discussion of simulation results

In the sequel, we present the obtained results along with their analysis and explanation.

The backoff manipulation technique (BDEV) allows the cheater node to gain a considerable bandwidth. The throughput of the cheater increases with the increase of the misbehavior coefficient (Mc). The maximum load (200 packets) is achieved when the misbehavior coefficient is 0.6 as shown in Figure 6(a). Meanwhile, a decrease in throughput of well-behaved nodes is observed from 63 packets/sec (Mc = 0) to 46 packets/sec (Mc = 0.6). From this point onwards (Mc =0.6), the cheater is able to capture the medium completely and transmit all its buffered packets since it chooses a very small backoff values from the interval $[0, (1 - Mc) \times CWmin]$.

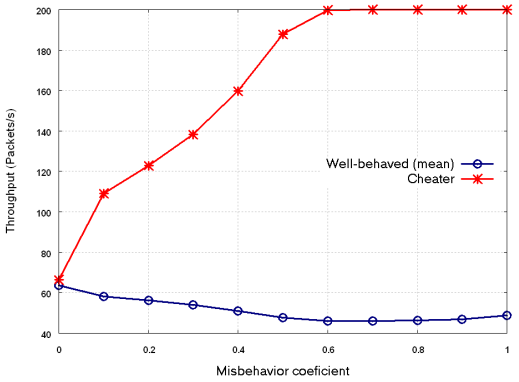
The Figure 6(b) depicts the impact of the DIFS value reduction on the fairness property. Here, the S-DIFS value varies from the lower value of {DIFS - 2 × slot time}(equal to SIFS) to the higher limit of DIFS standard value of IEEE 802.11b. The cheater's throughput increases with the increase of Mc value, until it reaches the maximum throughput of 142 packets/sec, and then decreases to 123 packets/sec when S-DIFS reduces to SIFS value. This is mainly due to the fact that the cheater may experience collisions with nodes sending a short frames (CTS or ACK). The throughput of well-behaved nodes decreases as well, till it reaches a minimum value of 55 packets/sec. Hence, it is clear that the S-DIFS technique has less impact on bandwidth share than the BDEV.

When the cheater scrambles the CTS packets destined to well-behaved nodes, the throughput of these nodes decreases sharply until it drops down to 0 packets/sec as depicted in Figure 6(c). On the other hand, the bandwidth share of the cheater increases with the increase in the proportion of the scrambled packets. In addition, The cheater is able to transmit all its buffered packets only by scrambling half of the CTS packets, while the well-behaved nodes transmit only 23 packets/sec. Therefore, the cheater can save more energy by scrambling half of the CTS packets rather than the whole packets since it still gains sufficient bandwidth for its own traffic.

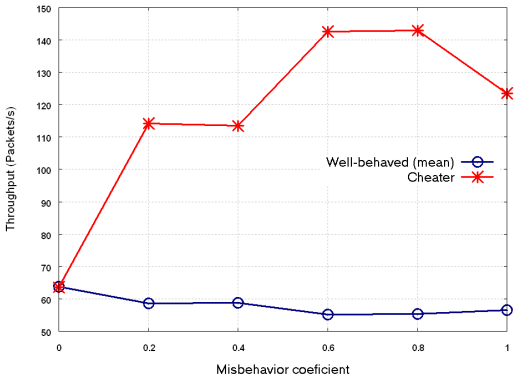
As we can notice from these results, the scrambling technique is more damaging than BDEV and S-DIFS ones and its negative impact can lead to a break down of other nodes throughput.

When the cheater increases the duration of its transmission, the neighboring nodes set their NAV durations to values larger than the time required to complete the current transmission. Therefore the medium will be free for a period of time during which no one among the neighbor

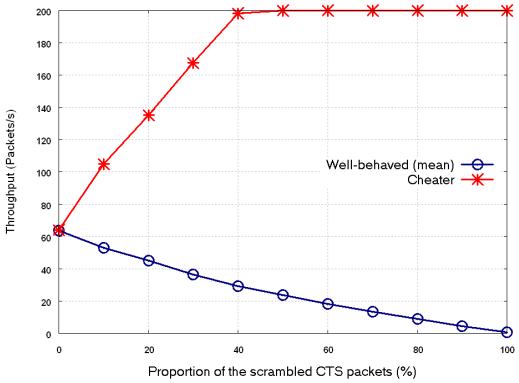
[‡]We use the mean of throughput of the other 7 nodes as there is a common bandwidth fair share for each of them.



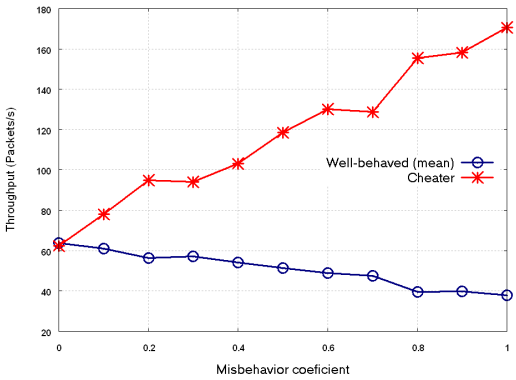
(a) Impact of backoff manipulation on Throughput



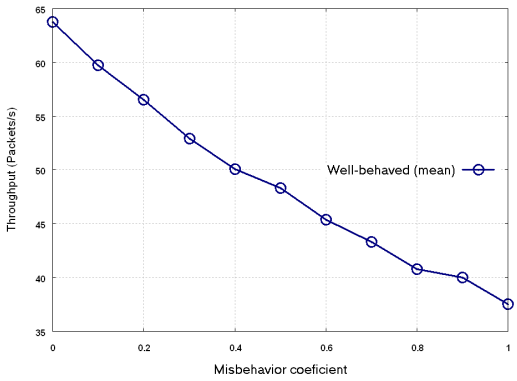
(b) Impact of DIFS value reduction on Throughput



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



(d) Impact of the adaptive cheater on Throughput



(e) Impact of the NAV increase misbehavior on Throughput

Figure 6. Impact of MAC layer misbehavior on throughput in WMNs

nodes can transmit because their NAV value is already set. Moreover, the cheater can earn an extra bandwidth if its buffer is always full since it accesses the medium with less contention. If the cheater has a less traffic load then a spatial reuse reduction is resulted. As depicted in the Figure 6(e), the throughput of the well-behaved nodes decreases when a cheater node launches the attack and it is proportional to the value of misbehavior coefficient.

To implement the adaptive cheater behavior, we generate 3 random numbers within the interval $[0, thresh_i \times Mc]$ 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 (see the flowchart depicted in Figure 8 for more details).

As depicted in the Figure 6(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.

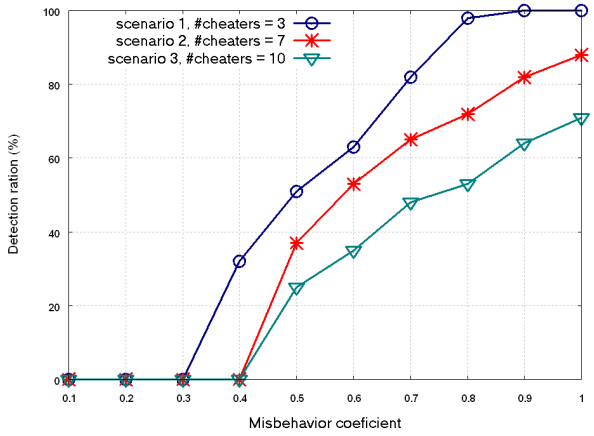
In the sequel, we evaluate the efficiency of our scheme in terms of detection ratio and speed. To do so, we increase the number of wireless clients to 29 and the number of cheaters to 10. Moreover, we generate three scenarios through which we vary the number of cheaters to 3, 7 and 10 in order to figure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

The Figure 7(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 6(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 for the MR running FLSAC to collect enough samples to decide about the nodes' behavior.

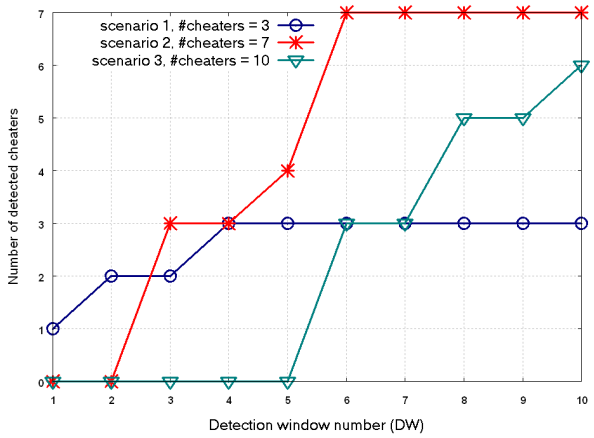
To observe the response latency (detection speed) of FLSAC, we set all the cheaters start misbehaving simultaneously, and we record the detection window number in which the cheater was detected. The Mc as well is set to the highest value. As shown in Figure 7(b), the cheater nodes in the scenario 1 were detected earlier than the other cheaters 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 MR 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.

6. CONCLUSION

This paper proposes a new scheme to cope with the adaptive greedy behaviors in wireless mesh networks. Our scheme can be regarded as an extension to DOMINO, aiming to detect greedy



(a) Detection ratio versus misbehavior coefficient



(b) Detection speed of FLSAC

Figure 7. FLSAC's performance

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.

APPENDIX

The scheme adopted by the cheater for switching among the three strategies (Scrambling CTS frames, BDEV and S-DIFS) is described in the flowchart shown in Figure 8. The abbreviations used in this flowchart are explained as follows:

- X_{RTR}, X_{BDEV} and X_{S-DIFS} are the number of times the cheater node can disobey the protocol by applying the scrambling CTS frames, $BDEV$ and $S-DIFS$ strategies respectively during one monitoring period.
- $Mis-Str$ is the randomly chosen strategy for transmitting the ongoing packet.
- Mc is the misbehavior coefficient and $thresh_i$ is the estimated threshold configured by the MR for the misbehaving strategy i.

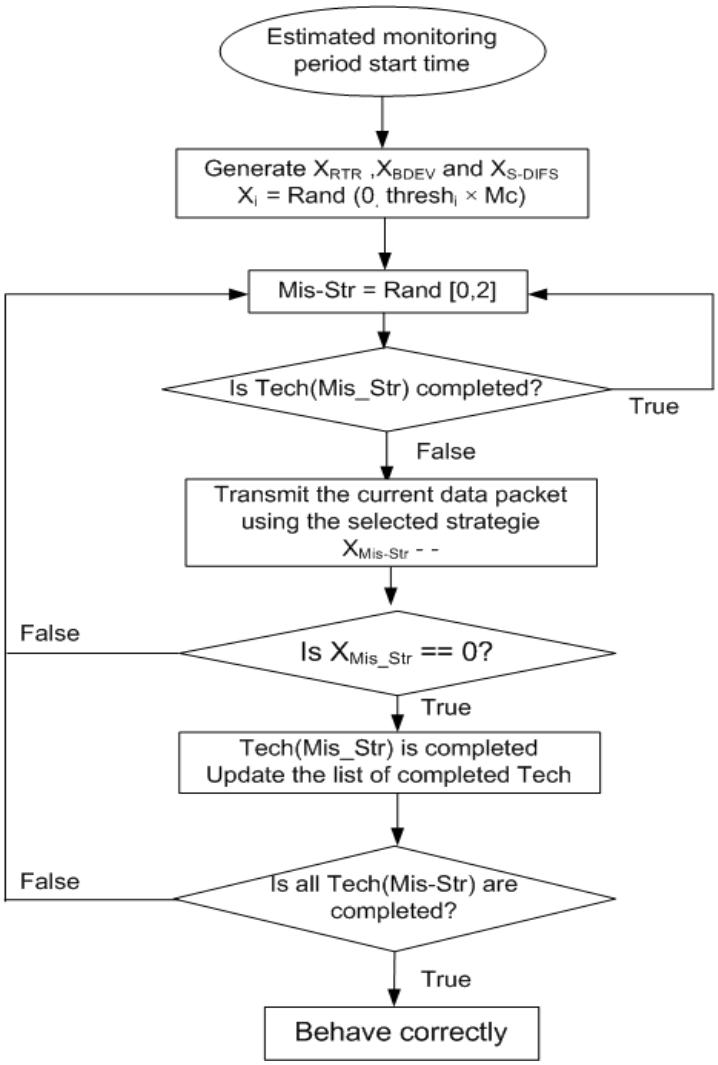


Figure 8. The switching scheme used by the adaptive cheater to switch over the cheating strategies

REFERENCES

1. Wald A. *Sequential Analysis*. John Wiley & Sons, New York, 1947.
2. Blum M. Coin flipping by telephone: a protocol for solving impossible problems. In *Proceedings of the* 24*th IEEE Spring Computer Conference, (COMPCON)*,San Francisco, California, USA, February 1982.
3. Verdu S, Poor HV. On minimax robustness: a general approach and applications. *IEEE transactions on Information Theory*, Vol. 30, No. 2, March 1984.
4. Klir G, Yuan B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall PTR, Upper Saddle River, NJ, 1995.
5. Passino KM, Yurkovich S. *Fuzzy Control*. Addison-Wesley, 1998.
6. IEEE 802.11 wireless LAN media access control (MAC) and physical layer (PHY) specifications, 1999.

7. Bianchi G. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 3, March 2000.
8. Rohatgi V, Saleh A. *An Introduction to Probability and Statistics*. Wiley-interscience, 2001.
9. Buchegger S,Le Boudec JY. Performance Analysis of the CONFIDANT Protocol. In *Proceedings of the* 3*rd ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC'02*), Lausanne, Switzerland, June 2002.
10. Gupta V, Krishnamurthy S, Faloutsos M. Denial of service attacks at the MAC layer in wireless ad hoc networks. In *Proceedings of Military Communication Conference MILCOM 02*, Anaheim, CA, October 2002.
11. Bellardo J, Savage S. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the* 12^{th} *USENIX Security Symposium*, Washington, DC, USA, August 2003.
12. Cardenas AA, Radosavac S, Baras JS. Detection and Prevention of MAC layer Misbehavior in Ad Hoc Networks. In *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, Washington DC, USA, October 25, 2004.
13. Cagalj M, Ganeriwal S, Aad I, Hubaux JP. On selfish behavior in CSMA/CA networks. In *Proceedings of the IEEE INFOCOM05*, Miami, USA, March 2005.
14. Akyildiz IF, Wang X. A Survey on Wireless Mesh Networks. *IEEE Communications Magazine*, Vol. 43, No. 9, September 2005.
15. Kyasanur P, Vaidya NH. Selfish MAC Layer Misbehavior in Wireless Networks. *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, September/October 2005.
16. Radosavac S, Baras J, Koutsopoulos I. A Framework for MAC Protocol Misbehavior Detection in Wireless Networks. In *Proceedings of* 4^{th} *ACM Workshop on Wireless Security*, Cologne, Germany, September 2005.
17. Boukerche A. *Handbook of Algorithms for Wireless Networking and Mobile Computing*. CRC Chapman Hall, 2005.
18. Rong Y, Lee SK, Choi HA. Detecting Stations Cheating on backoff Rules in 802.11 Networks Using Sequential Analysis. In *Proceedings of IEEE INFOCOM*,Barcelona, Spain, April 2006.
19. Guang L, Assi C. Mitigating Smart Selfish MAC Layer Misbehavior in Ad Hoc Networks. In *Proceedings of the* 2^{nd} *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 06)*, Montreal, Canada, June 19-21, 2006.
20. Lolla VN, Krishnamurthy LKL, Ravishankar SV, Manjunath C. Detecting MAC Layer backoff Timer Violations in Mobile Ad Hoc Networks. In *Proceedings of the* 26^{th} *IEEE International Conference on Distributed Computing Systems, ICDCS 2006*, Lisboa, Portugal, July 2006.
21. Raya M, Aad I, Hubaux JP, El Fawal A. DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots. *IEEE Transactions on Mobile Computing*, Vol. 5, No. 12, December 2006.
22. Radosavac S, Baras JS. Application of Sequential Detection Schemes for Obtaining Performance Bounds of Greedy Users in the IEEE 802.11 MAC. *IEEE Communications Magazine 2008*; 46(2):148-154.
23. OPNET Technologies, "OPNET Modeler", http://www.opnet.com/.

AUTHOR'S BIOGRAPHY

Soufiene Djahel is pursuing his PhD at the University of Sciences and Technologies of Lille, within the MISC research group, since October 2007. Prior to that, he was a researcher engineer at INRIA Lille Nord Europe research center from July to December 2007. He has obtained his degree of engineering in computer science and a magister degree with majors in networking and distributed systems from University Badji Mokhtar of Annaba and University Abderrahmane Mira of Bejaia in 06/2004 and 02/2007, respectively. The research interests of Soufiene Djahel lie in the field of security in mobile ad hoc networks, MAC layer misbehavior issues, and security in vehicular networks. He is reviewer of many conferences and journals in the networking and security field. Soufiene Djahel is an IEEE student member.

Farid Naït-Abdesselam (farid.nait-abdesselam@lifl.fr) obtained his engineering degree in computer science from University of Sciences and Technologies Houari Boumediene (USTHB) Algiers, Algeria, in 06/1993 and a master degree in computer science from University of Paris Descartes - France, in 09/1994. After two years spent in the industry working as a software engineer, he joined the University of Versailles Saint Quentin, (UVSQ) France in 01/1996, and got his PhD degree in computer science in 01/2000. During the year of 1998, he worked as an associate researcher at University of Western Ontario, London Ontario Canada, on distributed interactive virtual environment and multimedia communications over ATM networks.
From 09/1999 to 08/2000 he was an assistant professor at University of Sciences and Technologies of Lille - France. From 09/2000 to 08/2003 he worked as an associate professor at INSA of Lyon and a research member of INRIA Rhne Alpes. Since 09/2003 he is an associate professor at University of Sciences and Technologies of Lille and till 09/2007 a research member of the INRIA Lille Nord Europe. His research interests lie in the field of computer and communication networks with emphasis on architectures and protocols for quality of service and security in IP based networks, mobile ad-hoc, sensor, vehicular, and mesh networks, and overlay networks.
Farid Naït-Abdesselam has been on the technical program committee of different IEEE and ACM conferences, including GLOBECOM, ICC, LCN, and MSWiM, and regularly invited to chair some of their sessions. He is chairing/has chaired the IEEE International Workshop on Wireless Local Networks, the IEEE/ACS International Workshop on Internet Services, and the International Workshop on Peer to Peer Networking. He is currently serving as Editorial Liaison chair of the IEEE LCN Conference, and publicity co-chair of many conferences. Farid Nat-Abdesselam is a member of the IEEE, IEEE Communications Society, and IEEE Computer Society.