Université des Sciences et Technologies de Lille

Thèse

Présentée par
Soufiene DJAHEL

Sécurité des Protocoles de Routage et d'Accès au Médium dans les Réseaux Sans Fil Multi-Sauts

Jury

Président :	Prof. Jean-Luc Dekeyser	-	Université de Lille 1
Rapporteurs :	Prof. Abdelmadjid Bouabdallah	-	Université de Technologie de Compiègne
	Prof. Pascal Urien	-	Telecom Paris Tech
Examinateurs :	Prof. Ashfaq A. Khokhar	-	University of Illinois Chicago
	Dr. Mohammed Achemlal	-	France Telecom R&D
Directeur :	Prof. Farid Naït-Abdesselam	-	Université de Paris Descartes

Décembre 2010

University of Sciences and Technologies of Lille

PhD Thesis

by
Soufiene DJAHEL

Secure Routing and Medium Access Protocols in Wireless Multi-hop Networks

Committee

President :	Prof. Jean-Luc Dekeyser	-	University of Lille 1
Reviewers :	Prof. Abdelmadjid Bouabdallah	-	University of Technology of Compiègne
	Prof. Pascal Urien	-	Telecom Paris Tech
Examiners :	Prof. Ashfaq A. Khokhar	-	University of Illinois Chicago
	Dr. Mohammed Achemlal	-	France Telecom R&D
Advisor :	Prof. Farid Naït-Abdesselam	-	University of Paris Descartes

December 2010

Declaration of Authorship

I, Soufiene Djahel, declare that this document titled, 'Securing Communications in Wireless Multi-hop Networks: An in Depth Study of Routing and MAC Layers Misbehaviors' and the work presented in it are my own. I confirm that:

- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, the work presented in this document is entirely my own work.
- I have acknowledged all main sources of help.

Signed: _____

Date: _____

Abstract

While the rapid proliferation of mobile devices along with the tremendous growth of various applications using wireless multi-hop networks have significantly facilitate our human life, securing and ensuring high quality services of these networks are still a primary concern. In particular, anomalous protocol operation in wireless multi-hop networks has recently received considerable attention in the research community. These relevant security issues are fundamentally different from those of wireline networks due to the special characteristics of wireless multi-hop networks, such as the limited energy resources and the lack of centralized control. These issues are extremely hard to cope with due to the absence of trust relationships between the nodes.

To enhance security in wireless multi-hop networks, this dissertation addresses both MAC and routing layers misbehaviors issues, with main focuses on thwarting black hole attack in proactive routing protocols like OLSR, and greedy behavior in IEEE 802.11 MAC protocol. Our contributions are briefly summarized as follows.

As for black hole attack, we analyze two types of attack scenarios: one is launched at routing layer, and the other is cross layer. We then provide comprehensive analysis on the consequences of this attack and propose effective countermeasures.

As for MAC layer misbehavior, we particularly study the adaptive greedy behavior in the context of Wireless Mesh Networks (WMNs) and propose FLSAC (Fuzzy Logic based scheme to Struggle against Adaptive Cheaters) to cope with it. A new characterization of the greedy behavior in Mobile Ad Hoc Networks (MANETs) is also introduced. Finally, we design a new backoff scheme to quickly detect the greedy nodes that do not comply with IEEE 802.11 MAC protocol, together with a reaction scheme that encourages the greedy nodes to become honest rather than punishing them.

Keywords: Wireless Multi-hop Networks, MANETs, WMNs, Black hole attack, MAC layer misbehavior, Greedy behavior, Routing protocols security, OLSR

Résumé

Récemment, les comportements malveillants dans les réseaux sans fil multi-sauts ont attiré l'attention de la communauté scientifique. La prolifération rapide du nombre de dispositifs sans fil ainsi que la diversification des applications basées sur ces réseaux ont grandement contribué à l'amélioration de la qualité de vie ainsi que la modernisation de la société. Cependant, la nécessité de sécuriser ces réseaux et de garantir la robustesse de leurs services est devenue une préoccupation majeure. En effet, les caractéristiques spécifiques de ces réseaux, telles que l'absence d'infrastructure et l'absence d'une entité centrale de confiance, font que les réponses à leurs problèmes de sécurité sont tout à fait différentes de celles des réseaux filaires. De plus, le manque de confiance entre les nœuds rend ces problèmes encore plus critiques.

L'objectif de cette thèse vise à contribuer au renforcement de la sécurité dans les réseaux sans fil multi-sauts. Elle se focalise sur l'étude des comportements malveillants au niveau des couches MAC et réseau. Nous nous intéressons au développement de nouvelles solutions pour faire face à l'attaque du trou noir "Black hole" dans le contexte du protocole OLSR, ainsi qu'analyser le comportement des nœuds cupides "Greedy" au niveau de la couche MAC, dans toutes ses versions.

Une attaque de trou noir peut être menée suivant deux scénarios. Le premier scénario consiste à lancer l'attaque, exclusivement, au niveau de la couche réseau. Le second scénario consiste en une attaque multi-couches. Dans le cadre de cette thèse, nous analysons l'impact de ces deux types d'attaques et proposons des contre-mesures appropriées.

Au niveau de la couche MAC, nous étudions particulièrement le comportement cupide adaptatif dans le cadre des réseaux sans fil maillés et nous proposons une solution originale baptisée, FLSAC, afin de prévenir ce type de menace. Dans le cadre des réseaux mobiles ad hoc (MANETs), nous définissons un nouveau modèle de comportement des nœuds cupides. Nous développons aussi un nouvel algorithme de backoff, dont l'avantage principal est d'assurer une détection rapide des nœuds cupides non conformes aux spécifications du protocole IEEE802.11. Cet algorithme offre un mécanisme de réaction qui incite un nœud cupide à se comporter correctement en lui donnant la chance de se repentir après détection.

Mots clés: Réseaux sans fil multi-sauts, MANETs, WMNs, Attaque de trou noir, Comportement cupide adaptatif, Sécurité des protocoles de routage, OLSR

Acknowledgements

First of all, I would like to express my sincere gratitude and appreciation to my advisor Professor Farid Naït-Abdesselam for his valuable guidance throughout my research work. I am inspired by his insight, and I have learned a lot from him. I would like to thank him for his encouragement and support throughout my Ph.D study.

I would also like to thank, Prof. Abdelmadjid Bouabdallah, Prof. Pascal Urien, Prof. Ashfaq Khokhar, Prof Jean-Luc Dekeyser and Dr. Mohammed Achemlal for agreeing to serve on my committee. I particularly thank them for their valuable comments and remarks that will help to improve my future work.

I am in particular indebted to Dr. Zonghua Zhang for his encouragement and insightful comments and advices during my PhD studies. Many thanks to my current and former colleagues, namely Yassine, Dalil and Intessab, for their help in reading earlier versions of my papers and their valuable comments. I would also like to thank Dr. Youcef Begriche for his collaboration in modeling the problem of greedy behavior in Wireless Mesh Networks.

Finally, I would like to thank with all my heart my family for their infinite love and support throughout my life. Their love gave me strength to go on during the most difficult moments of my life and the least I can do is to dedicate this thesis to them.

Dedication

To my parents, for their unconditional love and support throughout my whole life.
To my uncle *Chemam Mokhtar*, allah yarhamou, who passed away in last september.

Contents

List of Figures

List of Tables

Acronyms

ACK	ACKnowledgment
AODV	Ad hoc On Demand Distance Vector
BDEV	Backoff DEViation
BEB	Binary Exponential Backoff
CBR	Constant Bit Rate
CRC	Cyclic Redundancy Code
CTS	Clear To Send
CW	Contention Window
DCF	Distributed Coordination Function
DIFS	DCF Inter Frame Space
DoS	Denial of Service
MAC	Medium Access Control
MANETs	Mobile Ad hoc NETworks
MC	Misbehavior Coefficient
MCL	Mesh CLient
MPR	Multi-Point Relay
MR	Mesh Router
NAV	Network Allocation Vector
OLSR	Optimized Link State Routing
QoS	Quality of Service
RREP	Route REPly
RREQ	Route REQuest
RTR	Retransmission Rate
RTS	Request To Send
S-DIFS	Short DCF Inter Frame Space
SIFS	Short Inter Frame Space
SRL	Short Retry Limit
TC	Topology Control
VANETs	Vehicular Ad hoc NETworks
WLAN	Wireless Local Area Network
WMNs	Wireless Mesh Networks

Chapter 1

Introduction

1.1 Background and motivations

At the turn of the twenty first century, we have all witnessed a steadily growth of deployment of wireless and mobile communications networks. Initially, this new paradigm of communication was aiming to provide communication services in some situations where infrastructures are no longer available, such as in battlefield, emergency and rescue operations. Then, due to the advances in mobile computing devices' technology (e.g., laptops, handheld digital devices, PDAs and wearable computers) and their increasing use in our daily life, the applications that use such networks have shown a tremendous growth. Therefore, wireless networks deployment in cities, inside building and to interconnect cars has become a prerequisite. To this end, several network architectures have been proposed and standardized such as, WLAN, MANETs, Wireless Mesh Networks (WMNs) and VANETs. To satisfy the users' requirements, these architectures need to provide a robust and secure service. Robust service means ensuring a high performance for the running applications by offering a Quality of Service (QoS) guarantees. In another hand, network security revolves around the three key principles of confidentiality, integrity and availability of service.

Security in wireless ad hoc networks is a vital element for basic network functions like packets forwarding and routing. Network operation can be easily jeopardized if effective countermeasures are not embedded into basic network functions at the beginning of their design. Unlike networks using dedicated nodes to support their critical functions, in ad hoc environment those functions are carried out by every node in the network. This difference constitutes the main cause of the security problems that are specific to such environment. As opposed to the dedicated nodes in traditional networks, the nodes in an ad hoc network cannot be trusted to ensure correct execution of critical network functions.

When tamper-proof hardware and strong authentication infrastructure are not available, for example, in an open environment where a common authority that regulates the network does not exist, any node of an ad hoc network can endanger network operations. To ensure reliable functioning of the network, each node must correctly execute critical network functions, as well as perform fair share of its resources. This means that it devotes the same efforts and resources for processing and transmitting its own packets as well as for those of its neighbors. This latter requirement seems to be a strong limitation for wireless mobile nodes where power saving is a major concern. The threats considered in MANETs scenario are, therefore, not limited to maliciousness; a new type of misbehavior called selfishness should also be taken into account. The maliciousness and selfishness behaviors may target both MAC and routing layers leading to sharp collapse of network performance. These misbehaviors are

logic consequence of the specific characteristics of wireless ad hoc networks. The stringent energetic resources push the nodes to abstain from relaying neighbors' packets to save their energy and thus extend their lifetime. Likewise, these nodes tend also to decline the proper use of MAC protocol rules to increase their bandwidth, because the availability of this resource is limited. Furthermore, these vulnerabilities in different layers highlight the growing need for conducting an in-depth investigation on routing and MAC protocols to identify the potential source of threats in their functioning and design, which can be exploited by a foe to compromise their security. Therefore, the design of effective defense mechanisms remains a compulsory task for preventing/detecting these misbehaviors or, at least, alleviate their impact.

The above discussed misbehaviors motivate us to pursue the research work presented in this dissertation. The problem we want to solve is the following. How can we detect misbehaving nodes that refuse to forward the routing packets, even though they launch the attack intelligently (i.e. through cross layer collusion)? How can we prevent greedy nodes from misusing the MAC protocol rules? and how can we detect them if they overcome the prevention scheme?. The answers to these questions will be provided throughout the subsequent chapters.

1.2 Dissertation organization

In order to make our dissertation easier to read and understand, we organize it in the following way.
This dissertation is mainly composed of two parts; in the first part we address the black hole attack whereas the second part is devoted to MAC layer misbehaviors. Prior to these two parts, Chapter 2 provides some basic definitions of security requirements in wireless multi-hop networks, their specific characteristics and presents the attacks that targets both routing and MAC protocols. This chapter helps the reader for better understanding the contributions presented in parts 1 and 2.

The part1, in its turn, is divided into two chapters 3 and 4, respectively. Chapter 3 analyzes and deeply investigates the black hole attack in MANETs. It, first, explains how an attacker or a group of attackers can mount such attack in both reactive and proactive routing protocols. Then, a classification of the existing solutions is given along with a critical discussion of each of them by outlining its advantages and drawbacks. Last, we highlight the challenges need to be tackled in order to design robust routing protocols.

Chapter 4 presents two solutions to struggle against single and colluding black hole attack in OLSR. The former solution uses an authenticated three hops acknowledgment to detect the malicious MPR nodes that refuse to forward TC messages. On the other hand, the latter solution uses hash functions and modifies the format of MAC frames to prevent a cross layer scenario of black hole attack.

In the second part, MAC layer misbehavior issues are addressed in both WMNs and MANETs. The first chapter in this part (Chapter 5) focuses particularly on the adaptive greedy behavior in WMNs. As the existing schemes are unable to deal with such misbehavior, we propose in this chapter our FLSAC scheme that exploits the strength of fuzzy logic to distinguish the adaptive cheater nodes from the well-behaving ones. Afterwards, to enhance its accuracy a Bayesian technique is developed and then integrated with it.

The second chapter of part2 (Chapter 6) is consecrated to investigate the greedy behavior in MANETs. This investigation aims to define new characterization of such misbehavior, which is more advantageous for the greedy node as compared to that adopted in many existing works. Additionally, we present robust scheme that allows prevention and fast detection of greedy nodes.

Finally, Chapter 7 concludes our dissertation and gives some future research directions.

1.3 Summary of our contributions

The major contributions of our dissertation lie in both of routing and MAC layers misbehaviors. More specifically, we propose a bunch of schemes to prevent these misbehaviors or at least alleviate their devastating impact on network performance. Those contributions are summarized in the following.

Three hops acknowledgement based scheme to cope with single and colluding black hole attack in OLSR

It is worth noting that the black hole attack, targeting routing protocol's control packets, is one of the most devastating attacks in wireless multi-hop networks. The goal of this attack is to force the well-behaving nodes to choose the hostile nodes as relays to disseminate the topological information, thereby they exploit the functionality of the routing protocol to retain/drop control packets. In particular, in optimized link state routing (OLSR) protocol, if a single or collusive black hole attack is launched during the propagation of the topology control (TC) packets, the topology information will not reach the whole network, which disrupts the routing operation (i.e. the route establishment). In this contribution, a three hops acknowledgment based scheme is proposed to deal with this attack. This scheme adds two extra packets to OLSR, Hello-rep packet which is a slight modification to Hello message and a small acknowledgment packet. The main idea of this scheme can be described as follows. Each MPR node M needs to learn the list of its 3-hop neighbors reached through a distinct pairs of two MPR nodes (M1, M2), where M2 is the MPR node of M1 and this latter is the MPR of the node M. Then, the node M selects one node, from this list, for which authenticated acknowledgment is requested. This acknowledgement is the unique proof that the TC packet has not been dropped during relay. Notice that the authentication function is performed using a pre-established secret key between node M and the requested node. When the number of missed acknowledgements overtakes a predefined threshold, then the MPR nodes M1 and M2, relaying M with the requested node, are deemed as misbehaving and consequently they will never be selected as MPRs.

Lightweight solution to cope with cross layer black hole attack in OLSR

It is well known that security attacks in MANETs are becoming a serious problem that may lead to harmful consequences on network performance. Despite that, many routing protocols still unable to cope with these attacks. Moreover, the vulnerability of MAC layer protocols to misbehaviors exacerbates the damage caused by the attacks at higher layers. Therefore, cooperation between layers becomes a must to face such attacks. In this work, we address one of the attacks targeting neighbor discovery operations in OLSR. This attack is launched at routing layer by carrying out a virtual link attack leading to establishment of fake symmetric link between the victim nodes, which are currently connected through an asymmetric link. So, an incorrect Multi-Point Relay (MPR) set may be elected by the victim nodes as well as by their neighbors, which leads to choose broken routes to forward data packets. Subsequently, the collaborator of the previous attacker carries out a false validation attack at MAC layer in order to reinforce the former attack and make it more destructive. To face this cross-layer attack, we propose a cross-layer scheme which requires that the routing layer gets a confirmation from MAC layer regarding the status of a specific link before advertising it to the network. In order to check the symmetry of a link, the Request to Send (RTS) and Clear to Send (CTS) frames formats are modified by replacing the destination address field by the hash value of a combination of the shared secret key between the sender and receiver nodes and RTS sequence number. Thus, the attacker node acting at MAC layer cannot validate the reception of the RTS and DATA frames being transmitted by one of the victim nodes since it cannot generate the correct value of the

destination address field. Therefore, this cross-layer attack can never be launched successfully unless the secret key is divulged.

Fuzzy Logic based scheme to Struggle against Adaptive Cheaters (FLSAC)

The growing popularity of Wireless Mesh Networks has opened the door to a bunch of attacks that may target their core functioning, which leads to sharp collapse of their performance. Hence, the need of robust and fast detection of these attacks became a major concern in order to guarantee an efficient and fair share of network resources among the nodes. One of these devastating attacks is the greedy behavior which leads to severe decrease of the bandwidth acquired by the neighbors of the attacker. In this study, we focus on such misbehavior and particularly on the adaptive greedy behavior, wherein the greedy node prefers to frequently switch between several cheating strategies rather than always applying one technique; thereby it avoids detection by the deployed schemes. To cope with such misbehavior, we propose a fuzzy logic based detection scheme, dubbed FLSAC. FLSAC is implemented in the mesh router/gateway to monitor the behavior of the attached wireless mesh clients. It then carries out a global assessment of all observed MAC parameters by applying fuzzy rules issued from IEEE 802.11 specification to identify any deviation from the proper protocol rules. The combined deviation of these parameters will be further used to classify a node into greedy or honest set following the membership function. According to the simulation results, FLSAC shows robustness and strong ability to quickly identifying the adaptive greedy nodes. Nevertheless, its performance didn't fulfil our goals. To enhance its detection accuracy, we integrate it with a bayesian probabilistic model that we have developed. The resulted scheme from this integration illustrates better detection rate and accuracy as compared to FLSAC.

Novel strategy for greedy behavior in MANETs

The strategy used in many research works to describe the greedy behavior in MANETs is assumed to be similar to that in WLAN. However, this assumption is neither realistic nor sustainable, since the greedy node that tries to monopolize the wireless medium, as it did in WLAN, will disrupt its own traffic flows due to the specific constraints and characteristics of MANETs. Therefore, in order to have a more beneficial greedy behavior strategy in MANETs, a node must adopt a different approach that allows it to achieve better performance for its own traffic flow as well as for the crossing flows of interest. To this end, we propose a novel strategy that uses conflict graphs and takes into account nodes' density and collision rate in the neighborhood of the greedy node to adjust the greediness parameters used by this latter to achieve its goals. Besides, our strategy is hard to be detected since the continuous adjustment of the cheating parameters will hide the greediness of the node from the monitor. Thus, it is suitable to be applied in MANETs.

New backoff scheme for IEEE 802.11 MAC protocol

This work designs new backoff scheme that ensures fast detection of the greedy nodes which either fabricates small backoff value or refuse to increase its contention window (CW) after an unsuccessful transmission. This scheme uses one way function to generate the backoff values with respect to the CW, and modifies the RTS frame format by piggybacking the data packet's CRC value along with the number of retransmission attempts. So any cheating attempt will be detected by the receiver node as well as by the other neighbors of the cheater, as long as the monitoring conditions are held. Furthermore, this scheme is robust against sender-receiver collusion and provides a novel reaction mechanism that gives the detected cheaters a chance to repent and comply again with

the protocol rules through the use of special warning message. The performance evaluation has confirmed the efficiency of this scheme and shown that it achieves a fairness index comparable to that of BEB algorithm, in several scenarios.

Chapter 2

Security Threats in Wireless Multi-hop Networks

In this chapter, we give a snapshot of three emergent classes of wireless multi-hop networks and explore the security threats related to each of them. We particularly describe the specific characteristics of Mobile Ad Hoc Networks (MANETs), Wireless Mesh Networks (WMNs) and Vehicular Ad Hoc Networks (VANETs), and show that these characteristics make those networks more vulnerable to security attacks. Moreover, we address some examples of attacks that target routing and MAC layers.

2.1 Network security requirements

In this section, we discuss the security requirements usually expected to be met by a secure system. The security services of wireless multi-hop networks are not altogether different from those of other systems. The purpose of these requirements is to protect the exchanged information and nodes' resources from any misuse. The main requirements that effective security architecture must ensure are explained in the following;

Authentication is needed in order to be sure about the identity of the sender or receiver of a packet. It also ensures that communication from one node to another is genuine. In other words, it prevents an attacker from impersonating a trusted node.

Confidentiality protects the content of a message and ensures that it cannot be understood by anyone other than its (their) desired receiver(s). Data confidentiality is typically enabled by applying symmetric or asymmetric data encryption. This property is particularly important in wireless networks since the broadcast nature of the medium facilitates eavesdropping.

Integrity ensures that the transmitted messages are not illegitimately altered during transmission. This alteration includes changing the content, status, deleting, delaying or replaying of the transmitted messages by the intermediate nodes.

Availability ensures that the services are usable when needed, in spite of the presence of attacks. In wireless multi-hop networks of potentially low powered devices, sleep deprivation or incorrect forwarding of messages are effective threats that lead to denial of service (DoS). So, to guarantee the availability property, it is required to develop robust mechanisms that struggle against all types of DoS attacks.

Non Repudiation in computer networks, non-repudiation property means that a node cannot repudiate having sent or received a message. A typical attack is masquerading that may be prevented by using digital signature.

2.2 Wireless Multi-hop Networks: an overview

In general, wireless networks refer to the use of radio frequency signals to share information and resources between devices. Due to the fundamental differences found in their physical layer, wireless networks and devices show distinct characteristics from their wire line counterparts, specifically,

- higher interference due to the broadcast nature of transmission.
- limited bandwidth and much lower transmission rates, typically much slower speed compared to the wired networks, which leads to severe degradation of QoS, including jitter and delays.
- Eavesdropping: the unreliable wireless links facilitate eavesdropping. By using an antenna which is placed at an appropriate location, any node can overhear the packets sent by its neighbors.
- The lack of protected wired link makes it easier for an attacker to impersonate a legitimate user.
- To be mobile the wireless device must be small, which means that it has limited storage, computing capabilities, and energy resources. The energy issue is the most significant since technological progress on batteries is much slower than on electronics. To solve this problem, the number of computational operations to be performed by the mobile devices should be reduced. Therefore, it is required that security protocols do not generate high computational overhead, which may significantly reduce their efficiency.
- multiple paths are likely to be available due to sufficient node density. Therefore, multiple copies of a packet or parts of it can be routed through several paths to increase the probability of a packet being delivered successfully to its destination.
- weaker security: as the wireless channel is accessible to everyone, network security is more difficult to implement, as attackers can interface more easily.

In what follows, we provide a brief description of the most promising models or classes of wireless multi-hop networks, namely MANETs, WMNs and VANETs.

2.2.1 Mobile Ad Hoc Networks (MANETs)

MANETs [1] are wireless multi-hop networks dynamically constructed by mobile nodes without the aid of any established infrastructure. This new paradigm of wireless communications aims to make communication possible in some situations where the services offered by both wired networks and WLAN are unavailable. MANETs are mainly useful in military and other tactical applications such as emergency rescues. Moreover, we can set up an ad hoc network at a conference to distribute files and discuss talks without using any wireless infrastructure that would have to be paid. Figure 2.1 illustrates some examples of MANETs applications. In this network, nodes should collaborate with each other to support the network functions. However, due to the self-organized nature and insufficient resources, some of them may misbehave to fulfill their individual interests (e.g., drop or mis-route packets). Hence, countering such misbehavior is a critical issue that we will tackle throughout the next chapters.

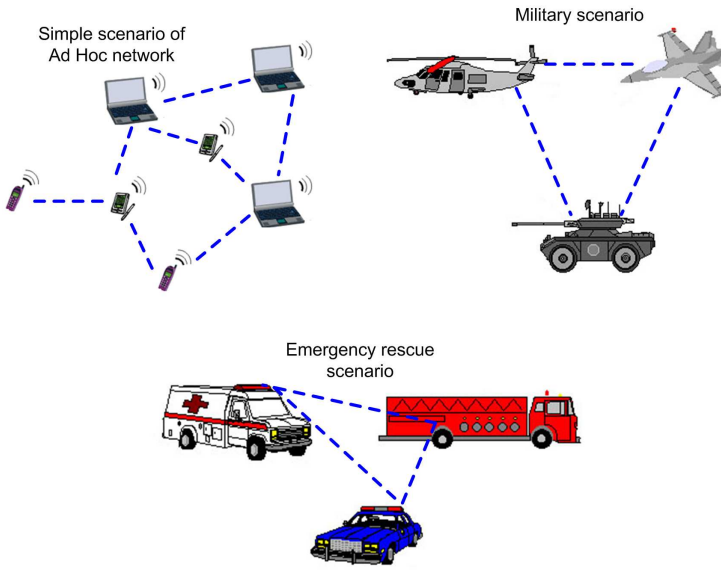


FIGURE 2.1: Example of applications of Mobile Ad Hoc Networks

2.2.2 Wireless Mesh Networks (WMNs)

WMNs are particular wireless multi-hop networks, which consists of three entities: mesh routers (MRs), gateways (GWs) and mesh clients (MCLs), as shown in Figure 2.2. The gateways are usually equipped with multiple interfaces (wired and wireless) and serve as internet access points to the MCLs. These gateways can be either stationary or mobile (e.g, airplane, buses/subway). In WMNs, a large number of MRs is required in order to provide reliable service. Each MR has at least one wireless interface and acts as a repeater to transmit data from nearby routers/clients to the remote peers. In such networks, the MCLs are the only sources/destinations of data traffic flows. The connection to the mesh network is provided through the wireless MRs (or directly through the GWs). In what follows, we summarize the main advantages of WMNs as compared to MANETs [2];

- WMNs provide 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 MANETs, WMNs provide larger coverage, connectivity and robustness due to redundancy.
- WMNs can integrate with several types of networks, including Internet, WiMAX [3], Wireless Sensor networks (WSNs) [4], etc., using gateways technologies.
- Mesh connectivity significantly enhances network performance, such as fault tolerance, load balancing, throughput and protocol efficiency.

As a consequence of these characteristics, WMNs are widely used in many applications. Consequently, WMNs should provide high resistance to various abuses and security attacks.

2.2.3 Vehicular Ad Hoc Networks (VANETs)

VANETs are composed by vehicles equipped with wireless network devices that are able to spontaneously interconnect each other without the aid of any infrastructure. VANETs are a cornerstone of the envisioned Intelligent Transportation Systems (ITS)[5]. Vehicles communicate with each other via Inter Vehicle Communication (IVC)

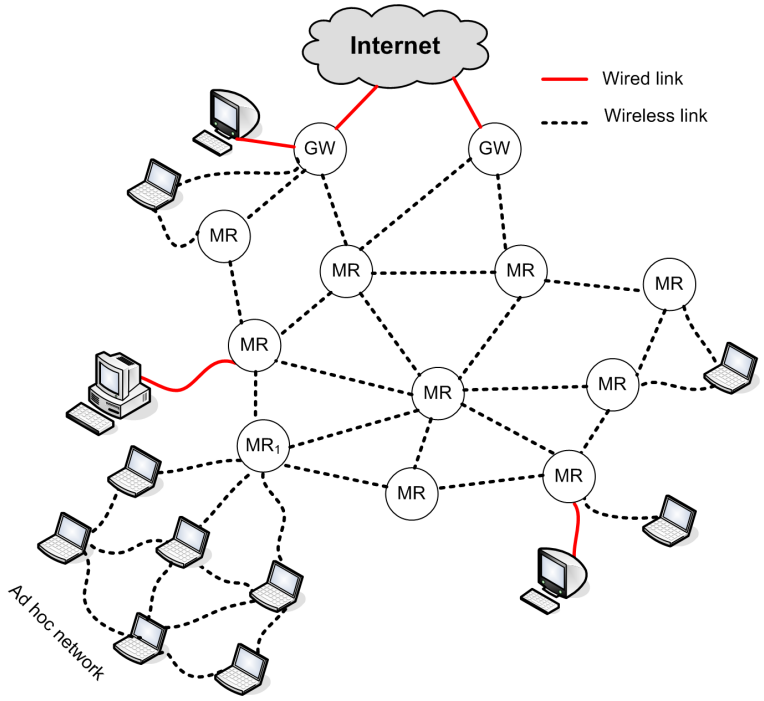


FIGURE 2.2: Wireless Mesh Network Architecture

as well as with roadside base stations via Vehicle-To-Infrastructures Communication (V2I). The main goal of IVC and V2I technologies is to provide each vehicle with timely information about its surrounding and road conditions in order to assist the driver avoiding potential dangers.

VANETs represent a particularly challenging class of MANETs and one of their concrete applications, characterized by relatively high mobile nodes with speeds varying from 0 to 30 m/s. Moreover, unlike many other MANETs environments, where node's movement occurs in an open field (such as conference rooms and cafes), vehicles are constrained to predefined streets often separated by buildings, trees or other obstacles, thereby increasing the average distance between vehicles and, in most cases, reducing the overall signal strength received at each vehicle.

Recently, a number of attractive applications of VANETs have emerged. These applications can be classified into three main categories as follows [6]:

- **Information and warning functions**: its main role is the dissemination of road information, such as car accidents, traffic density, surface condition, etc., to vehicles far away from the subjected site. These warning messages are exchanged between nearby vehicles as well as between vehicles and road side infrastructure to ensure a large spreading of information and enhance the road safety, as shown in Figure 2.3.
- **Communication-based longitudinal control**: exploiting the "look-through" capability of IVC technology to help avoiding accidents and platooning vehicles to improve road capacity.
- **Cooperative assistance systems**: it aims to coordinate vehicles at critical points such as blind crossings (a crossing without light control) and highway entries.

To enable the deployment of IVC system, a set of security mechanisms must be designed to ensure its safety; otherwise, the efficiency of the transportation systems, as well as the physical safety of vehicles, drivers, and passengers could be jeopardized. In addition, VANETs are particularly challenging to secure due to the tight

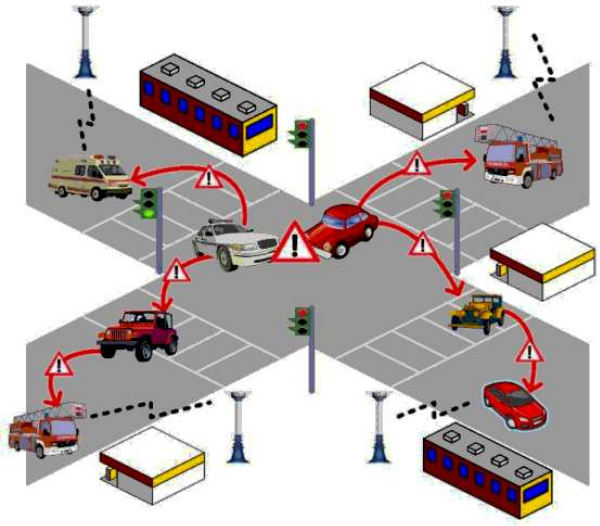


FIGURE 2.3: An example of VANETs application

Features	**MANETs**	**WMNs**	**VANETs**
Transmission	Multihop	Multihop	Multihop (IVC mode) One hop (V2I mode)
Network entities	Mobile Nodes	Mesh Clients Mesh Routers Gateways	Vehicles Roadside Infrastructure
Network size	Hundreds	Thousands	Hundreds (Urban area) Dozens (Rural area)
Mobility	Medium	Stationary entities (MRs, GWs) Mobile entities (MCLs)	High
Moving trajectory	Random	Random for MCLs	Predefined
Wired connectivity	No	Yes	No
Limitations	Energy Bandwidth Processing capabilities	Bandwidth Fixed deployment Processing capabilities for MCLs	Bandwidth Frequent vehicles disconnection due to the high speed

TABLE 2.1: The key difference between MANETs, WMNs and VANETs

coupling between applications and the networking material, as well as additional societal, legal, and economical considerations, which raise an unique combination of operational and security requirements.

2.3 Attacks targeting Wireless Multi-hop Networks

Wireless multi-hop networks are exposed to a bunch of attacks at different layers. In this dissertation, we focus on the security attacks targeting both network and MAC layers. This choice is justified by the fact that those two layers constitute the foundation stone of the network functions since MAC layer is managing the access to the wireless medium and network layer establishes routes towards distant nodes. Therefore, any fail at these layers will affect the functioning of the rest of the upper layers and jeopardize the network performance.

2.3.1 Attacks at network layer

We distinguish two types of attacks at this layer; attacks targeting routing function and those targeting the forwarding function, as described below.
Attacks on routing: in this type, the attacker node misbehaves when it generates the control packets that transport routing information; it may for example

- advertise false information for distance vector routing approach.
- change the path for source routing approach.
- advertise false links set or state information for link state routing approach.

Attacks on forwarding: these attacks occur at the forwarding phase where the intermediate nodes misuse the received packets rather than forwarding them correctly. The following are some of these misuses

- dropping data or control packets.
- altering their content.
- intercepting and redirecting packets.
- eavesdropping.

Based on the above analysis, we now describe some examples of these attacks.

2.3.1.1 Black hole attack

In this attack, a misbehaving node exploits the vulnerabilities of routing protocols in order to get involved in different routes, and thereby it drops all the data or control packets passing through it. This attack will be deeply studied in next chapter.

2.3.1.2 Wormhole attack

A wormhole attack [7] is composed of two attackers and a wormhole tunnel. To launch such attack, the attackers create a direct link, referred to as a wormhole tunnel, between them. Wormhole tunnels can be established by means of a wired link, a high quality wireless out-of-band link or a logical link via packet encapsulation. After building a wormhole tunnel, one attacker receives and copies packets from its neighbors, and forwards them to the other colluding attacker through the wormhole tunnel. This latter node receives these tunneled packets and replays them in its neighborhood. In a wormhole attack using wired links or a high quality wireless out-of-band

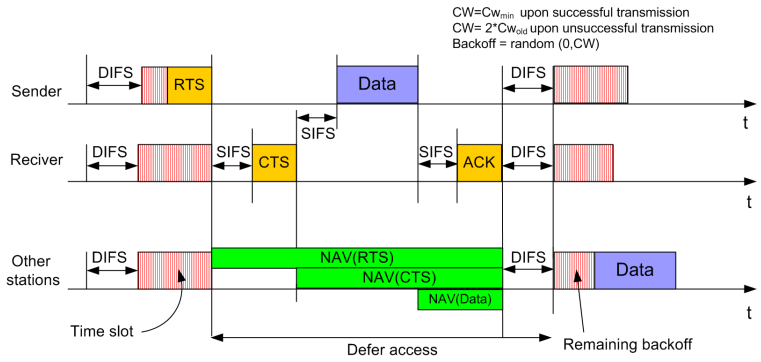


FIGURE 2.4: IEEE 802.11 DCF protocol functioning

links, attackers are directly linked to each other, so they can communicate swiftly. However they need special hardware to support such communication. On the other hand, a wormhole using packet encapsulation is relatively much more slower, but it can be launched easily since it does not require any special hardware nor special routing protocols.

2.3.1.3 Sybil attack

A sybil attack is essentially an impersonation attack, in which a malicious node obtains multiple fake identities and pretends to be multiple, distinct nodes in the network, and behaves accordingly. There are, mainly, two different ways through which a sybil node can get an identity; it can fabricate one (for instance, creating an arbitrary identifier if the network has no restriction to the allowed identities) or it can steal an existing valid one from a legitimate node. By impersonating a large number of nodes in the network, the malicious node is able to ”out vote” the well-behaved nodes in collaborative tasks such as Byzantine failure defenses. One important result reported in [8] states that without a logically centralized authority, sybil attacks are always possible (i.e. the sybil nodes may escape from detection systems) except under extreme and unrealistic assumption of resource parity and coordination among nodes.

2.3.2 Attacks at MAC layer

Before discussing the potential attacks that may be launched at this layer, we give an overview on IEEE 802.11 MAC protocol and explain its main operations.

2.3.2.1 IEEE 802.11 MAC protocol overview

The 802.11 standard [9] specifies a common 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 it. Often viewed as the ”brain” of the network, the 802.11 MAC layer uses a dedicated physical layer, such as 802.11b or 802.11g, to perform the tasks of carrier sensing, transmission, and reception 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)

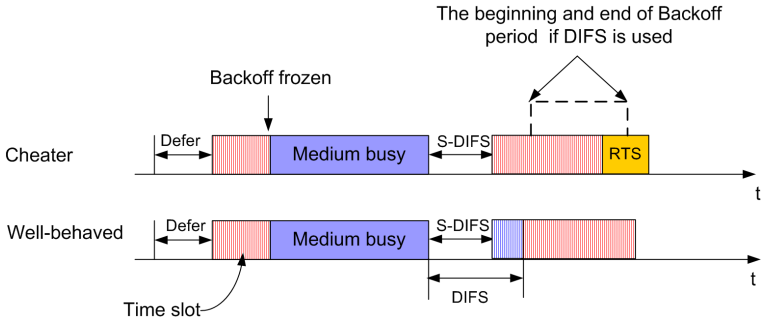


FIGURE 2.5: Cheater node gains access to the medium ahead of schedule because it starts decreasing its backoff before the well-behaved nodes

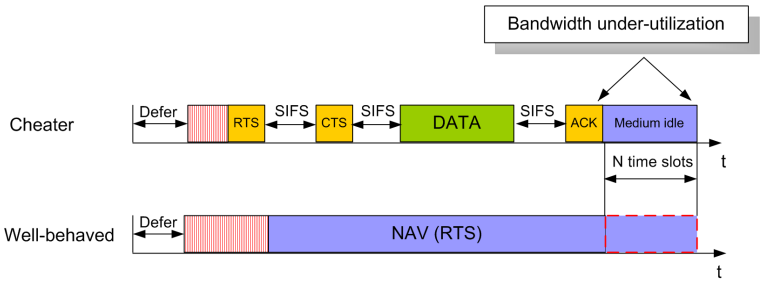


FIGURE 2.6: Consequences of NAV inflation misbehavior

mechanism. It is designed to support best effort traffic, like internet data, that does not require any service guarantees. Additionally, IEEE 802.11 incorporates an optional access method in which the 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 required.

In the DCF mode, a node 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 node may wait for DCF Inter Frame Space (DIFS) time slot after a successful transmission or Extended Inter frame Space (EIFS) period when collision occurs. If the medium is sensed busy, the node freezes its backoff timer and sets its Network Allocation Vector (NAV)[1] to the expected duration of transmission indicated in the received frame. Transmission shall start whenever the backoff timer reaches zero.

If a CTS or an ACK (ACKnowledgement) frames corresponding to a DATA packet are not received within a predetermined period of time (timeout), then the sender assumes a transmission failure. A transmission failure, like collision, leads the node 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. However, in case of an unsuccessful attempt the CW is doubled until it reaches CWmax and remains unchanged till it is reset to CWmin. Notice that the CW is reset either when the DATA packet is delivered successfully or the maximum retry limit is reached. In this latter case, the retransmission shall stop, CW will be reset to CWmin and the DATA packet is discarded.

[1]The NAV is a timer that indicates the amount of time during which the wireless medium will be reserved

2.3.2.2 MAC layer misbehavior in DCF mode

As stated above, two medium access techniques exist in IEEE 802.11 MAC protocol, PCF and DCF. While PCF is reserved for Wireless Local Area Networks (WLAN), the DCF technique can be used in both modes WLAN and infrastructure-less wireless networks. Therefore, we only discuss, in what follows, the potential vulnerabilities of the DCF mode. For better understanding of MAC layer misbehaviors, we classify them into two categories, misbehaviors conducted by the sender of MAC frame and those carried out by the receiver node.

Sender-side misbehavior

A misbehaving sender node may disobey the MAC protocol rules to gain more bandwidth over the honest nodes. To do so, it should modify the MAC layer parameters. However, this is possible only if network access cards run the MAC protocol on software. In this case, the misbehaving node can easily implement the following misbehavior techniques.

- 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 misbehaving node behaves correctly however it doesn't not double its CW after collision. Moreover, it can adopt different retransmission strategies when experiencing unsuccessful transmission. Furthermore, it may also access the medium without passing through the backoff procedure or always wait for a constant short period.
- when the channel is sensed idle, it transmits before the required DIFS time slots elapse, i.e. it waits for a shorter period called Short-DIFS (S-DIFS). The consequence of this misbehavior technique is similar to that of selecting a small backoff as illustrated in Figure 2.5.
- Scrambles CTS, ACK or DATA frames sent by its neighbors nodes in order to increase their CWs, as shown in Figure 2.7.
- amplifies the value of the duration field in Request To Send (RTS) or DATA packets, such that the receiver nodes update their NAVs according to the received duration value, as illustrated in Figure 2.6. 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. Notice that the misbehaving node can increase its duration filed up to (2^{15}-1) or 32767 μs, which is the maximum allowed value in IEEE 802.11 (i.e. because we only have 16 bits to define the duration field in each MAC frame).

A misbehaving node which has some knowledge about the deployed detection scheme can easily switch frequently between the above techniques to avoid detection. Moreover, it is worth noting that it is not compulsory for the misbehaving node to know the exact parameters of the detection system. A rational switch rate between the different misbehavior techniques, without large deviation from the standard (for each technique), allows it to acquire more bandwidth than the well behaving nodes.

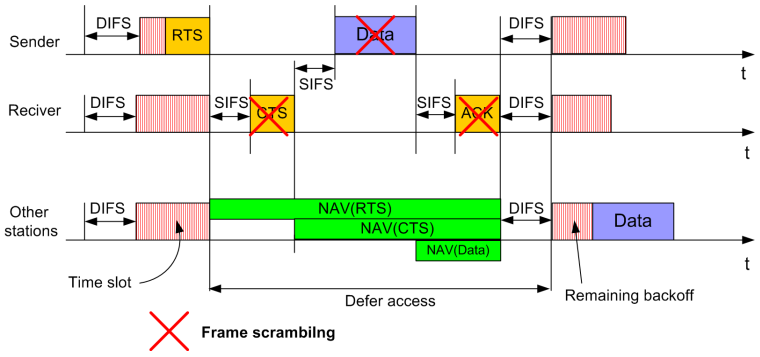


FIGURE 2.7: Scrambling MAC frames (CTS, DATA and ACK)

Receiver-side misbehavior

All the previous discussed misbehaviors are conducted by the sender node which wants to increase its gained bandwidth, however the receiver node may also misbehave for either saving its resources (selfish intention) or causing damage to its neighbors (malicious intention). To this end, the receiver node may apply several strategies to achieve its objectives. In what follows, we describe these strategies and their impacts on network performance.

- a selfish receiver node may deny the response to an RTS frame destined to it, which corresponds to the Route REPly (RREP) packet used in reactive routing protocols, leading to timeout expiration at the sender side. Therefore, after several retransmission attempts the corresponding data packet will be discarded and the receiver will not be involved in the established route.

- similarly to the sender, a misbehaving receiver may also inflate the NAV value in CTS and ACK frames in order to silence all its neighbors for a longer period than the required time for the ongoing transmission. Hence, this allows the sender node to access easily to the wireless medium since the number of contending nodes around it is reduced. This is due to the fact that according to IEEE 802.11 rules each neighbor of the receiver node updates its NAV value only if it is smaller that the received one and also if this node is not the destination of the received frame. If the misbehaving receiver is the destination of TCP traffic, it can also inflate the NAV value in RTS and Data frames corresponding to the TCP ACK packet. This misbehavior allows the receiver to significantly increase its received throughput, whereas it leads to sharp collapse of the received throughput of the contending flows.

- the format of MAC frames, particularly CTS and ACK, presents a vulnerability that can be exploited by the misbehaving receiver to cause severe damage to TCP flows running by its neighbors. This misbehavior is effective only if the TCP Data frame is lost or corrupted. In this case, the misbehaving receiver sends ACK frame on behalf of the intended destination. Upon reception of this ACK, the sender of the Data frame schedules the transmission of the next frame, instead of performing Data retransmission. As a consequence, the sender of TCP flow will decrease the size of the congestion window due to the lack of the TCP ACK packet corresponding to the lost packets. This decrease of the contending TCP flows rate allows the misbehaving receiver to increase its received traffic rate.

  Notice that even though the Data frame is not lost or corrupted, the fake ACK sent by the misbehaving receiver will collide with the ACK sent by TCP flow's destination, which decreases its received packets rate.

2.4 Conclusion

Securing routing and MAC layers' protocols against attacks in wireless multi-hop networks, remains a challenge that needs careful investigation from the research community. This task is particularly challenging due to the specific characteristics of wireless multi-hop networks, which add more constraints on the developed security solutions. For example, the limited battery power and processing capacity of wireless devices require that these solutions be lightweight in computation.

Throughout this chapter we have presented the common features of wireless multi-hop networks and highlighted their impacts on the security of these networks. As stated earlier, these features such as shared wireless medium, mobility of nodes and strict energetic constraints (for MANETs and WMNs) make the task of ensuring the security requirements hard to be accomplished. Even worse, the proposed solutions to fulfill these requirements may open the door to DoS attacks, especially if they use cryptographic schemes for authentication and data integrity preservation purposes.

Additionally, we have presented the main attacks that may target routing protocols and given some examples of them. Among them the black hole attack to which we will devote Chapters 3 and 4 to analyze its causes, consequences and propose effective countermeasures to deal with it. On the other hand, we have provided a snapshot on the different misbehavior techniques used by the misbehaving nodes at MAC layer. In this case, we have shown that both the sender and receiver nodes may misbehave in order to gain advantages over their honest neighbors, where each of them uses specific techniques according to its objectives.

To conclude, this chapter serves as an introduction to the subsequent chapters and its content will help the reader for better understanding of our contributions.

Part I

The Black Hole Attack

Chapter 3

Studying Black Hole Attack in MANETs

Nodes, in MANETs, usually cooperate and forward each other's packets in order to enable out of range communication. However, in hostile environments some nodes may refuse to do so for either saving their resources or intentionally disrupting regular communications. This type of misbehavior is generally referred as *packet dropping* attack or *black hole* attack, which is considered as one of the most destructive attacks that leads to the deterioration of network performance. In this chapter, we will analyze this attack and conduct a comprehensive investigation on state-of-the-art countermeasures to face it. Furthermore, we examine the challenges that must be tackled for constructing an in-depth defense against such sophisticated attack.

3.1 Introduction

Mobile ad hoc networks (MANETs) are usually formed by a group of mobile nodes, interconnected via wireless links, which agree to cooperate and forward each other's packets. One of the basic assumptions for the design of routing protocols in MANETs is that every node is honest and cooperative. That means, if a node claims it can reach another node by a certain path or distance, the claim is trusted/true; similarly, if a node reports a link break, the link will no longer be used. While this assumption can fundamentally facilitate the design and implementation of routing protocols, it meanwhile introduces a vulnerability to several types of denial of service (DoS) attacks [10], particularly packet dropping attack. To launch such attack, a malicious node can stealthily drop some or all data or routing packets passing through it.

Due to the lack of physical protection and reliable medium access mechanism, packet dropping attack represents a serious threat to the routing function in MANETs. A foe can easily join the network and compromise a legitimate node then subsequently start dropping packets that are expected to be relayed in order to disrupt the regular communications. Consequently, all the routes passing through this node fail to establish a correct routing path between the source and destination nodes.

Although upper layer acknowledgment, such as TCP ACK (Transmission Control Protocol ACKnowledgment) can detect end-to-end communication break, it is unable to identify accurately the node which contributes to that. Moreover, such mechanism is unavailable in connectionless transport layer protocols like UDP (User Datagram Protocol). Therefore, securing the basic operation of the network becomes one of the primary concerns in hostile environments in the presence of packets droppers. The challenge lies in securing communication meanwhile maintaining connectivity between nodes despite of the attacks launched by the foes and the frequently changing

topology. It is thus obvious that both phases of the communication, mainly route discovery and data transmission phase, should be protected, calling for comprehensive security studies.

While a number of surveys [11, 12, 13] and [14], dealing with security threats against routing protocols in MANETs, have provided some insightful overviews on different threats and countermeasures, none of them focuses on a specific attack and examines all its characteristics in different routing techniques. To complement those efforts, this work studies the packet dropping attack, which is known as one of the most destructive threats in MANETs, and illustrates in depth the different schemes used by adversaries targeting on both reactive and proactive protocols. Furthermore, we conduct an up-to-date survey of the most valuable contributions aiming to avoid the packet droppers. The careful examination and analysis has allowed us to carry out a comparative study of the existing security schemes in terms of specific design rationale and objectives. The ultimate goal is to identify the strengths and weaknesses of each scheme in order to devise a more effective and practical solution which can achieve a better trade-off between security and network performance.

The remainder of this chapter is structured as follows. In next section, we discuss the root causes of dropping packets in MANETs. Section 3.3 describes the Black hole attack in both reactive and proactive routing protocols. An overview of the proposed security schemes for defending against this attack is given in section 3.4. In section 3.5, some open challenges related to the herein presented attack and solutions are highlighted. Finally, section 6.3 concludes this chapter and points out future research directions.

3.2 Root causes of packet dropping in MANETs

Before analyzing the packet dropping attack in details, let us first summarize the different motives that incite some nodes to drop a packet rather than sending or relaying it. In general, a packet can be dropped at either MAC or network layers due to the following reasons:

- The size of packets' transmission buffer at MAC level is limited; therefore whenever the buffer is full any new packet arriving from higher layers will be dropped (buffer overflow).
- IEEE 802.11 protocol's [9] rules: a data packet is dropped if its retransmission attempts or the one of its corresponding RTS (Request To Send) frame has reached the maximum allowed number, owing to node's movement or collision (a lot of contending nodes).
- A data packet may be dropped or lost if it is corrupted during transmission due to some phenomenon specific to radio transmissions such as interference, hidden nodes and high bit error rate.

In addition to these causes, a selfish node may refuse to relay a packet aiming to economize its energetic resources in order to extend its lifetime or simply because its battery power is drained. Moreover a malicious node involved in a routing path may intentionally drop the packets at network layer in order to provoke a collapse in network performances. Furthermore, it can modify the IEEE 802.11 MAC protocol's parameters to provoke packet dropping. According to this analysis, packet dropping problem still open the door to new challenges in MANETs. For example, how can we recognize the reason leading a node to drop others' packets? In other words, how can we know the intention of a node to accuse it as malicious, selfish or legitimate?

3.3 Black hole attack in MANETs

The black hole attack in MANETs can be classified into several categories in terms of the strategy adopted by the malicious node to launch the attack. In particular the malicious node can intentionally drop all the forwarded packets going through it (black hole), or it can selectively drop the packets originated from or destined to certain nodes that it dislikes. Furthermore, a special case of black hole attack dubbed *gray hole* attack is introduced in [15]. In this attack, the malicious node retains a portion of packets (one packet out of N received packets or one packet in a certain time window), while the rest is normally relayed.

In order to launch a black hole attack, the first step for a malicious node is to find a way that allows it to get involved in the routing/forwarding path of data/control packets. To do so, it exploits the vulnerabilities of the underlying routing protocols which are generally designed with strong assumption of trustworthiness of all the nodes participating in the network. Thus, any node can easily misbehave and provoke a severe harm to the network by targeting both data and control packets.

Dropping data packets leads to suspend the ongoing communication between the source and the destination node. More seriously, an attacker capturing the incoming control packets can prevent the associated nodes from establishing routes between them. To facilitate understanding, we illustrate them using two representative routing protocols in MANETs, OLSR (Optimized Link State Routing)[16] and AODV (Ad hoc On Demand Distance Vector)[17], which are table-driven and on-demand respectively.

3.3.1 Routing protocol-specific attack

We first address black hole problem in the two routing protocols cited above.

3.3.1.1 Black hole attack in AODV

In order to discover a new path towards a faraway destination, the source node broadcasts a RREQ (Route REQuest) message with unique identifier to all its neighbors. Each receiver rebroadcasts this RREQ to all its neighbors until reaching the intended destination as depicted in Figure 3.1. On receiving the RREQ message, the destination node updates the sequence number of the source node and sends a RREP (Route REPly) message back to its neighbor which has relayed the RREQ. On the other hand, an intermediate node having a route to the destination with destination sequence number greater or equal to that in RREQ can send back a RREP packet to the source node without relaying the RREQ to the destination. Notice that the links between nodes may be lost due to nodes' mobility, so a RERR (Route ERRor) message is generated and forwarded back to the source node to report the link failure. Thus, the source node initiates a new route discovery to replace the failed path.

The DSR (Dynamic Source Routing) [18] protocol uses the same mechanism as AODV to discover new routes, however the complete path to the destination is chosen by the source node and loaded in the packet header. All the intermediate nodes have to relay the packets with respect to the route specified in the packet header. This feature is important in some cases in order to satisfy QoS [19] requirements by performing load balancing between the relay nodes. In this case, the source node sends the packets through different paths to avoid overloading any node in the network.

In on-demand routing protocols, dropping control packets might be the greatly benefit for both selfish and malicious nodes. Specifically, once dropping the RREQ packets, a selfish node prevents the established routes from passing through it and consequently it saves its energy for transmitting its own packets. Likewise, a malicious node

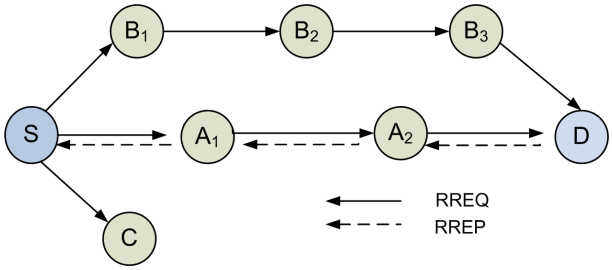


FIGURE 3.1: Route discovery in AODV

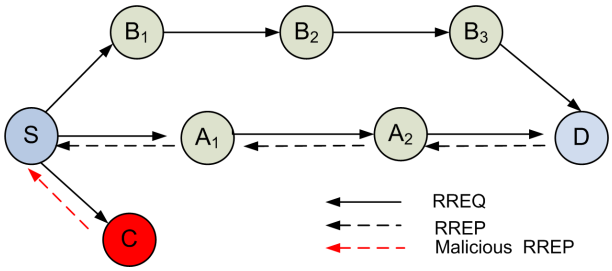


FIGURE 3.2: Black hole attack in AODV

can drop the RERR packets in order to prolong the duration of use of the broken routes. As a result, the network throughput collapses sharply since no packet reaches its destination.

A prerequisite for a node to launch a black hole attack is to be involved at least in one routing path. To this end, the malicious node applies the strategies illustrated below.

- As shown in Figure 3.2, C is a malicious node whereas S and D are the source and destination nodes, respectively. First, the node S broadcasts RREQ packet to its one hop neighbors. Then, upon receiving this packet each neighbor node is supposed to rebroadcast it if a route cache towards the destination is unavailable. However, the node C disobeys this rule and claims that it has the shortest path to the destination and sends a RREP packet back to node S. Consequently, if the RREP packet sent by node D or any honest intermediate node, which has a fresh route to D, reaches the node S before the C's RREP then everything works well. Otherwise, the source node S deems that the route passing through the node C is the shortest path, and thus it starts transmitting data packets towards C which in its turn drops them.
- Another strategy to launch the attack can be described as follows: an intermediate node C spoofs the IP address of the destination D, inciting the source node S to establish the path towards C, instead of D. To illustrate that let us consider the network topology depicted in Figure 3.2, when the attacker node C receives a RREQ packet it transmits a RREP packet to reply back to S claiming that it is the intended destination. Moreover, it increases the Destination Sequence Number (Dst-Seq-Num) received in RREQ packet by a value larger than one as shown in Table 3.1, where the node C sets Dst-Seq-Num to 55 rather than 41 to guarantee that the source node S chooses it as the actual destination. The consequences of this attack strategy are similar to the previous one.

3.3.1.2 Black hole attack in OLSR

OLSR is a proactive routing protocol designed for large and dense networks. The main optimization of this protocol is achieved through the use of MPRs (MultiPoint Relays) which are a set of neighbor nodes that represent

	RREQ			**RREP**			
Sender	S	A_1	A_2	D	A_2	A_1	**C**
IP-src	S	A_1	A_2	D	A_2	A_1	**D**
Dst-adr	D			S			S
Dst-Seq-Num	40			41			**55**

TABLE 3.1: The values of the different fields of RREQ and RREP packets sent or forwarded by both legitimate and malicious nodes: (i) the nodes A_1 and A_2 forward correctly the RREQ and RREP packets (ii) the node C spoofs the destination node's address (D) and augments illegitimately the Dst-Seq-Num

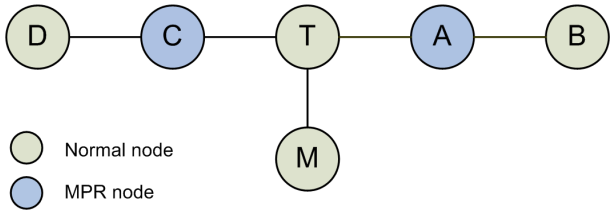


FIGURE 3.3: The MPR set of node T before launching the attack

Originator-adr	**1-hop neighbors**
M	T, **B**, **D**

TABLE 3.2: Example of Hello message sent by node M

the unique responsible for spreading the local link state information to the whole network, thereby reducing the induced overhead. Notice that the local link state information is periodically advertised by the MPR nodes via the transmission of TC (Topology Control) messages. In OLSR, each node selects its MPR set from its one hop neighbors set such that it can easily reach all its two hop neighbors with minimum number of retransmissions. The MPR selection function depends on the number of two hop neighbors reachable through the candidate node and its 'Willingness' value obtained from Hello message. This value indicates the readiness of a node, according to its own resources, to forward the packets of its neighbors. Nodes with higher willingness value are given higher priority to be selected as MPR.

The main functionality of OLSR is neighbor sensing and topology dissemination. Neighbor sensing is accomplished through the periodic exchange of Hello messages, in which every node advertises its neighbor set along with the state of the link connecting it to each neighbor. In addition to that, it indicates whether a given neighbor has been chosen as MPR or not. To disseminate the topological information, each MPR node broadcasts periodically a TC message that contains its MPR selectors set. Using this information, each node constructs a partial topology graph of the network which allows it to establish routes to non-neighboring nodes.

Since TC messages are flooded across the whole network, the attack can occur either at the origin or forwarding point. The damage resulted from targeting a TC message is more severe than that caused by misusing Hello messages as the TC messages are used globally by the whole network for routes calculation. A malicious node may simply send a TC message claiming to be the MPR of nodes although it is not. Therefore, as the network depends on the MPRs for routing services, a malicious node that manages to become an MPR can easily launch a black hole attack on the network. In what follows, we present the strategy adopted by a node to launch a black hole attack.

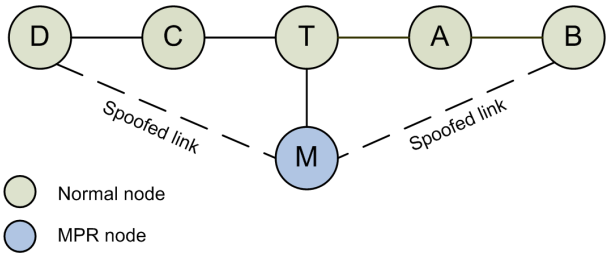


FIGURE 3.4: The new MPR set of node T after the spoofing link attack is launched

- **Gain an MPR position in the network**: a simple way for a malicious node to be an MPR is to set constantly its willingness field to the highest allowed value regardless of its available resources. Thus it compels all its neighbors to elect it as MPR. Besides, it may force a target node to select it as the only MPR by spoofing links with all its 2-hop neighbors as described below. To illustrate this scenario, let us consider the network topology depicted in Figure 3.3, where the nodes A and C constitute the MPR set of the node T. The malicious node M generates its Hello message in which it advertises the non-neighboring nodes B and D as its neighbors, as illustrated in Table 3.2. According to the MPR computation heuristic [16], the node T must choose M as the only MPR node, as shown in Figure 3.4, since it has connections to the whole set of its two hop neighbors (B, D). Notice that the node M can learn the T's two hop neighbors set by analyzing the received TC messages along with the T's Hello message.

- **Drop all control or data packets supposed to be relayed**: as an MPR, a node can carry out the following disruptions:

  - Correctly participates to TC message forwarding function but fails to deliver data packets for other nodes.
  - Drops all TC messages sent or relayed by its MPR selector nodes. For example, in the network topology depicted in Figure 3.5 the malicious node M refuses to relay the TC messages generated by the node T. Thus this makes the routes towards the MPR selectors of node T unknown for the rest of the network. The Figure 3.6 illustrates that, where the nodes A_1, A_2 and A_3 are hidden from the nodes B and C because the T's TC message has not been received.
  - Colludes with another neighbor MPR node to make the previous attack harder to be detected as illustrated in [20].

In heterogeneous networks such as MANETs the status of asymmetric links is more likely to be observed. As an example, the topology depicted in Figure 3.7 shows two asymmetric links connecting T_1 with T_2 and T_2 with M. Malicious nodes (such as node M in Figure 3.7) may get benefits from that and exploit it to launch a black hole attack. To do so, the node M tries to create a false symmetric link between T_1 and T_2. The establishment of this fake symmetric link requires five steps as follows:

msg1: $T_1 \longrightarrow * : Hello, \{\emptyset\}$.

During neighbor discovery phase, the node T_1 broadcasts an empty Hello message that reaches both nodes T_2 and M.

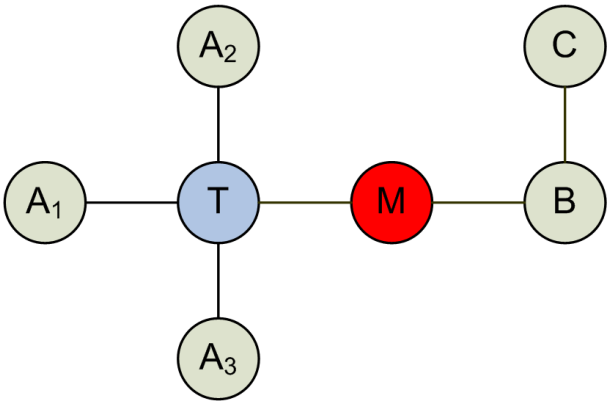


FIGURE 3.5: The network topology held by the nodes B and C before the attack, where they are able to communicate with the T's MPR selectors nodes.

msg2: $T_2 \longrightarrow * : Hello, \{T_1, ASYM\}$.

Next, the node T_2 advertises, in its Hello message, that the node T_1 is an asymmetric neighbor.

msg3: $M \longrightarrow T_1 : Hello, \{T_1, ASYM\}$.

Upon receiving the message msg2, the node M maliciously forwards it to the node T_1 albeit it is not supposed to do.

msg4: $T_1 \longrightarrow * : Hello, \{T_2, SYM\}$.

When the message msg3 reaches the node T_1, it finds its identity included in the advertised neighborhood list and consequently it concludes that it is a symmetric neighbor of T_2. Hence, it advertises this new link status in its Hello message.

msg5: $T_2 \longrightarrow * : Hello, \{T_1, SYM\}$.

On receiving the Hello message msg4, the node T_2 changes its link status with T_1 to symmetric.

As a result, the victim nodes T_1 and T_2 infer that they are connected through a symmetric link while it is not. So, all control packets, such as TC messages, generated by the MPR selectors of node T_2 will not reach the whole network. As a result, the network may be partitioned.

Notice that $*$ denotes the dissemination of a message and $\{Id,\ link\}$ refers to the content of Hello message, where *Id* is the neighbor identity and *link* is the status of the link connecting the sender of the message and the node *Id*.

3.3.2 Inter-layer attack

In this attack, the malicious node modifies the default configuration of IEEE 802.11 MAC protocol, for example it denies the response to the RTS packet sent by its neighbors rather than sending a CTS packet after the SIFS period. When the CTS timeout expires, the sender of RTS infers that the malicious node didn't receive it correctly (i.e, a collision is occurred), as stated in [9]. Thus it retransmits the RTS after waiting for a new backoff time. After

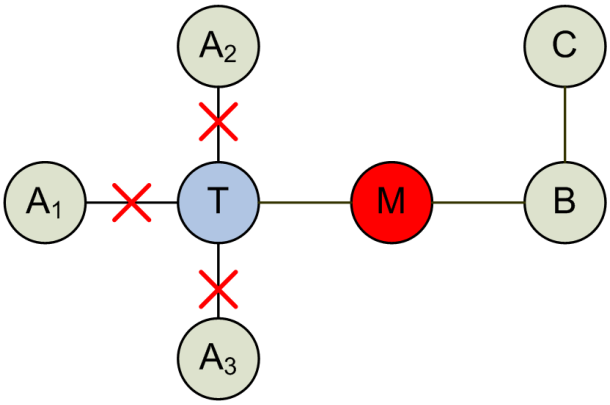


FIGURE 3.6: The network topology held by the nodes B and C after the attack, where the nodes A_1, A_2 and A_3 are unknown for them.

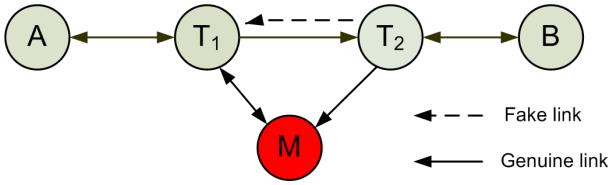


FIGURE 3.7: Fake symmetric link created between nodes T_1 and T_2

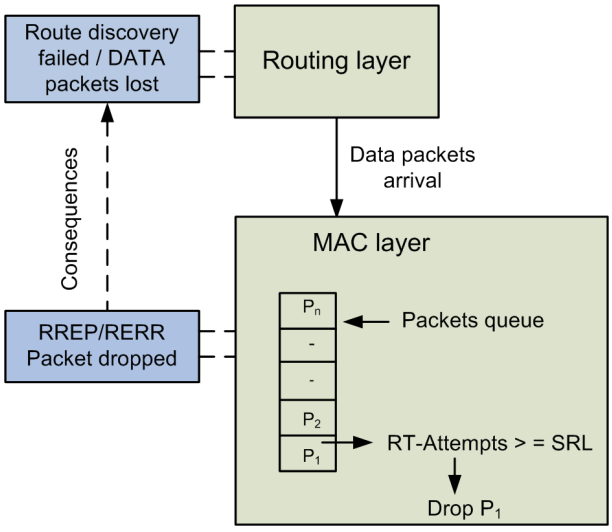


FIGURE 3.8: Inter-layer attack description

several retransmission attempts (RT-attempts), the sender of RTS abandons the transmission of the corresponding data frame whenever the number of attempts reaches the SRL (Short Retry Limit) as depicted in Figure 3.8. This attack may disrupt the route discovery process in reactive routing protocols, such as AODV, when the malicious node drops the RTS of the RREP packet, which leads to initiating a new route discovery. Moreover, this misbehavior can trigger a route maintenance process since the sender node will conclude that the link with the malicious node is broken. Consequently, the network performance degrades sharply.

		Approach		
		Symmetric-key cryptography	**Asymmetric cryptography**	**One way hash chain**
Comparison Criteria	**Speed**	Fast	Slow	Fast
	Scalability	Not scalable	scalable	scalable
	Computational overhead	Moderate	High	Lighweight
	Clock synchronization	No	No	Mandatory
	Storage capacity	Large	Large	O(log (n))[1]
	DoS Resiliency	Resilient	Not resilient	Resilient

TABLE 3.3: Cryptographic primitives comparison

3.4 Secure MANETs against Black hole attack

Recently, many investigations have been done in order to improve the security in MANETs, most of which are relied on cryptographic based techniques in order to guarantee some properties such as data integrity and availability. In what follows, we give a snapshot of the mostly used cryptographic primitives in MANETs.

3.4.1 Overview of the cryptographic primitives

As MANETs become more ubiquitous, the need for providing adequate security tools gets to be more obvious. The existing security schemes in such networks use generally one or more of the following cryptographic technologies: symmetric-key cryptography [21], digital signature [22], threshold cryptography [23] and one way hash chain [24]. Each of these cryptographic primitives has its specific advantages and drawbacks. For example, the security schemes based on digital signature and threshold cryptography generate much more computational overhead than those based on symmetric cryptography. However, the security approaches that are solely based on symmetric-key cryptography are less robust and offer less security than asymmetric key cryptography, due to the higher probability that the shared keys being compromised. As one way chains are known to be very efficient for verification, they became increasingly popular for designing security protocols for hand-held devices. This is due to the fact that the low-powered processors are able to compute a one way function within milliseconds, but would require tens of seconds or up to minutes to generate or verify a traditional digital signature [25], [26]. Consequently, recent wireless ad hoc network's security protocols extensively use one way chains to design protocols that scale down to resources constrained devices.

These cryptographic schemes are known to be efficient to ensure several properties such as confidentiality, data integrity and non repudiation. However, they cannot be adopted in MANETs since a Certificate Authority (CA) or a Key Distribution Center (KDC) are not always available. Moreover, these techniques cannot prevent a malicious node from dropping packets supposed to be relayed, which is our focus in this survey. In Table 3.3, we point out the main advantages and drawbacks of the cryptographic primitives presented above.

[1] n is the chain's length.

3.4.2 Taxonomy of the proposed solutions in the literature

There are basically three defense lines devised to protect MANETs against the packet dropping attack as illustrated in Figure 3.11. The first defense line (for prevention purposes) aims to forbid the malicious nodes from participating in packet forwarding function. Whenever the malicious node exceeds this barrier, a second defense line (for incentive purposes) is launched, which seeks to stimulate the cooperation among the router nodes via an economic model. Finally, once the two previous defense lines have been broken, a third one (for detection/reaction purposes) is launched aiming to reveal the identity of the malicious node and excludes it from the network.

3.4.2.1 First defense line schemes

Many researchers have been interested to develop several mechanisms to identify the malicious nodes that attempt to involve themselves in the routing path, and then take control over data/control packets. In the sequel, we give an overview of the major proposals which aim to recognize the malicious nodes at earlier stage of misbehaving before causing any damage to the network.

The authors of [27] have proposed a solution to cope with the black hole attack in AODV. First, they suggest to disable the ability of an intermediate node to send a RREP and allow only the final destination to do that. This technique avoids the black hole problem but increases the route establishment delay, especially in the case of large networks. Furthermore, since no authentication is used in RREP message a smart attacker can forge a RREP message on behalf of the legitimate destination (by spoofing its IP address). As such, this solution is inappropriate for coping with this attack. To overcome these shortcomings, they have proposed another solution which requires that the intermediate node adds its next hop's information to the RREP packet before sending it. On receiving this packet, the source node sends a special packet to the next hop of the intermediate node in order to verify that it has a route to the destination and also it is a neighbor of the intermediate node. This special packet contains a field dubbed check result which might be filled by the next hop node. When the source node receives the reply to this packet it extracts the check result information and decide accordingly whether this route is safe or not. If so, it sends out the data packets, otherwise it initiates a new route discovery or waits for subsequent RREPs. While this solution can avoid the black hole attack launched by a single node, it is unable to detect a collusive attack conducted by both of intermediate and next hop nodes. Moreover, its main disadvantage is the induced overhead if the check process is repeated for each intermediate node replying to the RREQ.

To ascertain the safety of the established path, a new scheme is proposed in [28] to secure AODV. This scheme can be briefly described as follows; once the normal path discovery procedure is finished, the source node sends special control packets to request each originator of RREP packet to send back its current neighbor set. On receiving more than one reply, the node starts comparing the received neighbor sets. If the difference between them is larger than a predefined threshold then a black hole attack is identified. To mitigate its impact, a cryptography-based reaction mechanism is designed, whereby the source node recognizes the true destination. Subsequently, a new control message is sent to the destination to establish the correct path. This method can reduce the likelihood of a successful black hole attack, but it cannot guarantee its prevention.

To secure OLSR against the colluding black hole attack, in which two malicious MPR nodes collude each other to prevent TC messages from being relayed correctly, a solution is proposed in [29]. This solution is based on a slight modification to the standard Hello message by adding the 2-hop neighbors set to the advertised set of one hop neighbors. Based on this information, any node can detect whether one of its neighbors has sent a false Hello message by searching any contradiction between the received neighbor sets. This solution can prevent the nodes that spoof links with non-neighboring nodes from being selected as MPR. However, the high mobility of nodes

can paralyze the network due to the huge number of the induced false alarms. Moreover, these contradictions can no more stand if the attacker spoofs links with far-away (more than two hops away) or not existing nodes.

TOGBAD approach was proposed in [30] to defend against colluding black hole attack in tactical MANETs, in which a successful attack can lead to human life loss. The proposed solution is designed to secure OLSR protocol, however it is suitable for any routing protocol based on Hello message exchange. Each network node extracts the neighbors list from the received Hello messages and sends it to the supervisor node. This latter, which is the only node running the TOGBAD scheme, uses the received information to construct the network topology graph. This graph is built based on the Cluster-Based Anomaly Detector (CBAD) introduced in [31] and [32]. Next, upon reception of a message from a node, the supervisor node extracts the number of neighbors claimed by the sender node and compares it with the size of this sender's neighbor set as calculated from the topology graph. If the difference between the claimed neighbors set and the one extracted from the graph exceeds a predefined threshold, then the supervisor concludes that this is an attempt to launch an attack and consequently an alarm is triggered. The extra messages sent by each node to the supervisor leads to an enormous control overhead increase in the network. Likewise, an excessive increase in computation overhead at the supervisor node is also observed. Therefore, this scheme is not suitable for MANETs due to the limited energy and computation resources of wireless nodes.

The herein described approaches aimed at attacks avoidance by means of preventing malicious nodes from being selected as part of the routing path of data packets. According to [33], the attacks can be avoided by prevention based mechanisms only if the applied techniques are perfect, which is hard to achieve in MANETs. Otherwise, someone will find out how to get around them; for example, in OLSR a malicious node can participate correctly to MPR selection phase however it fails to forward data packets when it is selected as MPR. In such case prevention techniques are useless. Besides, most of the attacks and vulnerabilities have been the result of evading the prevention mechanisms. Given this reality, detection and response are vital approaches for MANETs.

3.4.2.2 Second defense line schemes

As we have mentioned in section 3.2, a selfish node does not want to waste its resources for the benefit of other nodes. Hence, it refuses to forward other's packets but it still uses their services to communicate. To cope up with such behavior, one possible solution is to deprive the selfish node from the services provided by the rest of the network. Therefore, it will be obliged to cooperate. Otherwise it will be isolated from the network and never get its packets forwarded. This class of solutions is also referred to as *Incentive based schemes*.

One of the most reputable works in this category is the model introduced in [34]. This work proposes the use of a virtual currency, dubbed nuglets, as a payment currency in order to motivate each node to forward other's packets. Using nuglets, the authors have proposed two payment models: the Packet Purse Model (PPM) and the Packet Trade Model (PTM). In the former model, the packet sender loads some nuglets in the packet before sending it. The forwarder of this packet earns some nuglets as a payment for the service. If the quantity of nuglets in the packet reaches zero, then it is dropped. In the latter model, as opposed to the former one the packet's final destination rewards the intermediate nodes using its own nuglets. This model can be described as follows: each intermediate node earns some nuglets by buying a packet from its previous node for some nuglets and then selling it to the next node for more of nuglets, and the total cost will be paid by the destination. The main drawback of this technique is how to ensure that some nodes do not sell the same packet to more than one neighbor to earn extra money? And how to ensure that each receiver indeed has enough money to pay for the service?

To implement both of these models, each node is equipped with a tamper resistant security module that maintains the nuglets counter in order to prevent the nodes from illegitimate increase of their own nuglets.

Another sound work is the protocol SIP (Secure Incentive Protocol) proposed in [35]. In contrast to the previous schemes, SIP adopts a payment model in which node remuneration is accomplished by charging both source and destination nodes and rewarding the intermediate nodes. Moreover, the adopted model allows a node to transmit some extra packets when it has not enough credit for all the packets ready to be sent. The security of the payment process is achieved by dint of tamper-proof module embedded in each node. SIP is designed to work with any secure reactive protocol such as Ariadne [36] and ARAN [37]. The major weakness of this technique is the unfairness problem. The nodes situated in the network edges are less involved in the routing path due to their locations, therefore they cannot earn enough credit to send their packets.

3.4.2.3 Third defense line schemes

Most of the proposed solutions to handle packet droppers fit into this defense line. Hence, to conduct an in depth study we have classified them into five categories according to their basic ideas:

- *Passive feedback based schemes*: it encloses all the solutions whose the principle consists in overhearing the neighbor's transmission to check its legitimacy.
- *ACK-based schemes*: in this category, a node might request an acknowledgment from its succeeding neighbors to confirm the well reception of its packet.
- *Reputation-based schemes*: it represents the solutions that judge a node is malicious or well-behaved according to an assessment of its trustworthiness level which is computed based on several observations of its behavior.
- *Cross-layer cooperation based schemes*: this class illustrates the cooperation between two or more layers to either detect or enhance the detection accuracy of packet droppers. A given layer might make another layer aware of the beginning and the end of some operations or the values of some metrics in order to ensure better efficiency and accuracy.

A. Passive Feedback based schemes

Watchdog [38] is the first work that has dealt with the problem of nodes which agree to forward packets but never do so. It is designed to secure the DSR protocol and is based on the passive feedback technique, described as follows: (i) first, the watchdog node A transmits the packet (p) to its next hop B, as shown in the Figure 3.9. (ii) then it overhears the medium, using the promiscuous mode [2] to ensure that B has correctly forwarded the packet (p) towards C. If a misbehaving node is identified in the path towards the destination node, then a response mechanism dubbed Path-rater is launched. The goal of path-rater is to establish a new route that avoids the misbehaving nodes.

This scheme suffers from several weaknesses, as stated in [38]. Since a packet collision might occur and prevent the packet to reach the intended receiver, a forwarder node should not immediately be accused of misbehaving, but rather observed for a longer period to make an accurate decision. So, the detection of malicious nodes can take a long time. Moreover, power control transmission and collusion between group of nodes can trick the watchdog node. Finally, a malicious node can falsely accuse a legitimate node as misbehaving in order to exclude it from the network.

Many techniques have been proposed to enhance the robustness of Watchdog. Among them, the work presented in [39] which proposes to choose more than one Watchdog node to avoid the devastating impact of false

[2] When the promiscuous mode is enabled, it allows the node to capture all the frames sent in its vicinity regardless of their destination addresses, and then sends them to the higher layers for analysis purposes.

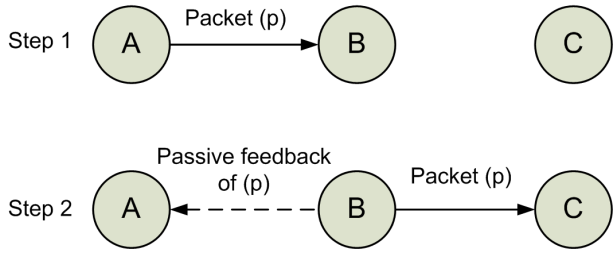


FIGURE 3.9: The principle of passive feedback

reports sent by the malicious nodes. To this end, the nodes are classified to ordinary, trusted and Watchdog nodes in terms of their trustworthiness. The trusted nodes are assumed to be the first nodes that initially form the network. The Watchdog nodes are selected periodically from the trusted nodes exclusively. On receiving the first reply for the route discovery process that has launched, the source node sends out in the secure Watchdog channel a special message to inform the Watchdog nodes about the ongoing transmission. Then, these nodes start monitoring the intermediate nodes connecting the source and destination nodes in order to report any misbehavior. This scheme can indeed detect and isolate the malicious nodes acting alone or in groups, however the induced overhead due to the new control messages is important.

In order to cope with the aforementioned problem of false reports, Ex-Watchdog is proposed in [40]. In this scheme, each node maintains a table containing information about all the paths it is involved in. Each entry of this table stores the following information: identifiers of the source and destination nodes, the identifier of the path connecting the source to the destination and finally the sum of all packets sent, forwarded or received through this path. Upon receiving a message reporting an intermediate node as malicious, the source node will not increase the failure tally of this node immediately as the Watchdog does. However, it sends out a special message to the destination node through an alternative path. This message contains the same fields as each entry in the table except that the path identifier is replaced by the malicious node's address. When the destination node receives this message, it checks first if there is a matching entry for the source and destination addresses in the table.
If so, then it compares the sum value received and the one kept in its table. If the two values match then the accused node is not malicious since all the packets sent by the source are received at the destination. In contrast, if the two values are different, then a reaction mechanism is triggered.

If no matching entry exists, then the reported node is malicious. As a result, a confirmation message is sent back to the source node. The absence of an alternative path to the destination makes the source unable to check the correctness of the report, and thus cannot recognize which node is malicious; the reporter or the reported.

B. ACK based schemes

To circumvent the limitations of the passive feedback based solutions, an explicit acknowledgment has been used by several schemes as a way to confirm the well reception of a packet by the far-away neighbors.

B.1. ACK-based schemes in reactive protocols

Two hop ACK based scheme is proposed in [41] to overcome the limitation of passive-feedback technique when power control transmission is used. To implement this scheme, an authentication mechanism is used to prevent the next hop from sending a forged ACK packet on behalf of the intended two hop neighbor. The main drawback of this scheme is the huge overhead. In order to reduce the overhead, the authors have proposed in [42] that each node asks its two hop neighbor to send back an ACK randomly rather than continuously. Likewise, this

extension also fails when the two hop neighbor refuses to send back an ACK. In such situation, the requester node is unable to distinguish who is the malicious node, its next hop or the requested node.

To overcome the previous ambiguity in determining the true malicious node, [43] focuses on detecting malicious links instead of malicious nodes. The authors propose the 2ACK scheme to detect malicious links and to mitigate their effects. This scheme is based on 2ACK packet that is assigned a fixed route of two hops in the opposite direction of the received data traffic's route. In this scheme, each packet's sender maintains the following parameters; (i) list of identifiers of data packets that have been sent out but have not been acknowledged yet, (ii) a counter of the forwarded data packets, (iii) and a counter of the missed packets. According to the value of the acknowledgement ratio (Rack), only a fraction of data packets will be acknowledged in order to reduce the incurred overhead. This technique overcomes some weaknesses of the Watchdog/pathrater such as: ambiguous collisions, receiver collision and power control transmission.

Both of the previous works remain vulnerable to the attacks launched by group of nodes. To counter these attacks, [44] provides a framework to mitigate the damage caused by the colluding black hole attack in AODV. The proposed technique has a moderate overhead induced by the ACK sent back by the destination during selected intervals of data transfer period. Throughout the data packets transmission, a flow of special packets is transmitted at random intervals along with the data. The reception of these special packets invokes the destination to send out an ACK through multiple paths. The ACK packets take multiple routes to reduce the probability that all ACKs being dropped by the malicious nodes, and also to account for possible loss due to broken routes or congestion in certain nodes. If the source node does not receive any ACK packet, then it becomes aware of the presence of attackers in the forwarding path. As a reaction, it broadcasts a list of suspected malicious nodes to isolate them from the network.

B.2. ACK-based schemes in proactive protocols

The authors of [45] have proposed a simple mechanism to detect the malicious nodes that drop TC packets in OLSR. To do so, these nodes spoof links with the target's two hop neighbors in order to gain an MPR position in the network. This approach requires that each node receiving TC message has to send an authenticated ACK back to the TC's source node. This requirement is carried out only if the receiver node is two hop neighbor of the TC's source node. In this scheme, each MPR node maintains a table containing its entire two hop neighbors set of link tuples and their corresponding trust values. During MPR selection phase, any node involved at least in one tuple whose the trust value equal to 0 should not be chosen as the unique MPR. Therefore, any misbehavior from a neighbor node can be easily detected.

We have proposed in [20] a three hops acknowledgment based scheme to cope with the cooperative black hole attack in OLSR. Our scheme adds two extra packets to OLSR, Hello-rep packet which is a slight modification to Hello message and a small acknowledgment packet. In this solution, each MPR node M acquires the list of its 3-hop neighbors reached through a distinct pairs of two consecutive MPR nodes (M1, M2), where M2 is the MPR node of M1 and this latter is the MPR of the node M. Afterwards, the node M selects one node, from this list, to which it requests an authenticated acknowledgment as a confirmation of the reception of the TC message that it has generated/forwarded. Notice that the authentication process is carried out using a pre-established secret key between node M and the requested node. If the number of missed acknowledgements overtakes a predefined threshold then the MPR nodes M1 and M2, relaying M and the requested node, are considered as malicious and consequently they will never be selected as MPR.

B.3. Requirements of ACK-based schemes

All the nodes running a solution based on acknowledgment need to maintain a timeout (To) value. This timeout represents an upper bound of the time that the sender node has to wait for the ACK to arrive. The determination of this timeout value is critical since a small value induces a large number of false accusations and a large value increases the memory required to store the outgoing packets for further comparisons. Figure 3.10 depicts an example of the lower bound of the timeout value maintained by node A for the reception of Two hop ACK from node C. The timeout value should be greater than the estimated threshold (Th) value which can be calculated as follows

$$Th = T_2 - T_1 \tag{3.1}$$

where T_1 and T_2 are the sending (reception) time of the packet (ACK), respectively. This threshold is estimated for a successful transmission at MAC layer without any retransmission, which is not a realistic assumption in MANETs, thus the timeout value should satisfy the following condition

$$To > Th + (AVG\_RT \times 1\_hop\_delay) \tag{3.2}$$

where AVG_RT is the average number of retransmissions of a packet at MAC layer, and 1_hop_delay is the one hop transmission delay which includes packet transmission delay, random backoff delay at the MAC layer and the processing delay.

C. Reputation based schemes

The reputation is the art of using historic observation about the behavior of a node to determine whether it is trustworthy or not. Each node must form an opinion regarding the other nodes based on their observed past behaviors. Then the nodes with low reputation are punished or avoided while establishing routes. The major drawback of this category is the excessive traffic exchange needed for sharing the reputation information between the nodes. Moreover, a serious vulnerability of reputation based schemes is the fact that any compromised node can send forged reputation information in order to decrease the trust level of some nodes. In what follows, we describe three representative schemes that use the reputation mechanism.

In [46], CONFIDANT protocol is introduced in order to secure source routing protocols against adversary nodes. This protocol aims to exclude the malicious nodes from participating to the route discovery phase and route the packets around them. The exclusion of these nodes is carried out using a dedicated reputation system. CONFIDANT consists mainly of the following elements: (i) the monitor, (ii) the reputation system, (iii) the trust manager, and (iv) the path manager. The role of the monitor is to ensure that each packet sent by a node is correctly forwarded by its next hop. This is achieved through the use of passive-feedback technique or by observing route protocol behavior. If an anomaly is detected, the node triggers an action via the reputation system. This latter manages a table containing the identifiers of all the known nodes and their corresponding rating. This rating is updated only if a sufficient evidence of misbehavior is acquired. In its turn, the trust manager is responsible for sending and receiving alarm messages that inform the nodes about the detected adversaries. Finally, the path manager is responsible for launching the adequate reaction and guarantees the establishment of safe routes.

CONFIDANT is suitable for small networks with low mobility; however it might be less efficient for large networks since each node needs to maintain a huge table for reputation purposes. Likewise, the high mobility of

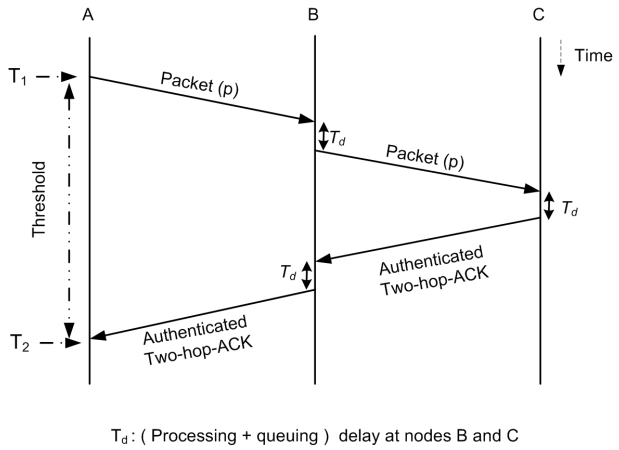


FIGURE 3.10: TWO hop ACK threshold for minimum timeout

nodes increases significantly the communication overhead. Additionally, this protocol inherits all the problems of passive-feedback based schemes since it uses this mechanism for the monitoring function.

Another scheme based on reputation system is the so called Friend and Foes that has been proposed in [47]. This scheme aims to prevent the selfish nodes from disrupting the network operations by refusing to participate correctly to the forwarding process. Its idea is inspired from the society principle which says that people agree to cooperate as long as they notice that there is a fair tasks distribution in the group. This scheme seeks to reward the cooperating nodes and punish the selfish nodes which refuse to cooperate. In this scheme, each node A advertises the set of nodes to whom it is not willing to forward packets along with the set of its friends. To do so, node A classifies the network's nodes in three sets, which are periodically updated, as described below. The friend nodes set for which a node accepts to relay the packets, the enemy nodes set for which no service is provided and the selfish nodes set which consists of nodes known to act as node A is an enemy. When node A sends a packet it searches for a route in which the next hop is a friend node and whenever it is requested to forward a packet it does so only if the requester node is a friend. The major drawback of this scheme is the large number of packets exchanged to advertise the friends and enemies sets.

A sound scheme is introduced in [48], in which the authors have proposed a new anomaly detection system dubbed RADAR to detect anomalous mesh nodes in Wireless Mesh Networks (WMNs) [2]. The salient features of RADAR can be summarized as follows: (i) reputation is used to evaluate each node's behavior by abstracting and examining the appropriate observations, e.g. data packets, a secure and dependable reputation management mechanism is then used to define, quantify and propagate the trust values of each node, ensuring the robustness and accuracy of the normal profiles feeding to detection engine; (ii) two light-weight anomaly detectors were employed to capture the node's behavior drifts in terms of reputation by exploring their temporal and spatial properties respectively, and they were seamlessly coupled to achieve higher detection accuracy and lower false positive rate. Notice that RADAR was specified and implemented with DSR routing protocol in order to detect misbehaving nodes that violate routing mechanisms at the network layer. It is found efficient in detecting nodes, involved in packet drop and spoofing attacks.

D. Cross-layer cooperation based schemes

Most of the existing solutions rely on the Watchdog technique to ensure the correct forwarding of packets by the neighboring nodes; however this technique suffers from certain weaknesses, particularly when power control is applied. In [49], the authors have proposed a low cost approach dubbed (SMDP) to circumvent the aforementioned drawbacks of Watchdog. They have designed a cross layer scheme that ensures higher detection accuracy. In this

scheme, it is required that the routing protocol be aware of the beginning and end of each continuous traffic routed through it. This can be accomplished through cross-layer cooperation between network and session layers.

At the end of each session, every node involved in the forwarding path sends out two signed packets, one to each successor node containing the number of packets sent to it, and the other packet towards its predecessor node contains the number of packets received from it. According to the received packets, each node broadcasts to its one hop neighbors a special packet called Forwarding Approval Packet (FPA) as a proof of its cooperation. On receiving this packet the neighbors of the sender can judge whether this node has correctly forwarded the packets or not. The main advantage of this scheme is its high detection accuracy that significantly reduces the number of false alarms.

3.4.2.4 Other schemes

In this section we give a brief overview on the major contributions which do not lie in any of the previous classes.

The authors of [50] have proposed two solutions to cope with the black hole attack in AODV. In the first solution, it is required that the source node waits until receiving more than two RREPs after each broadcasted RREQ (i.e. multi path routing). Upon reception of these messages the source node checks any appearance of shared nodes between the identified routes. If a shared node is identified then the source node sends the data packets to the destination through multiple routes using different packet IDs and sequence numbers. Otherwise, no packet will be sent. Notice that the appearance of shared nodes between different routes is not a sufficient condition to guarantee their safety since a malicious node might be involved in several routes. Moreover, this solution generates additional computational overhead due to the extra processed RREPs. Besides, if no shared node is identified then the source node delays or abandons the transmission of the data packets, leading to a severe degradation of the network performance.

To circumvent these drawbacks a second solution has been proposed. This new solution exploits the packet sequence number to detect the malicious nodes trying to hijack the traffic flow. To this end, each node maintains two extra tables containing the sequence numbers of the last packets sent (received) to (from) every node in the network, respectively. Upon reception of a RREP packet, the source node that has initiated the RREQ compares the sequence number extracted from the RREP and the one saved in its table. If they match then the safe route is identified, otherwise the responding node is deemed as malicious. This solution is faster than the previous one, however a malicious node can easily analyze the traffic passing in its vicinity and update its tables by the adequate packet sequence number, thereby it avoids the detection scheme.

Since the mobility of nodes is the most apparent feature of ad hoc networks, the conventional schemes based on static training data might not be efficient to deal with the black hole attack in such environment. [51] provides an alternative proposal that takes into account the rapidly changing topology of the network. This proposal uses the destination sequence number as a metric to detect any deviation from the normal network state. This state is updated dynamically at regular time intervals in order to enhance the detection accuracy. A special component named *discrimination module of anomaly detection* is used to distinguish the normal states from the abnormal ones. Its role is to measure the amount of deviation and compare it to a predefined threshold to find out whether an attack is occurred in the path toward the destination. To conclude, this scheme is effective to deal with the black hole attack in highly mobile networks however the update interval is a critical metric that should be assigned an appropriate value that ensures a better accuracy and performance.

In addition to the herein presented contributions, the reader may refer to the following papers to enrich its knowledge regarding the packet dropping attack [52], [53], [54], [55] and [56].

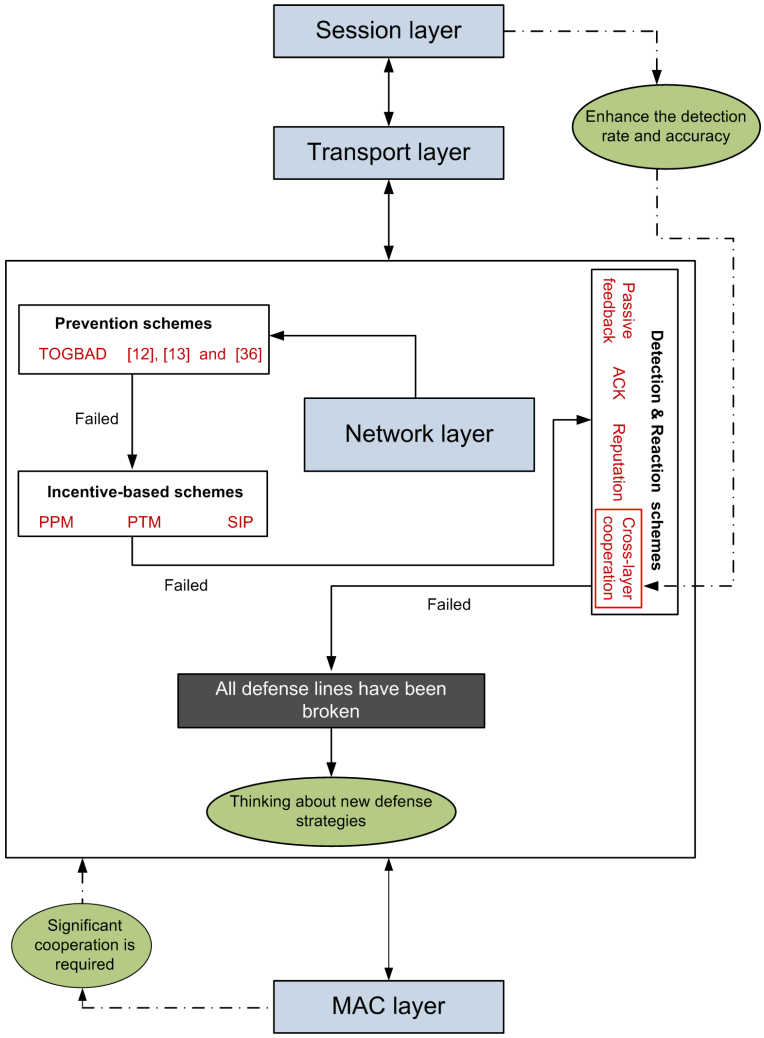


FIGURE 3.11: A holistic perspective on the defense lines against packet dropping attack

3.4.3 Discussion

As described in Figure 3.11, most of the solutions in different defense lines are routing layer dependent but cooperation with session layer would improve the detection rate as stated in [49]. Furthermore, since packets might be dropped due to MAC protocol rules as illustrated in section 3.2, an additional diagnostic provided by MAC layer remains a key component for a robust detection scheme. This cooperation may significantly reduce the false alarms by discerning normal behavior from the malicious one (i.e. the inter-layer attack described in section 3.3.2).

A summary of the characteristics of the surveyed schemes is presented in Table 3.4. In this table, we emphasize the most prominent features of each scheme in terms of its robustness, scalability, induced overhead and the reaction mechanism adopted to exclude the detected attackers. Moreover, this table allows us to identify the strong and weak points of each scheme in order to develop an eventual hybrid solution that merges two or more schemes, from different defense lines, together to ensure a perfect protection against the packet droppers. The features of each scheme are highlighted based on the following metrics:

- The defense line to which the scheme belongs.

- Its robustness against the collusive black hole attack, in which two or more nodes collude to launch the attack.
- The additional overhead generated by the scheme in terms of the new packets sent and the extra computations required to carry out the scheme.
- The impact of the scheme on routing protocol's performance such as end-to-end delay and packet delivery ratio.
- Is the scheme providing any reaction technique to penalize the detected attackers?
- Is the scheme scalable to large networks? i.e. whether the scheme maintains its efficiency when the network becomes larger and dense.
- The architecture of the scheme: centralized, distributed or stand-alone; defined as follows:

  Centralized: the core part of the scheme is running on an unique supervisor node which monitors the whole network and the rest of nodes need to report to the supervisor node for information processing.
  Distributed: all the nodes run the same scheme and exchange information between each other.
  Stand-alone: similarly, each node runs the same scheme however the communication between nodes is not necessary.

Another summary of the main assumptions and limitations of each class of the schemes studied throughout this chapter is provided in Table 3.5. As we can see from this table all these approaches are built on a set of assumptions that are either unrealistic or hard to achieve in a hostile environment like MANETs. Hence these assumptions limit the applicability of these approaches to some specific network configurations and constitute their major drawbacks.

3.5 Challenges

As discussed in the previous sections, most of the proposed solutions are built on a number of assumptions which are either hard to realize in a hostile and energy constrained environment like MANETs or not always available due to the network deployment constraints. Moreover, these solutions are generally unable to launch a global response system whenever a malicious node is identified. In contrast, they either punish the malicious node locally without informing the rest of the network or divulge its identity to the network through costly cryptographic computations. Moreover, even though the malicious node is punished in a part of the network it can move to another part and continues causing damage to the network until it is detected again. Due to these reasons, many challenges have to be carefully considered in order to design a robust solution to cope with the packet dropping attack. These challenges can be summarized as follows. First, the attackers' behaviors are tailored to the specific routing protocol, making it impossible to build a general model for characterizing the attacker. Secondly, how to use this model to achieve a high-level resistance against these attacks while maintaining network performance. Recently, most of the proposed solutions are focused on adding new components to the original protocol to assess the deviation of the neighboring nodes and monitor their behaviors. However the use of these additional components might remove an important performance optimization. A simple way to secure MANETs against the increasing threat of the packet droppers without affecting their performance is to take into account the security metric at an earlier stage of the design process of routing protocols. This new design process could be similar to the co-design technique used for developing the embedded systems. A complementary way to achieve the best trade-off between security and

performance is to aggregate the three defense lines discussed in this chapter to guarantee the cooperation of nodes in the network.

3.6 Conclusion

In this chapter, we have presented a survey of the main existing works that deal with packet dropping attack in MANETs. The attack schemes, as well as prevention, detection and reaction mechanisms have been explored. We categorized them into three categories according to their goals and their specific strategies. A comparative study between them was then conducted to highlight their respective effectiveness and limitations. We concluded that most of the proposed schemes in the first, second or third defense line are based upon certain assumptions that are not always valid due to the dynamic nature of MANETs and their specific characteristics. Many researchers have been motivated to apply game theory to enforce nodes cooperation in MANETs, such as the works done in [57] and [58], by examining its similarities with the social behavior of human in a community. These works assume that a node tries always to maximize its benefit by choosing whether to cooperate in the network or not. However, those works are generally based on the assumption that the majority of the nodes are misbehaving, which is not an usual case in MANETs. We believe it is an interesting and significant topic for further exploration with more realistic assumptions, especially tailored for packet dropping attack. In next chapter, we will focus on OLSR and presents two solutions to defend against two types of black hole attack that may target it, as described in section 3.3.1.2.

	Characteristics							
	Defence line	**Deal with collusive attacks**	**Computation overhead**	**Communication overhead**	**Latency**	**Scalable**	**Reaction scheme**	**Architecture**
Watchdog [38]	3^{rd}	N	Low	N	N	Y	N	Distributed
CONFIDANT [46]	3^{rd}	N	Low	Low	N	Y	Y	Distributed
H.Deng 1[27]	1^{st}	N/A	N	N	N	Y	N	Distributed
H.Deng 2 [27]	1^{st}	N	Low	Low	Y	Y	N	Distributed
B. Sun [28]	1^{st}	N/A	Low	Medium	High	Y	N	Distributed
TOGBAD [30]	1^{st}	N/A	Very high at the supervisor node	N	N	Y	N	Centralized
Friend & Foes [47]	3^{rd}	N/A	Low	High	High	N	Y	Distributed
Al-Shurman 1 [50]	N/A	N	Low	Medium	High	N	N	Stand-alone
Al-Shurman 2 [50]	N/A	N/A	Low	N	Low	N	N	Stand-alone
Nuglets (PPM) [34]	2^{nd}	N/A	Low	N	N	N	Y	Stand-alone
Nuglets (PTM) [34]	2^{nd}	N/A	Low	N	Low	N	Y	Stand-alone
SIP [35]	2^{nd}	N/A	Low	low	Low	Y	Y	Stand-alone
SA-OLSR [45]	3^{rd}	N/A	Low	High	N	Y	N	Stand-alone
B. Kan [29]	1^{st}	N/A	Low	N	N	Y	N	Stand-alone
N. Nasser [40]	3^{rd}	N/A	Low	Low	N	N	N	Distributed
SMDP [49]	3^{rd}	N	Medium	Medium	N	N	N	Distributed
K. liu [43]	3^{rd}	N	Low	Low	N	Y	N	Stand-alone
S. Ram [44]	3^{rd}	Y	Low	Low	N	Y	Y	Distributed
2-hop Ack [41]	3^{rd}	N	High	High	N	N	N	Stand-alone
Random 2-hop Ack [42]	3^{rd}	N	Low	Low	N	Y	N	Distributed
3-hop Ack [20]	3^{rd}	Y	Medium	Low	N	Y	N	Distributed
RADAR [48]	3^{rd}	N	Low	Low	N	Y	Y	Distributed

TABLE 3.4: Characteristics of the surveyed schemes

	Main assumptions	**Limitations**
Passive Feedback based schemes	Promiscuous mode operation is mandatory No collusion amongst nodes	Inherits all the Watchdog's drawbacks
ACK-based schemes	Authentication mechanism is deployed The requested node always sends back the intended Acknowledgement	Huge overhead generated due to the extra Acknowledgment packets sent. Decision ambiguity if the requested node refuse to send back an Acknowledgment.
Reputation-based schemes	Preestablished list of friend (trusted) nodes	Overhead induced in sharing reputation information amongst the nodes
Incentive-based schemes	Tamper resistant hardware is mandatory for the operations of this category of solutions.	Legitimate nodes might be punished indirectly due to their location in the network. A node can sell the same packet several times to earn more money.

TABLE 3.5: A comparison on the different approaches: assumptions and drawbacks

Chapter 4

Coping with Black Hole Attack In MANETs

In this chapter, we will describe two types of black hole attack and provide two solutions to cope with each of them. The first attack is the colluding black hole attack, in which both of the attacker nodes act at routing layer to launch the attack. In the second attack, one attacker node acts at routing layer, by exploiting OLSR's vulnerabilities, whereas its collaborator node acts at MAC layer by falsely validating the reception of MAC frames sent by the victim node.

4.1 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 Mobile Ad hoc Networks (MANETs), in which mobile nodes, within the transmission range of each others, can communicate directly over the wireless link, while those that are far apart use other nodes as relays. The properties of MANETs, such as shared wireless medium, open network architecture, stringent resource constraints and rapidly changing topology make them vulnerable to a bunch of attacks at different layers, especially at routing and MAC layers in which these attacks are launched easily. Therefore, the task of securing such network remains hard and necessitates careful investigation.

To address the routing problems with the intrinsic features of MANETs, numerous protocols have been developed and further standardized by the IETF (Internet Engineering Task Force), among which the Optimized Link State Routing (OLSR) protocol [16]. The major drawback of OLSR, as all other ad hoc protocols, is that it has not been designed and thought with respect to security issues. Hence it is exposed to many types of misuse leading to a dramatic drop of the network performance and services. Additionally, any node can misbehave and try to disrupt the routing process by injecting tampered or even fake information in the network. Notice that the lack of security considerations in the design of these routing protocols has penalized the neighbor discovery function since it becomes easy to spoof any identity/link of/with any node and disseminate false topology information to the whole network.

In this chapter, we will address two scenarios of black hole attack. In the former scenario, the attack is launched at routing layer by two colluding nodes and targets the topology control (TC) packets. To cope with this attack, we propose a three hops acknowledgement based scheme that requires slight modification to Hello and TC messages. This scheme adds two extra packets to OLSR, Hello-rep packet which is a slight modification to Hello message and a small acknowledgment packet. In this scheme, each MPR node M acquires the list of its 3-hops neighbors reached through a distinct pair of consecutive MPR nodes (M1, M2), where M2 is the MPR node of M1. Afterwards, the node M selects one node from this list to which it requests an authenticated acknowledgment. This acknowledgment aims to confirm the reception of the TC message generated/forwarded by M. Notice that the authentication process is carried out using a pre-established secret key between the node M and the requested node. If the number of missed acknowledgements overtakes a predefined threshold then the MPR nodes on this path are considered as malicious and consequently will never be selected as MPR. Moreover, no further packet will be forwarded for these detected nodes.

In the latter attack scenario, the attackers exploit vulnerabilities of OLSR to disrupt the neighbor discovery function. They, first, launch the attack at the routing layer by implementing a virtual link attack (VLINK), leading to establishment of false symmetric link between the target nodes that are connected through an asymmetric link. So, an incorrect MPR (Multi-Point Relay) set may be elected by the target nodes as well as by their neighbors leading to selection of broken routes to forward data packets. Subsequently, a false validation attack is initiated at MAC layer in order to reinforce the VLINK attack and make it more destructive. To counter this attack, we propose a cross-layer solution in which the routing layer needs to get a confirmation from MAC layer regarding the status of a specific link before advertising it to the network. In order to check the symmetry of a link, the RTS (Request to Send) and CTS (Clear to Send) frames format is modified to prevent the attacker node acting at MAC layer from falsely validating the well reception of the RTS and DATA frames being transmitted by one of the target nodes. Notice that this scheme can only prevent the attack but cannot identify the attackers.

4.2 OLSR and its vulnerabilities

The Optimized Link State Routing protocol (OLSR) [16] is a proactive routing protocol designed to work in large and dense networks. The main optimization of this protocol is achieved through the use of MPRs (Multi-Point Relay). The MPRs of a node are a subset of its one hop neighbors through which it can reach all its two hop neighbors. These MPRs are the unique responsible for generating and spreading the partial link state information through TC messages, thus reducing the induced overhead. In OLSR, the MPR selection process depends on the 'Willingness' value obtained from Hello message. This value indicates the readiness of a node, based on its own resources, to forward packets of other nodes. The higher the willingness is, the higher the priority the node will have to be selected as MPR. The value $Will\_never$ is chosen by nodes that are not willing to participate in the routing process. However, the value $Will\_always$ is reserved for nodes which are candidate to be selected as MPRs.

Control traffic in OLSR is exchanged through two different types of messages, namely Hello and TC messages. Hello messages are broadcasted periodically (every 2 sec) by every node to discover its neighborhood and advertise its MPR selection. The TC message is sent periodically (every 5 sec) to the whole network by each MPR node to declare its MPR selectors set. The information contained in TC messages is used to construct the routing tables at each node in the network. Generally, in proactive routing protocols, each node has two tasks to accomplish: (i) correctly generate control traffic in order to provide correct and concise information to other nodes, and (ii) accept to relay the routing traffic on behalf of its neighbors. Thus any attack targeting such protocols should focus

on the disruption of these tasks in order to provoke harmful decrease of the network performance. To do so, a misbehaving node may carry out one of the following misbehavior techniques.

- **False routing messages injection**: in which a malicious node generates regular control messages containing false or incomplete information regarding the current state of the network.
- **Refuse of control traffic generation/relay**: in this case, a malicious node refuses to generate its own control traffic, such as TC messages, or refuses to forward control packets of its neighbors (as it is expected).
- **Modification of routing control traffic**: in which a malicious node forwards packets on behalf of other nodes, however it modifies their contents by inserting wrong information or deleting some valuable information.
- **Replay attacks**: in such attack, a malicious node listens to the control traffic transmissions on its neighborhood or saves the received routing packets and later on injects possibly wrong and outdated information in the network.

4.3 Mono layer black hole attack

In this case, the black hole attack is exclusively carried out at routing layer.

4.3.1 The single black hole problem

In OLSR, a node without empty MPR selectors set should generate a TC message which will be relayed by its MPR nodes. A malicious node that advertises itself as having the highest value of willingness will be always selected as MPR by all its neighbors. Furthermore, due to the lack of security countermeasures in OLSR, this malicious node can launch a black hole attack by dropping the intercepted TC messages, which leads to the loss of the critical topological information. A simple solution for this attack is the use overhearing technique, but it has a number of shortcomings, as explained below.

In wireless ad hoc networks, node A can hear the transmission of its MPR due to the nature of the broadcast channel. By hearing its own MPR's transmission, the node A can check whether this MPR has relayed the TC message that it has sent/forwarded. Figure 4.1 shows a scenario in which the node A broadcasts its TC message which will be relayed by its MPR nodes, B and C. According to OLSR specifications, node D, as MPR of both of nodes B and C, rebroadcasts only the first TC message received, from its MPR selectors set, and ignores the next ones (which contain the same originator address and sequence number). In order to check whether its MPR node forwards the TC message sent, the node A overhears its MPR node's transmission and then accuses it as malicious or not accordingly. As shown in the Figure4.1, node C accuses node D as malicious whereas it is not.

4.3.2 Colluding Black hole attack model

In this section, we describe how two adjacent malicious nodes can launch a black hole attack in wireless ad hoc network. To facilitate the understanding of the illustration, we summarize bellow the notations and assumptions used throughout this chapter. We assume that the links between vehicles are often symmetric and each node holds a pair of public/private key in order to use the digital signature to authenticate the acknowledgement packet. Additionally, the following notations are also used to illustrate this attack in OLSR.

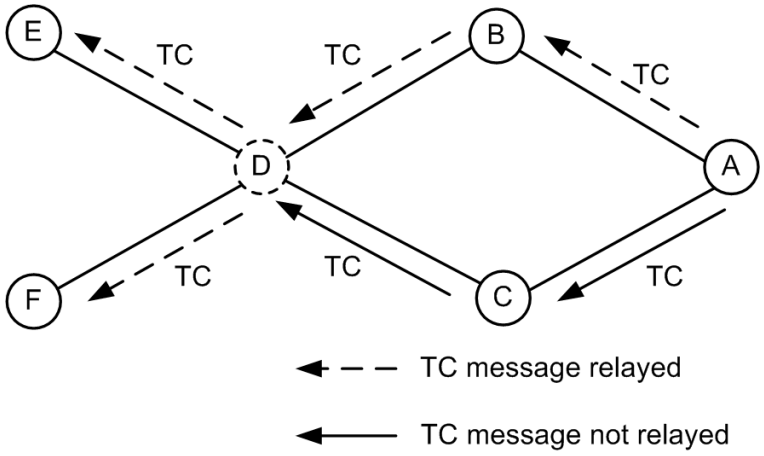


FIGURE 4.1: Shortcoming of the overhearing technique

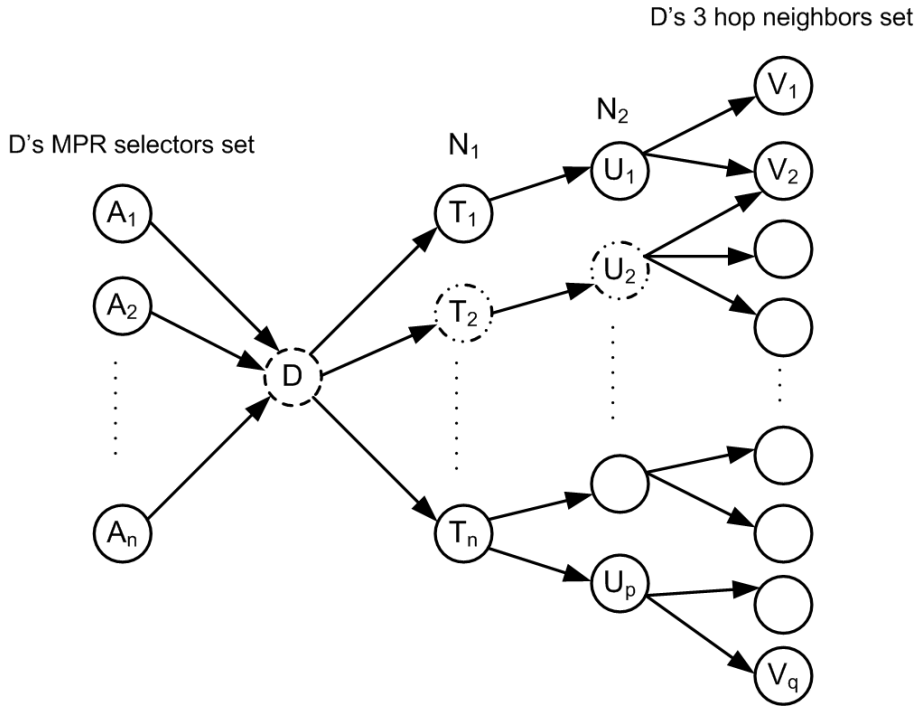


FIGURE 4.2: Colluding black hole attack model

- N_1: the MPR set of node D.
- N_2: the MPR set of N_1's nodes, which are not in D's one hop neighborhood. This can be expressed as follows

$$\forall n \ (n \in N_2 \Rightarrow \exists m \ (m \in N_1 \ \wedge T)) \tag{4.1}$$

where

$$T \equiv n \in MPR\_set\,(m) \wedge n \notin neigh\,(D)$$

Such that neigh (D) refers to the set of 1-hop neighbors of D.

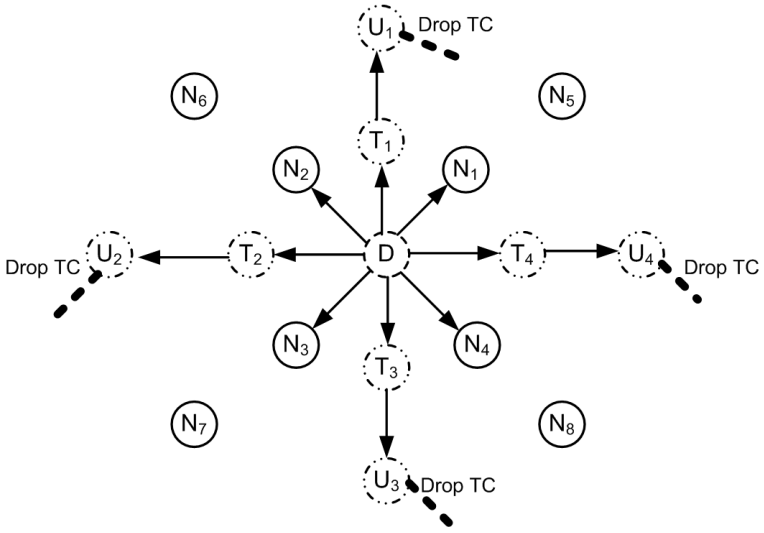


FIGURE 4.3: Multiple attackers around the victim node

- Sym_i: a symmetric 3- hop neighbor of D, which will send an $3hop\_ACK$ packet upon reception of D's TC message.
- S_j: a subset of the symmetric 3-hop neighbors set of D, which is connected to the node $j \in N_2$.

In order to launch the black hole attack in OLSR, a malicious node can force its election as MPR by setting constantly its Willingness field to $Will\_always$ in every HELLO packet. According to OLSR rules, its neighbors will always select it as MPR. Using this technique, a malicious node can easily get, as an MPR, a privileged position in the network. It can then exploit this position to launch denial of service attacks. In a more sophisticated way, two colluding MPR nodes m_1 and m_2 can launch a more destructive attack if the node m_2 drops all TC messages forwarded by node m_1. The victim node, which is an MPR selectors of m_1, cannot detect this misbehavior because node m_2 is out of its radio range and m_1 does not inform it.

Figure 4.2 shows an illustrative description of this colluding black hole attack. Let the nodes $\{A_1,..., A_m\}$ be a set of target nodes and T_2, U_2 the attacker nodes, $\{T_1,..., T_n\}$ the set of D's MPR nodes, $\{U_1,..., U_p\}$ is the subset of D's 2-hop neighbors which constitutes the N_1's MPR nodes and $\{V_1,..., V_q\}$ the set of D's 3-hop neighbors. The attack is launched as follows; node T_2 broadcasts its HELLO message with the value of $Willingness$ field as $Will\_always$. Then, all its 1-hop neighbors will choose it as an MPR node. Afterwards, it chooses the node U_2 as its unique MPR. Thereby, the node U_2 can perform the following misuses without being detected by node D or any other node in its neighborhood.

- Drop all TC messages of node D, which contain the IP addresses of nodes $\{A_1... A_m\}$. Thus, the nodes which are exclusively connected to U_2 cannot communicate with the MPR selectors of D.
- Drop TC messages that advertises the address of D to prevent link information of D from being disseminated to the network. Moreover, it can also drop all TC messages passed through it.
- Modify the content of TC messages generated by node D, which leads to dissemination of an inconsistent topological information.

The consequences of this attack are devastating if multiple colluding attackers exist around the victim node, as shown in Figure 4.3. The sender/forwarder of TC message can overhear its MPR nodes transmissions to ensure

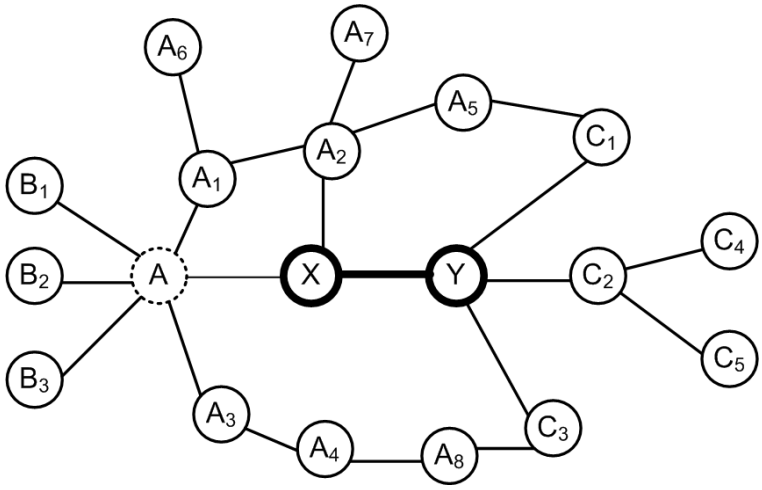


FIGURE 4.4: Colluding black hole attack description

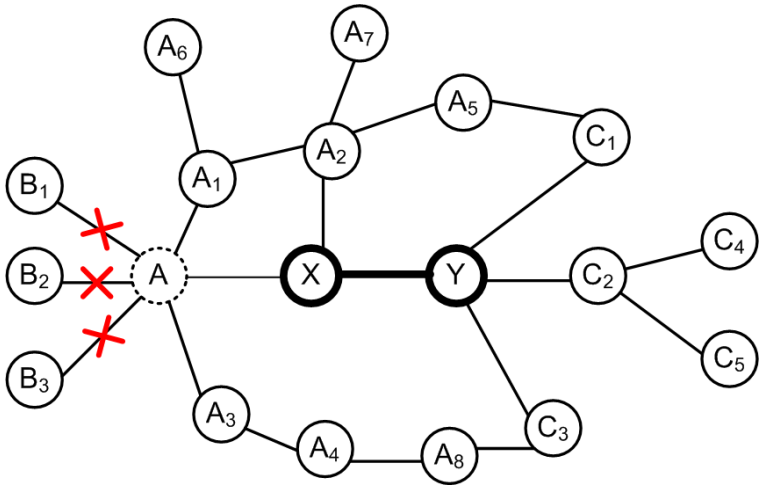


FIGURE 4.5: Topology perceived by nodes C_2, C_4 and C_5 after attack

that its TC message is relayed correctly, but while the TC dropping happens out of its transmission range then this technique fails to detect the attack.

Figure 4.4 gives an example of this attack. In this example, the nodes B_1, B_2 and B_3 constitute the MPR selectors set of node A. The first attacker X advertises itself as having sufficient resources to forward packets of other nodes by setting its willingness field to the highest allowed value (i.e. the value 7). According to OLSR specifications, X will be chosen as A's MPR. Afterwards, X chooses the second attacker Y as its unique MPR node. Thus, all TC packets generated by node A and relayed by X will be destroyed by Y. The consequences of this attack are illustrated in Figure 4.5, where nodes C_2, C_4 and C_5 cannot find a route towards D's MPR selectors because the D's TC messages have never been received (i.e. the topological information held by these nodes is incomplete).

4.3.3 Our proposed solution

In order to deal with the colluding black hole attack, we propose a new acknowledgment based scheme to mitigate its devastating impact. The functioning of this scheme can be summarized as follows. In addition to the basic control messages of OLSR, which are Hello and TC messages, our scheme introduces another two control packets,

MPR_node	$2-hop\ neighbors$
Y	$C_1\ C_2\ C_3$
A_2	$A_1\ A_5\ A_7$

TABLE 4.1: Example of $HELLO\_rep$ message sent by node X

named $3hop\_ACK$ and $HELLO\_rep$. The $3hop\_ACK$ packet is used by a node to acknowledge the reception of a TC message sent by its 3-hop neighbor, whereas the $HELLO\_rep$ packet is broadcasted by a node to advertise its 2-hop neighbors set upon request. For the request, we use one of the unused bits in HELLO message to indicate whether the receiver MPR nodes should generate $HELLO\_rep$ packet or not. Table 4.1 shows an example of $HELLO\_rep$ message.

Our scheme requires that each MPR node knows its 3-hop neighbors set in order to be able to verify whether a malicious node, out of its transmission range, has misused the transmitted TC messages. Our scheme requires also that each MPR node has to forward all the TC messages coming from its MPR selectors even if the same TC message has been already received. Thus, our scheme can distinguish whether a TC message is intentionally dropped by a malicious node or by a legitimate node due to duplication of two TC messages. To detect these malicious nodes, the sender or forwarder of a TC message maintains a list of TC packets' identifiers (IDs) that have not been acknowledged yet. Similarly, the following parameters need to be maintained also.

- ACK_{miss}: Counter of the number of $3hop\_ACK$ missed on the link $M_1 \longleftrightarrow M_2$ such that $M_1 \in N_1$ and $M_2 \in N_2$.
- TC_{drop}: Counter of the number of TC dropped by the direct MPR node.
- STN (Symmetric Three hop Neighbors): List of the nodes which should send a $3hop\_ACK$ for each pair of nodes $<M_1, M_2>$.
- δ_1 and δ_2: thresholds for the number of TC packets dropped and $3hop\_ACK$ packets missed, respectively, such that $\delta_1 < \delta_2$.
- $BlackList$: List of the detected malicious nodes.

When a node A sends or relies a TC message, it monitors its MPR nodes' transmissions to check whether they relay this TC packet or not. If A doesn't overhear the retransmission of its MPR node B, it then increases its TC_{drop}. If the TC_{drop} of B exceeds δ_1, then A adds B's identity to the $BlackList$ and recalculates the MPR set excluding the node B. Another more complicated case is when the node B forwards the TC message sent by A, but it doesn't punish its MPR node C which drops this TC message. To cope with such misbehavior, the node A increases the ACK_{miss} counter for the link $B-C$ whenever the timer, which is set up to limit the waiting delay for reception of $3hop\_ACK$ packets over this link, exceeds the predefined threshold. If the node A receives a Hello message from B in which C is not in B's MPR set, while ACK_{miss} does not exceed δ_2, it then adds C to its $Suspicious_{list}$; otherwise, it concludes that both of the nodes B and C are malicious and adds them to its $BlackList$.

$Suspicious_{list}$ is used by each MPR node to record the IDs of nodes that may have dropped the $3hop\_ACK$ packets, but have moved out of the current forwarding path of TC messages before their Ack_{miss} counter exceeds the threshold. As the network topology changes rapidly, those suspicious nodes can gain again the same position

Algorithm 1 HELLO reception

```
if (orig_adr ∉ BlackList) then
    if (Id = 1) then
        if (orig_adr ∈ MPR_sel_set) then
            prepare HELLO_rep pkt;
            send HELLO_rep pkt;
        end
    end
    process HELLO msg;
end
```

Algorithm 2 TC reception

```
if ((orig_adr, PSN)∉ duplicate_set) then
    if (my_Id ∈ Req_ack_list) then
        prepare 3hop_ack_pkt;
        send 3hop_ack_pkt;
    end
    process TC;
end
if (sender_addr ∈ MPR_selc_set) then
    for (each Request_ack_list) do
        TTLreq_i - - ;
        if (TTLreq_i == 0) then
            delete Request_ack_list_i;
        end
    end
    add my own Request_ack_list to TC_pkt;
    forward TC_pkt;
end
```

in the forwarding path of TC messages sent/forwarded by the node A. A then accumulates the new observations with the previous ones to ensure more accuracy. Once a Hello or TC messages are received, a node follows the steps described in algorithms 1 and 2, respectively.

In our scheme, each MPR node follows the two steps described below.

1. Get the list of its 3-hop neighbors.
   - For each node i of the set N_1, the node D asks it to send its MPR set along with its 2-hop neighbors reached through these MPRs.
   - Upon reception of $HELLO$ message in which $HELLO\_id$ is set to 1, each node of the N_1 replies with a $HELLO\_Rep$ packet containing the requested information.
   - Each MPR node sets the $HELLO\_id$ field to 1 whenever it detects a change in its MPR set or its N_2 set.
2. Add a request of $3hop\_ACK$ packet to TC messages.
   - For each node of the set N_2, the node D chooses a neighbor node such that:

$$\left.\begin{array}{c} \forall i \;\; sym_i \;\; \notin \bigcup\limits_{\substack{j=1,\ldots,p \\ k=1,\ldots,p}}^{j\neq k} \left(S_j \cap S_k \right) \\ p \;=\; \|N_2\| \end{array}\right\} \quad (c)$$

- When the node D generates/forwards a TC message, it inserts a request for $3hop\_ACK$ to all the nodes of the set X, before sending it out.

$$X = \{Sym_1, ..., Sym_p\}$$

To implement our scheme in OLSR, a slight modification to TC packet is required. The new format of the TC message is shown on Figure 4.6, in which the identifier list of the requested 3-hop neighbors is inserted. To prevent the increasing size of the TC message we have used a special TTL field. Its initial value is set to 3 in order to force the MPR nodes to delete this list after three hops.

- Since future vehicles in VANETs can be equipped with rechargeable source of energy, and extensive on-board storage capacities, then the processing power and storage efficiency are not a big issue as it is in MANETs. Therefore, to ensure the identity of the sender of $3hop\_ACK$, we may use public key cryptography in which each vehicle signs its outgoing $3hop\_ACK$ packets using its private key and authenticates the incoming $3hop\_ACK$ using other's public key. Notice that the aim of this authentication is to prevent malicious vehicles from sending a false $3hop\_ACK$ packet on behalf of the legitimate 3-hop neighbors.

4.3.3.1 Discussion

We need to prove that for each pair of nodes $<x, y>$ where $x \in N_1$ and $y \in N_2$, there is at least one node that satisfies the condition (c).
Let us suppose that

$$\bigcup_{\substack{i=1,..p \\ j=1,..p}}^{i \neq j} (\, S_i \cap S_j \,) = \mathrm{Z} \tag{4.2}$$

Such that Z represents the set of N_1's 2-hop neighbors that are reached through more than one MPR node. We have to prove now that $\forall\, i\ S_i \not\subset Z$. If we suppose that $S_l \subset Z$, then we get

$$\forall\, z(\ z \in S_l \Rightarrow \exists\, r\, z \in S_r \wedge l \neq r) \tag{4.3}$$

According to the MPR computation heuristic [16], the above formula indicates that the node l is not MPR, which is a contradiction because $l \in N_2$. Therefore $\forall\, i\ \ S_i \not\subset Z$ is proved. Consequently, the node D can monitor the forwarding path of each pair of nodes $<x \in N_1, y \in N_2>$.

In the case where two MPR nodes $M_1 \in N_1$ and $M_2 \in N_1$ have chosen the same MPR node $M_3 \in N_2$, their MPR selectors node A will validate the correct forwarding of both links $< M_1, M_3 >$ and $< M_2, M_3 >$ upon reception of one $3hop\_ACK$. This is due to the fact that when the node M_3 relays the first TC forwarded by M_1 and drops the second TC forwarded by M_2, this behavior doesn't disrupt the protocol functioning since all the neighbors of M_3 have already received the same link state information contained in the dropped TC message.

In Figure 4.7, the node D may choose the node C_3, where $C_3 \in (S_{Y_1} \cap S_{Y_2})$, to send back a $3hop\_ACK$. In this case, node C_3 can be reached through both of Y_1 and Y_2 . We suppose that Y_2 is an honest node while Y_1 is a malicious node. Since D maintains a list of the nodes that should send $3hop\_ACK$, then when the node C_3 receives a TC message forwarded by Y_2, it sends a $3hop\_ACK$ to D. Upon reception of this packet by D, it deletes C_3 from the STN list. Thus, the node D believes now that both of the nodes Y_1 and Y_2 have relayed its

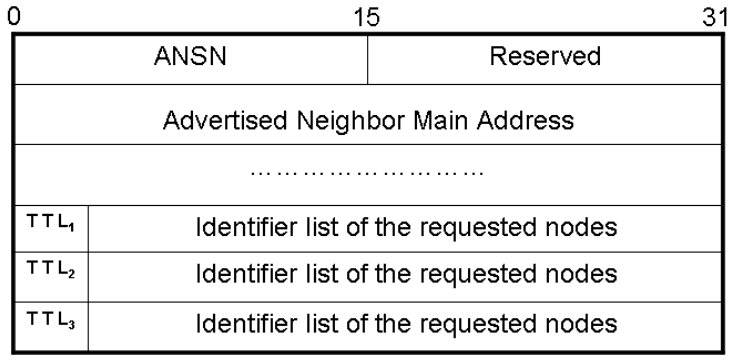


FIGURE 4.6: New format of the TC message

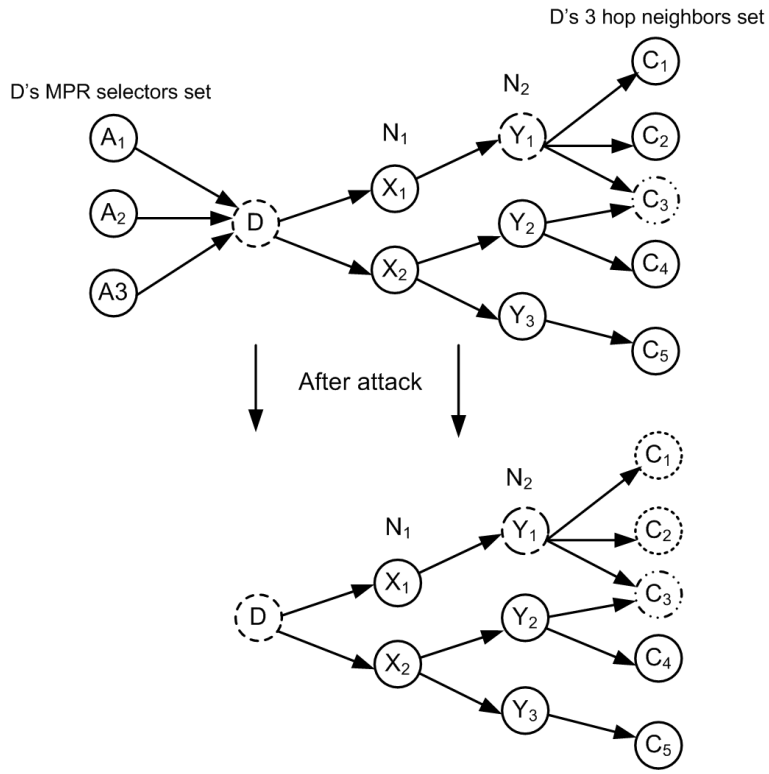


FIGURE 4.7: Topology perceived by nodes C_1 and C_2 when the condition c is not satisfied

TC message, but actually the node Y_1 did not do so. Even if the node Y_1 relies this TC message, the node C_3 sends back one $3hop\_ACK$ because it processes only the first received TC message and ignores the later one that has the same *originator address* and *MSN*(Message Sequence Number).

4.3.3.2 Timeout for acknowledgement reception

The parameter timeout, Δ, is used to set up a timer for $3hop\_ACK$ packet reception. If the timer expires before the expected $3hop\_ACK$ packet is received, the ACK_{miss} counter associated to the pair of nodes

$$< M_1 \in N_1, M_2 \in N_2 >$$

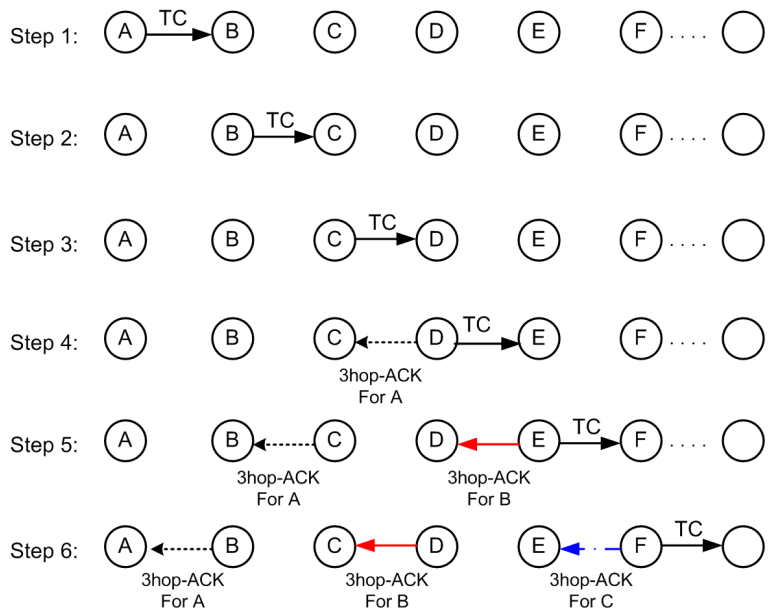


FIGURE 4.8: The $3hop\_ACK$ scheme functioning

will be incremented. Thus, an appropriate value of Δ is important for the successful operation of $3hop\_ACK$ scheme. It is clear that false alarms may be triggered if Δ is too small. On the other hand, if Δ is too large, each MPR node will have to maintain a longer list, requiring a large memory size. Therefore, Δ should be set to a value that is large enough to allow the occurrence of temporary link failures (i.e. the unsuccessful transmission due to node mobility or local traffic congestion). It is necessary that Δ satisfies the following condition.

$$\Delta > 6 \times (\ single - hop\ transmission\ delay)$$

Where a single-hop transmission delay includes packet transmission delay, random back-off delay at MAC layer and the processing delay.

4.3.3.3 The $3hop\_ACK$ scheme operations

Since our scheme operates at the network layer, it can be implemented as an add-on to OLSR. In our scheme, the $3hop\_ACK$ packet is assigned a route of 3 hops in the opposite direction of TC packet route as shown in Figure 4.8. In this figure, A is assumed to be an MPR node, $B \in N_1$, $C \in N_2$ and D is the A's 3-hop neighbor that should send back the $3hop\_ACK$. Whenever the node A detects any change in its N_1 or N_2 sets, it broadcasts a HELLO message with the $HELLO\_id$ field set to 1. Upon reception of this message, each node in the set N_1 sends a $HELLO\_Rep$ packet to A. When the TC_timer expires, the node A generates its TC message that includes a request for $3hop\_ACK$ to the node D. This TC will be forwarded by B and C, respectively. Each receiver of this TC message checks whether its identity is included in the request list; if so, it returns an authenticated $3hop\_ACK$ packet. Upon reception of this packet, node A decrypts it using the public key of the expected sender to check its authenticity and react accordingly. This process is repeated for every quadruplet of nodes, such as $< B, C, D, E >$, $< C, D, E, F >$ and $< D, E, F ... >$, where the first 3 nodes are MPR nodes and the last one may not be an MPR node.

4.3.3.4 Security analysis of $3hop\_ACK$ scheme

Our scheme is robust against several attack's scenarios conducted by either single or colluding malicious nodes. Particularly, we consider three typical scenarios as described below.

- First scenario: a given node in the set N_1 replies with a false or incomplete list of 2-hop neighbors, in which it adds non neighbor nodes or deletes some neighbors. According to our scheme, when the node D receives the $Hello\_rep$ from its MPR node X_1 it chooses one of the nodes in the 2-hop neighbors list to asks it to send back a $3hop\_ACK$. If the selected node is not a true 2-hop neighbors of X_1, then D will not receive a $3hop\_ACK$. As a result, the node X_1 is deemed as misbehaving.
- Second scenario: in which a misbehaving node, from the set N_2, drops the TC message and fabricates a $3hop\_ACK$ packet on behalf of the requested node. To do so, this node must spoof the requested node's identity and generates a valid a $3hop\_ACK$. However, it cannot do that due to the lack the private key of the requested node.
- Third scenario: in this case, a node $X_2 \in N_2$ correctly relays the TC message but it drops the $3hop\_ACK$ packet. This misbehavior would be detected as well, since the node $X_1 \in N_1$ monitors the transmissions of X_2 and deletes it from its 1-hop neighbor list if its ACK_{miss} counter exceeds the threshold δ_2.

4.3.4 Simulation model and results

This section reports the simulation results of our scheme using the network simulator OPNET 11.5 [59]. We have added the functions that implement our scheme to OLSR source code in OPNET. To evaluate the performance of our scheme in different scenarios, we generate random topologies with M nodes over a rectangular field, where M varies from 30 to 100. The rectangular field size is also varied from (1500 m x 1000m) to (2500m x 2000m). The maximum transmission range of each vehicle is 250m. The vehicles are able to communicate with each others using the IEEE 802.11 MAC protocol and OLSR at MAC and network layers, respectively.

Notice that we have opted for VANETs to evaluate the performance of our scheme due to the fact that the vehicles do not suffer from energy constraints like nodes in MANETs. Thus each vehicle is able to encrypt its acknowledgment packer and checks the authenticity of the received ones without affecting its lifetime. Moreover, the consequences of the black hole attack can be devastating in VANETs since it may lead to human life loss.

Notice that random waypoint model [60] widely used in MANETs cannot be adopted in VANETs due to the particular mobility fashion in this networks. In VANETs, vehicles motion is constrained by a predefined roads and streets, hence we use the vehicular mobility model (VMM) developed in [61]. In this model, initially, the vehicles are randomly placed on intersections. Then each vehicle chooses a desired speed and a target destination. Afterwards, it computes the shortest path to reach it, taking into account single flow roads. Finally, the vehicle moves and accelerates to reach a desired velocity according to streets regulations. When a vehicle moves near other vehicles, it tries to overtake them if the road includes multiple lanes. If it cannot do so, it decelerates to avoid the crash. When a vehicle is approaching an intersection, it first acquires the state of the traffic sign. If it is a stop sign or if the light is red, it decelerates and stops. If it is a green traffic light, it slightly reduces its speed and proceeds to the intersection. At target destination, the vehicle decelerates and stops, then it chooses a new destination.

We run the simulation for 600 seconds with vehicles speed varied from 0 m/s to 30 m/s. For Hello and TC messages generation, we use the default setting as indicated in OLSR specifications [16]. The percentage of

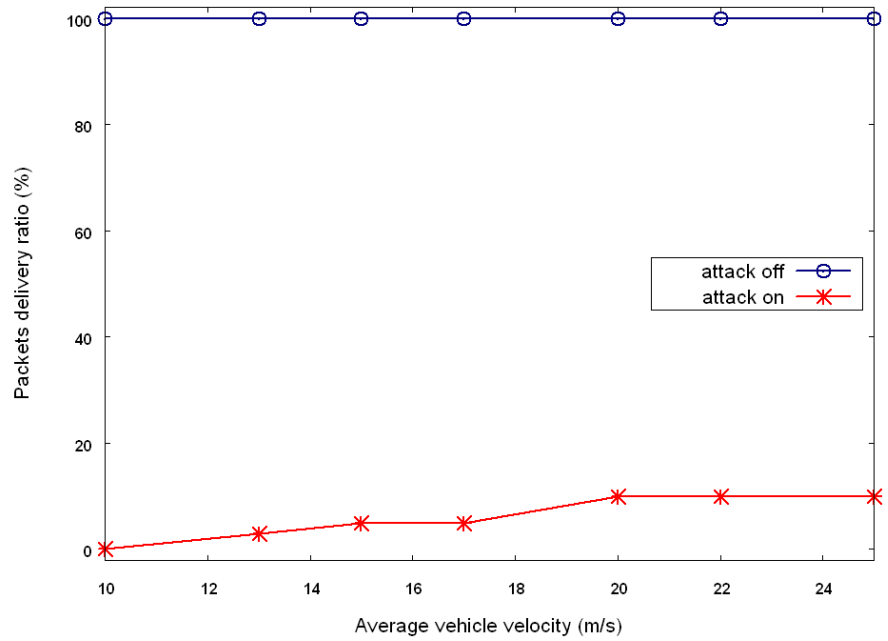


FIGURE 4.9: Delivery ratio vs. average vehicle velocity

malicious vehicles is varied from 25% (i.e. 4 colluding attacks are launched in network of 32 vehicles) to 40% (i.e. 20 colluding attacks are launched in network of 100 vehicles). To launch the colluding attack, the first attacker chooses randomly a victim vehicle from its MPR selectors set, which should be, in its turn, an MPR of its neighbors.

4.3.4.1 Colluding Black hole attack simulation

To illustrate the consequences of this attack, we have generated an exponential traffic between two nodes which are more than three hops away. Notice that the destination node of this traffic is an MPR selector of the victim MPR node. As an evaluation metric, we use the packet delivery ratio which represents the proportion of the number of received data packets to that of packets being sent by the source node. The results are shown in Figure 4.9, from which we observe that when the network is under attack, the delivery ratio is 0% when the average vehicle velocity is 10 m/s and increases up to 5% when the average vehicle velocity goes to 15 m/s. When the vehicles accelerate (i.e. the average vehicle velocity varies from 20 m/s to 25 m/s), the delivery ratio increases up to 10 %. The reason is that, when the destination node moves faster, it has more chances to select nodes other than the victims as MPRs and thus receives more packets; whereas the lower mobility forces the destination node to select the victim nodes as MPRs. Thus, only few routes can be established between the source and destination nodes.

4.3.4.2 Performance evaluation

We have conducted extensive simulations to assess the performance of our scheme. The curves plotted in Figure 4.10 show three scenarios which can be explained as follows:

- Scenario1(Attack on, Countermeasure disabled): plots the average number of receptions of each TC message generated by MPR nodes in case of attacks and our scheme is not used.

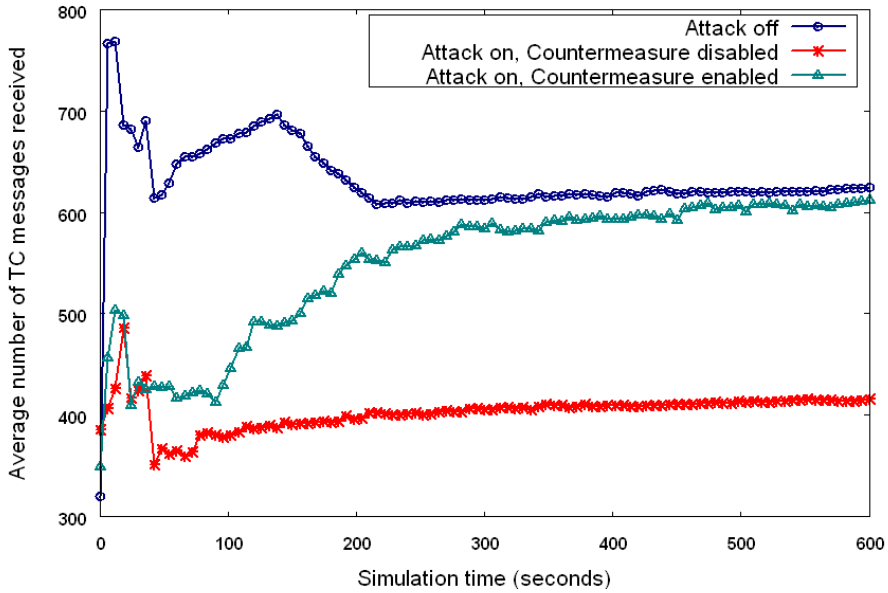


FIGURE 4.10: Comparison of the average number of TC messages received

- Scenario2(Attack on, Countermeasure enabled): similar to previous scenario but in this case our scheme is enabled.
- Scenario3(Attack off): plots the average number of receptions of each TC message where all the nodes are well-behaving.

As depicted in Figure 4.10, the average number of TC messages received decreases sharply in scenario 1. As a consequence, some nodes have a partial image of the network topology because of the missed information which is contained in the dropped TC messages. Therefore they cannot establish routes to the whole network. In scenario 2, we remark that the average number of TC messages received is initially small due to the attack's impact. This number increases gradually each time an attacker is detected until it reaches the number of TC messages received in scenario 3. The implicit reason is that, once an attacker is detected, the victim nodes elect a new MPR set of legitimate nodes which, in its turn, forwards the TC messages correctly.

Figure 4.11 reveals the relationship between the detection rate and the vehicle velocity. We observe that, generally, the detection rate decreases as the vehicle speed increases. When vehicles velocity is 25 m/s, they move continuously, and then the MPR sets of nodes change rapidly. As a result, the malicious vehicles can drop the TC messages and move before its failure counter reaches the misbehaving threshold. We observe also that the frequency of MPR set changing decreases as vehicles velocity decreases. This is due to the fact that when the vehicles speed decreases, the MPR sets become more stable over time and the performance of our scheme gets better.

In order to assess the number of false alarms triggered by our scheme, we generate a network of random topology and which consists of 100 vehicles, among which 40 % are malicious. As shown in Figure 4.12, Y axis represents the number of false alarms, X axis depicts the timeout value δ. The maximum speed of vehicles varies from 10 m/s to 25 m/s and is divided into 3 ranges. We observe that the number of false alarms is generally proportional to vehicles speed and reduces with the increase of timeout value. We observe also that $\delta = 0.3$ is the optimal threshold, after which false alarms never occur for vehicles speed ranges $[10, 15]$ and $]15, 20]$, while only 2 false alarms keep occurring for vehicles speed range $]20, 25]$.

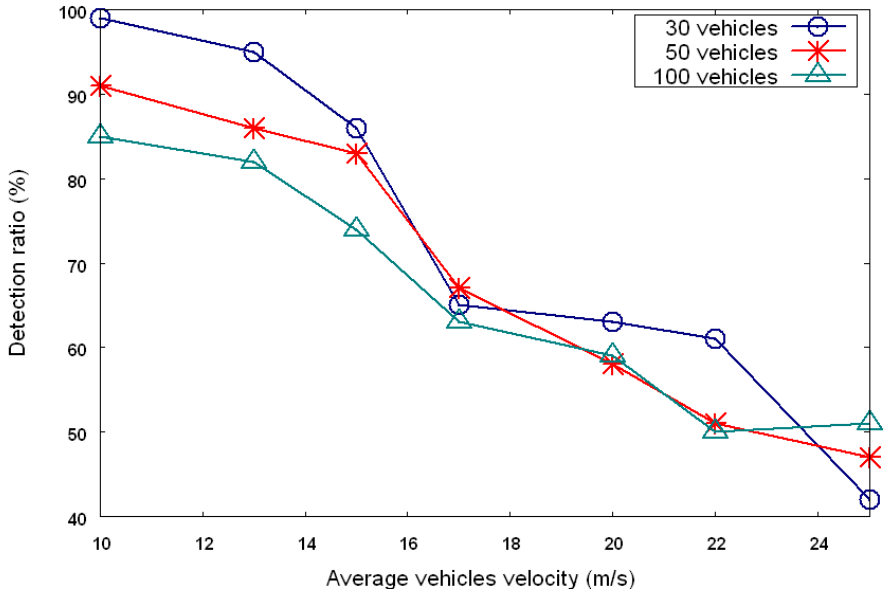


FIGURE 4.11: Detection rate vs. average vehicles velocity

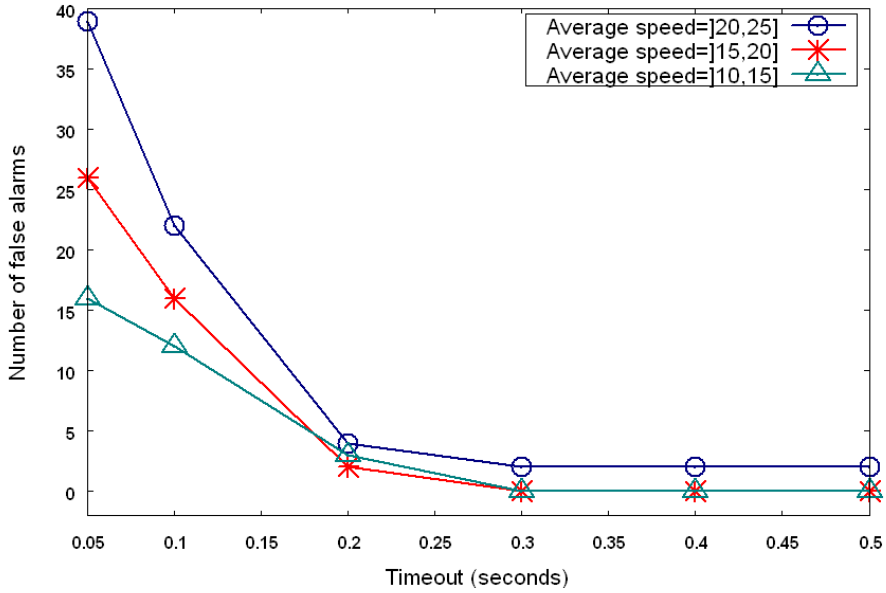


FIGURE 4.12: Number of false alarm vs. timeout value

4.3.4.3 Overhead evaluation

In Figure 4.13, we compare the overhead induced by the control messages of the original OLSR and the overhead added by our scheme (OLSR+). The higher routing overhead in our scheme is due to the transmission of $HELLO\_rep$ packets, $3hop\_ACK$ packets and the increasing size of TC packets. In our scheme, the size of TC packets varies due to both the changing number of MPR selectors as well as the number of requested 3-hop neighbors. When vehicles velocity increases the network topology changes rapidly, then they send extra $HELLO\_rep$ messages and request more acknowledgment from their neighbors. Consequently, the routing overhead in OLSR+ increases with the increase of vehicle velocity. It is worth noting that even though our scheme achieves good results in network characterized by moderate velocity of vehicles and low number of attackers, the generated overhead is

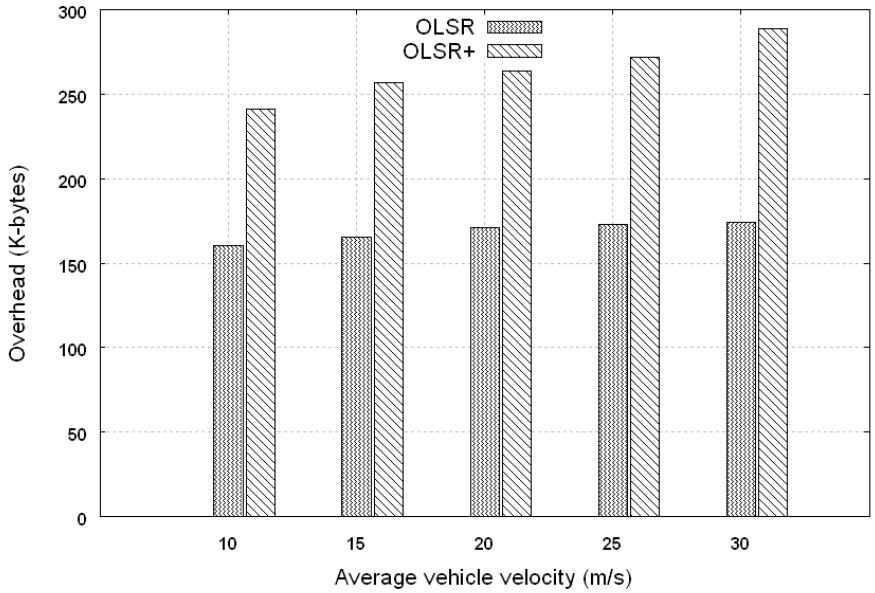


FIGURE 4.13: Routing overhead of 3hop_ACK scheme vs vehicle velocity

a bottleneck for its performance. Hence, reducing the overhead of our scheme while preserving its performance is still an open issue.

4.4 Cross layer black hole attack

This attack is launched through cross layer cooperation of routing and MAC layers.

4.4.1 Attack description

Many devices with different computation and communication capabilities establish temporary links to form an ad hoc network. As opposed to homogenous environment, where symmetric links are the more general observed fact, routing in a heterogeneous MANET is dominated by many asymmetric links. There are several reasons for the appearance of such links, some of which are stated as follows:

- Due to the varying transmission ranges the devices with stronger communication capabilities may reach the weaker ones but the opposite is not possible.
- In order to achieve power-aware communication, the wireless devices adjust their transmitting power according to their residual power such that their lifetimes are extended. In such communication circumstances, some of the symmetric links may become asymmetric when the communication capability of a node degrades due to decrease in the residual power. Figure 4.14 depicts an example of such situation where node A looses connectivity with node C due to the reduction of its transmission range.
- The transmission range of some devices having the same communication capabilities may vary due to fading [62] and random transient phenomenon.

Malicious nodes may get benefits from asymmetric links to launch attacks as depicted in Figure 4.15. In this figure, we consider nodes A, B and M1 running OLSR and having different transmission capabilities. Node A is beyond the transmission range of node B, and similarly B is unable to receive messages sent by node M1. During the neighbor discovery phase, the malicious node M1 relays B's Hello message towards A in order to establish a fake symmetric link (virtual link (VLINK) as dubbed in [63]). At the end of this phase, both nodes A and B believe that they share a symmetric link between them. Therefore a serious degradation of OLSR performance can be resulted as shown later in section 4.4.3.
To summarize, we present the following sequence of Hello messages being exchanged to set up the fake symmetric link.

- **Scenario 1**
  1. $A \longrightarrow * : Hello, \{\emptyset\}$.
  2. $B \longrightarrow * : Hello, \{A, ASYM\}$.
  3. $M1 \longrightarrow A : Hello, \{A, ASYM\}$.
  4. $A \longrightarrow * : Hello, \{B, SYM\}$.
  5. $B \longrightarrow * : Hello, \{A, SYM\}$.

- **Scenario2**
  1. $B \longrightarrow * : Hello, \{\emptyset\}$.
  2. $M1 \longrightarrow A : Hello, \{\emptyset\}$.

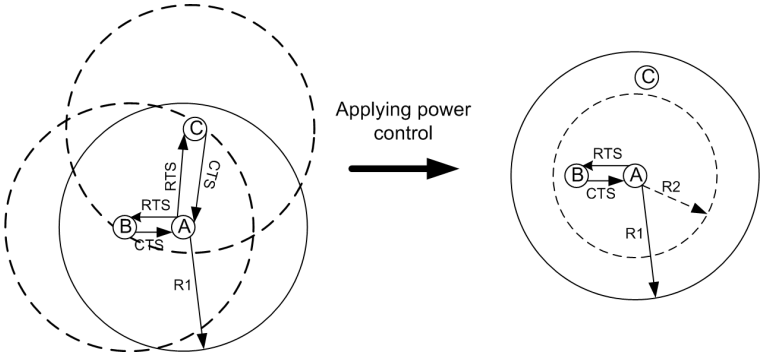


FIGURE 4.14: Impact of power control employment on the transmission range of node A, A's transmission range reduces from $R1$ to $R2$

3. $A \longrightarrow * : Hello, \{B, ASYM\}$.
4. $B \longrightarrow * : Hello, \{A, SYM\}$.
5. $M1 \longrightarrow A : Hello, \{A, SYM\}$.
6. $A \longrightarrow * : Hello, \{B, SYM\}$.

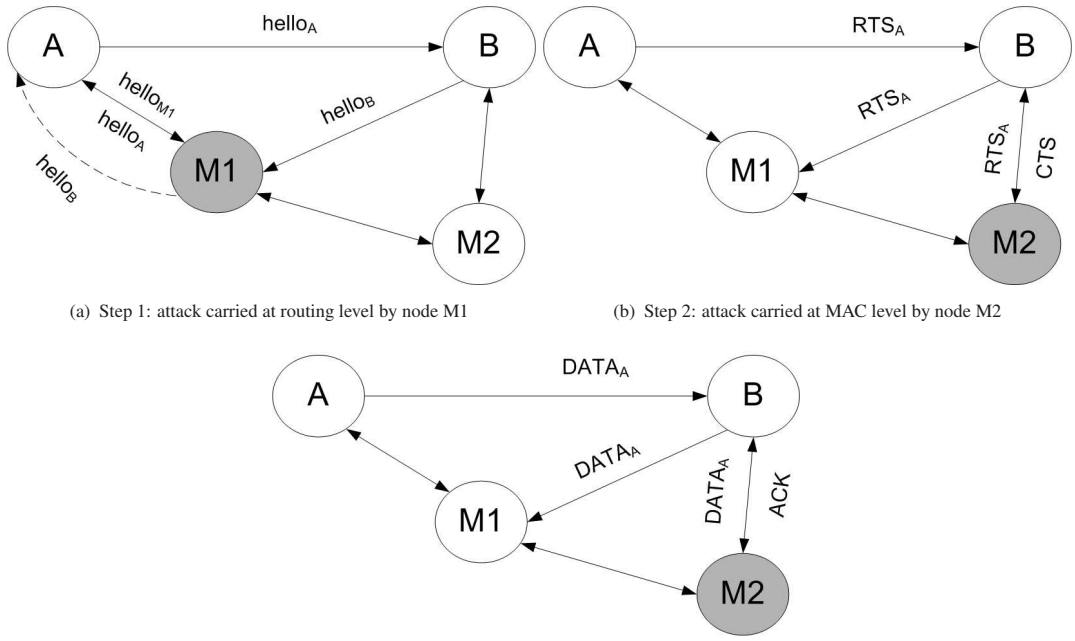
where $*$ denotes the dissemination of a message and {Id, link} refers to the content of Hello message where id is the neighbor identity and link is the status of the link connecting the sender of the message and the node id.

Notice that we distinguish two scenarios which lead to establish the fake symmetric link as illustrated above. In the first scenario the malicious node M1 has to relay the B's Hello message only once to launch the attack, however in the second scenario it has to relay this message twice. Therefore it spends more energy in this latter than the former case.

Since the default value of the interval separating two consecutive transmissions of Hello message is set to 2 seconds, then whenever the victim node B, transmitting packets towards the node A, detects a link break at MAC layer it launches a new shortest path search from the routing table. Notice that a link is lost if the number of missed CTS or ACK frames has overtaken a specified threshold. So after finding a new path the node B transmits its data packets successfully to the intended destination until the next Hello message from A is received again, and the same scenario will be repeated.

In order to prevent such situation, the malicious node M2 replies to all RTS and DATA packets sent by node B by sending back the corresponding validation frames CTS and ACK respectively as depicted in Figures 4.15(b) and 4.15(c). This misbehavior is called false validation attack. Therefore, the victim node B keeps constantly transmitting its packets through the compromised link and consequently none of them reaches its destination

For both traffic flows TCP and UDP this attack leads to data packets loss since no link break advertisement is sent to the higher layer to replace the broken link. For TCP flows the sender node reduces its sliding window size gradually each time the expected end-to-end acknowledgment is missed until it reaches zero and the flow is interrupted accordingly, however in UDP traffic the sender continues transmitting its packets until its completion and hence it consumes more energy uselessly. For the security point of view, this attack leads the sender node to falsely accuse the intended receiver as misbehaving or decreases its reputation and trust level if any monitoring system is set at routing layer such as watchdog or other schemes that require an explicit authenticated acknowledgment to verify that its next hop forwards the packets correctly. Therefore, false alarms may be triggered in the network and consequently longer paths and network partition may result.



(a) Step 1: attack carried at routing level by node M1

(b) Step 2: attack carried at MAC level by node M2

(c) Step 3: attack carried at MAC level by node M2

FIGURE 4.15: Attack description at both routing and MAC layers

4.4.2 The proposed solution

In this section we give an overview of the detailed functioning of our solution and its assumptions along with the analysis of the possible scenarios which may be conducted by the colluding nodes M1 and M2 in order to break the solution.

In order to cope with the attack described in the previous section, we have developed a cross-layer solution based on cooperation between routing and MAC layers. In this solution, we assume that each pair of nodes shares a secret which is undisclosed to any other node and that each node holds a collision free hash function such as SHA-1 (Secure Hash Algorithm 1) and MD5 (Message Digest 5). Notice that the preliminary distribution of keys or secrets between the nodes in MANET can be carried out using some well known schemes proposed in the literature such as [64] and [65].

Whenever a node receives a Hello message advertising its identity as an ASYM neighbor then it schedules a transmission of RTS+ frame towards the sender of this message. The frame RTS+, as depicted in Figure 4.16, is a special RTS frame in which we add a new field of 16 bits dubbed RTS sequence number (RSN) and replace the @R (destination address) field with the hash value of the shared secret (SS) between the sender and the intended receiver combined with the RSN value. The purpose of adding the field RSN is to prevent reply attacks. Moreover, we use the value hash (RSN \\ shared secret) as a destination address to prevent any malicious node from replying an old RTS+ frame in order to deplete the node A's energy. Note that the symbol \\ represents the concatenation operation of RSN and the SS.

The value of RSN is increased by 1 at each transmission or retransmission of RTS+ frame as well as upon reception of a CTS+ frame. Note that the duration field in RTS+ is calculated as follows:

$$Duration = T_{CTS+} + SIFS \tag{4.4}$$

because no DATA frame transmission will follow the reception of CTS+ frame. Note that T_{CTS+} refers to the transmission time of the CTS+ frame.

Each node receiving the RTS+ frame computes the hash value of the RSN value combined with its shared secret with the source node, if it is equal to the value received in RTS+ then the node sends back the corresponding CTS+ frame with duration field set to 0[1] and the @R field sets to the hash value of the shared secret combined with the value (RSN+1). The format of CTS+ is shown in Figure 4.17 where its size is 10 bytes larger than the standard CTS frame.

If the sender of RTS+ didn't receive the corresponding CTS+ within the timeout period for several times then this is a confirmation that the intended receiver is under attack launched by a third node and consequently no symmetric link with this victim node will be advertised in the next Hello message.

Remark Since the nodes in MANET are equipped with limited battery power and modest computation capabilities, we have opted for hash function rather than public/symmetric key cryptography as it is characterized by its low cost and fast operations. Notice that the operation speed is a strict requirement since the delay separating the reception time of the last bit of RTS+ and the transmission of the first bit of CTS+ should not overtake the SIFS duration.

Security analysis

Let us now analyze the possible scenarios by which the malicious nodes M1 and M2 try to compromise the proposed solution.

Despite the fact that the destination address of RTS+ frame is hidden the node M1 may relay all the heard RTS+ frames or a randomly chosen subset of them towards the victim node. In this case, the victim node will certainly receive one RTS+ in which it is the intended receiver, however due to the incurred delay (d), as a consequence of the retransmission of the RTS+ by the node M1, the CTS+ will be received after the expiration of the timeout value TO_{CTS+}. Hence, the link with the destination node is deleted. Moreover, even though the node M1 is equipped with a set of directional antennas it is unable to receive a frame using one antenna and transmits by another antenna simultaneously. Therefore, the incurred delay for forwarding the RTS+ frame remains important. The values of TO_{CTS+} and delay are calculated as shown in the equations below.

$$TO_{CTS+} = T_{RTS+} + SIFS + T_{CTS+} \tag{4.5}$$

$$d = TO_{CTS+} + T_{CTS+} \tag{4.6}$$

where TO_{CTS+} is the expected duration for receiving the CTS+ at the sender node, T_{RTS+} and T_{CTS+} are the transmission time of RTS+ and CTS+, respectively, whereas the signal propagation delay is ignored.

For more sophisticated scenario, we suppose that the malicious node M1 records the CTS+ at time t and sends it to node M2 via an encrypted packet. Then, the node M2 replays it later at time t+ Δ_t (because due to nodes mobility the links status change frequently and the nodes have to check the symmetry of every new established link) in order to falsely validate the subsequent RTS+ towards the same destination. The CTS+ frame replied by

[1]The duration field is set to 0 because no further DATA packet will be exchanged.

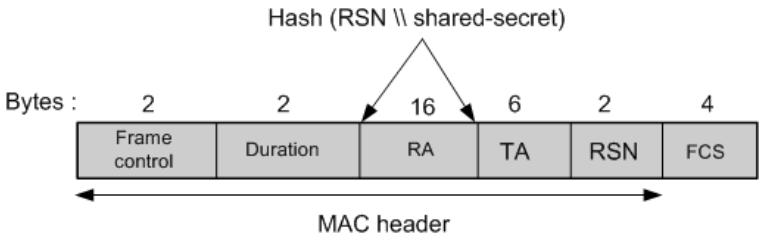


FIGURE 4.16: The format of RTS+ frame (32 bytes)

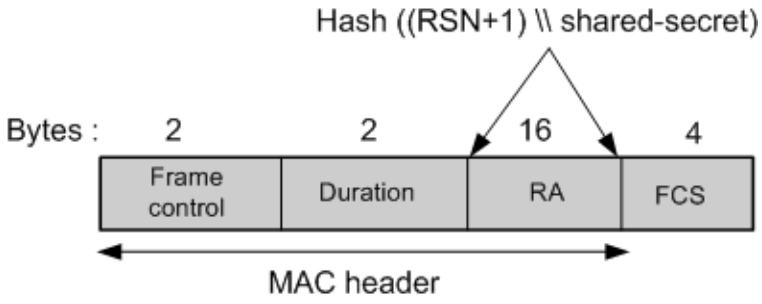


FIGURE 4.17: The format of CTS+ frame (24 bytes)

node M2 will not be considered as a valid CTS+ since the expected hash value would be calculated using a RSN larger than the one used for the old CTS+ kept by M2. Moreover, the node M2 is unable to compute the expected hash value as the shared secret is unknown.

The operation of the proposed solution is summarized in the flowchart given in Figure 4.18. This flowchart describes the treatment carried out by any node in the network upon reception of a Hello message. As we can see from this flowchart, after sending the RTS+ the node waits for CTS+ reception. If it is received after timeout expiration or not received at all and timeout is expired then it is ignored and the missed CTS+ counter is increased. Otherwise, if it is received before timeout expiration then its validity and authentication should be checked as well in order to prevent any forged or old CTS+ replied by a malicious node.

4.4.3 Simulation

This section reports the simulation results obtained by implementing the attack described in the section 4.4.1 in OPNET 14.0 network simulator [59]. The simulation settings are summarized in table 6.5. We consider a MANET consisting of 14 wireless nodes having different transmission ranges. These nodes are distributed within the area as shown on the topology depicted in Figure 4.19.

In order to highlight the impact of this attack a CBR traffic flow f (500 bytes/packet and 50 packets/s) is initiated from the node N_2 towards the node N_{10}. Notice that the transmission of data packets is started 20 seconds after the beginning of the simulation in order to allow each node to construct routes towards the rest of the network.

On the other hand, the nodes M1 and M2 colludes to launch a cross layer attack against the nodes A and B, by applying the same steps described in section 4.4.1. As a result of this attack, the MPR sets of nodes B and A are changed due to the new established fake symmetric link as illustrated in Table 4.3 and consequently the routes towards far away nodes are also changed accordingly. Therefore the new shortest paths shown in Table 4.5 replace

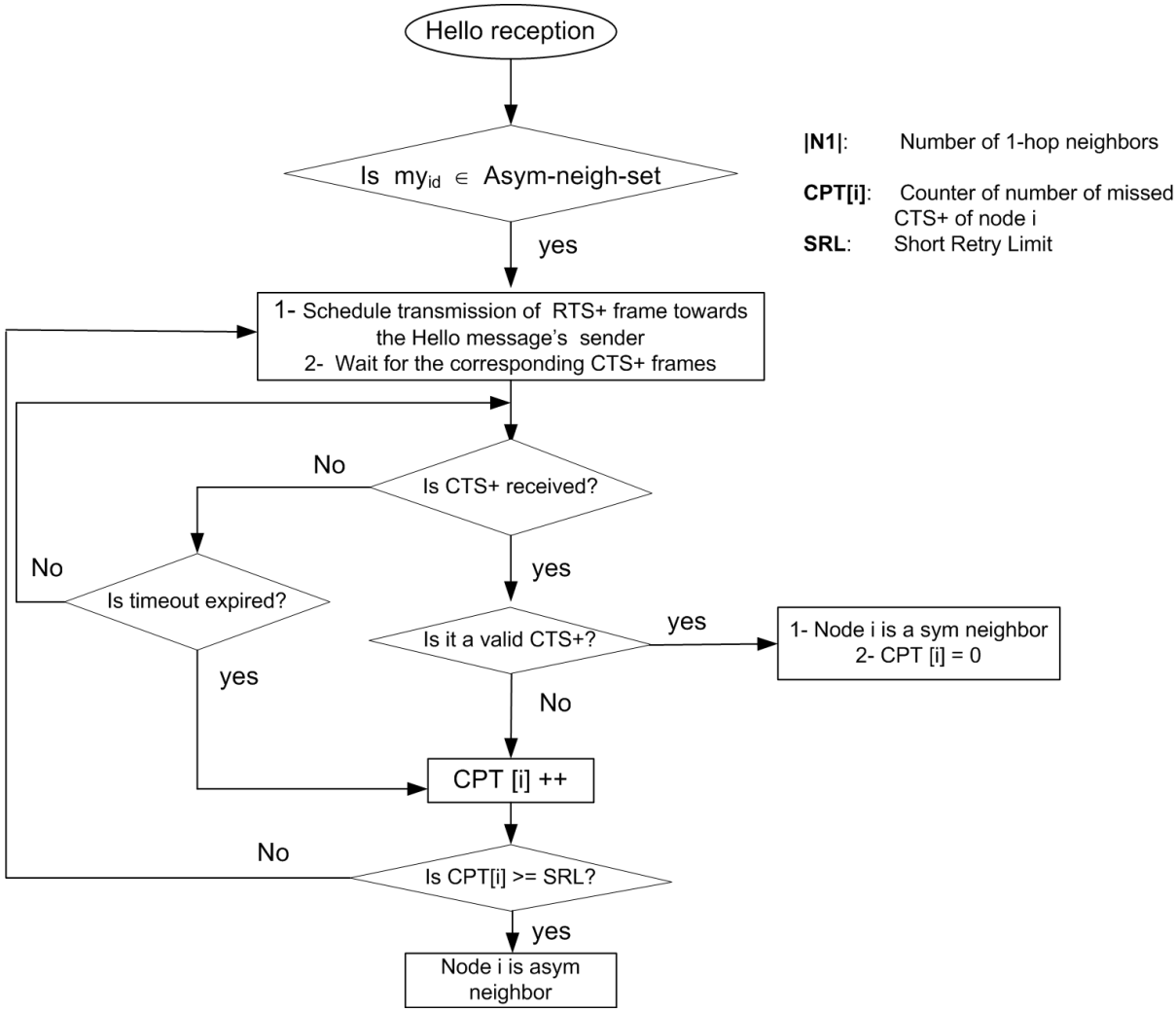


FIGURE 4.18: Flowchart describing the functioning of our solution

the earlier paths depicted in Table 4.4. Hence, the link (B, A) is becoming a black hole which absorbs all the packets routed through it.

The Figures 4.20 and 4.21 graph the data packet delivery ratio of the flow f in the case where only VLINK attack is carried out and the case where both attacks are launched together, respectively. As we can see from these figures, when only VLINK attack is launched the delivery ratio decreases dramatically however a number of transmitted data packets still able to reach their destinations. This is due to the fact that when link break is detected at MAC layer as a consequence of missed CTS packets for SRL (short retry limit) times, node B proceeds for selection of a new path towards the node N_{10}. This new path is maintained until it is replaced again by the compromised route upon reception of the subsequent Hello message forwarded by the malicious node. In the other hand, when both of the attacks are launched (cross layer attack) the delivery ratio drops sharply because, in this case, no link break is detected as each data packet transmitted by node B is validated by node M2 (M2 sends back the corresponding CTS and ACK frames) which forces the node B to keep transmitting/forwarding data packets over this path.

Moreover, as depicted in Figure 4.22, when both of the attacks are launched together the node B will consume more energy in forwarding the data packets passing through it as compared to the case where VLINK attack is launched solely. As a consequence the energy of node B will deplete quickly which decreases its life time. As a

Parameters	Values
Area	1500m *1000m
MAC protocol	IEEE 802.11b
Transmission range	250 m
	M1: 150m
	M2: 200m
	A : 300m
	B : 250m
Traffic type	CBR
Data rate	11mbps
CBR packets size	512 bytes
Buffer size	62 packets
Short Retry Limit (SRL)	7
Long Retry Limit (LRL)	4
Mobility model	Random way point
Hash function	MD5 (128 bits)
Simulation time	600s
No. of simulation epochs	5

TABLE 4.2: Simulation settings

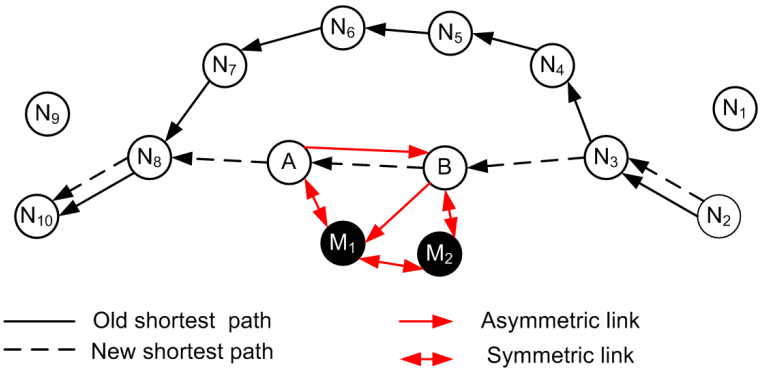


FIGURE 4.19: Network topology illustrating an example of the studied cross layer attack

Node	MPR set before attack	MPR set after attack
A	N8	N8, B
B	N3	N3, A

TABLE 4.3: The MPR sets of nodes A and B

simple comparison, the node B transmits 520 bytes (500 bytes corresponds to one data packet's size and 20 bytes corresponds to the RTS frame's size) rather than 140 bytes (7 transmissions of RTS frame which is equal to SRL) in case where VLINK attack launched solely.

Dest addr	Next hop	Number of hops
A	N3	7
N8	N3	6
N10	N3	7

TABLE 4.4: Routing table of node N2 before the attack

Dest addr	Next hop	Number of hops
A	N3	3
N8	N3	4
N10	N3	5

TABLE 4.5: Routing table of node N2 after the attack

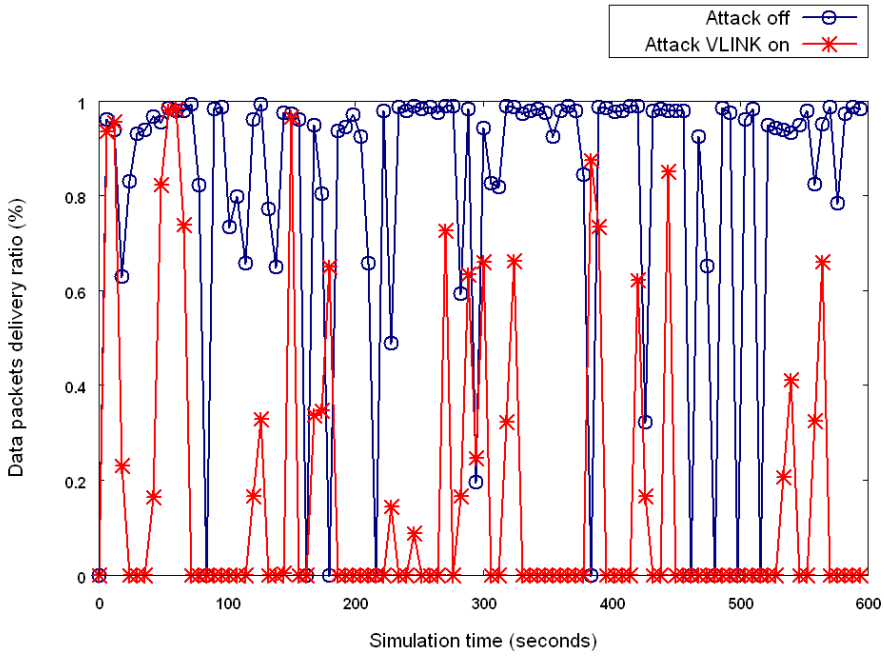


FIGURE 4.20: Data packets delivery ratio under VLINK attack solely

Solution efficiency and overhead evaluation

In the sequel we set up a network consisting of 30 nodes with different transmission ranges, among these nodes 4 are attackers which collude each other to launch cross layer attacks against the other nodes. The nodes are randomly placed within the area and 8 CBR traffic flows are generated in the network (500bytes/packet, 50 packets/s).

Figure 4.23 shows that our solution performs well when the speed of nodes is low (0 m/s and 5 m/s) because the lower mobility of nodes allows a faster verification of links symmetry using the proposed technique, hence almost the same packets delivery ratio is maintained as compared to the case of network without attackers. When the nodes move faster the link verification phase may take a longer delay and therefore some data packets may be dropped due to the lack of an established path to the destination or data buffer overflow at MAC layer. Despite that, our solution keeps ensuring around 84% of the packets delivery ratio reached in the case where no attack is launched.

To assess the overhead generated by our scheme in terms of the number of the extra bytes sent by each node we vary the mobility speed of nodes as well as their pause time. Note that in the case of static network (nodes speed =

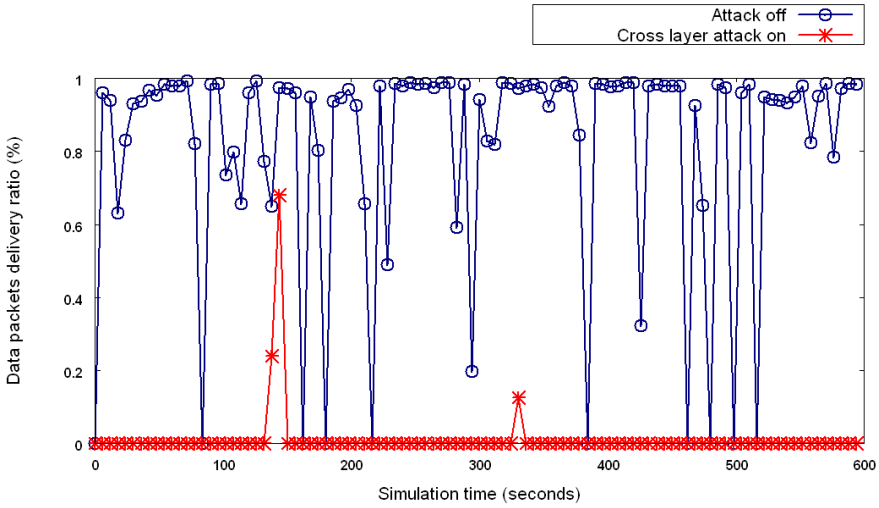


FIGURE 4.21: Data packets delivery ratio under the cross layer attack

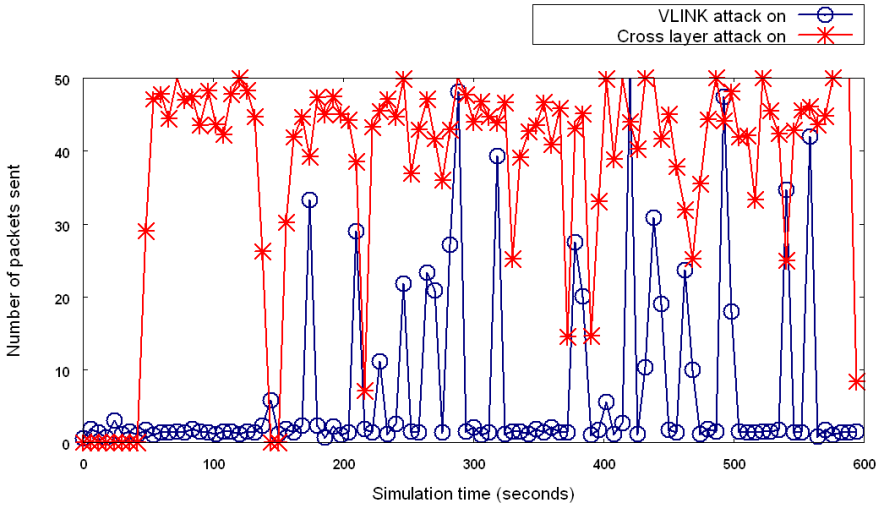


FIGURE 4.22: Data packets forwarded by node B

0m/s) the value of pause times is insignificant. According to the results shown in Figures 4.24(a) and 4.24(b) we can see that the extra bytes transmitted by the nodes to prevent the cross layer attack are very small compared to the transmitted bytes representing Hello messages, for a static network. This difference decreases gradually with the increase of nodes speed until the overhead induced by RTS+ and CTS+ surpasses the one induced by Hello messages when the nodes speed reaches 20m/s and their pause times is 0 and 10 seconds. This increase is justified by the rapidly change of the one hop neighbors set due to the high mobility of nodes, therefore links are appeared and disappeared quickly which increase the number of the transmitted RTS+ and their corresponding CTS+ frames in order to verify the symmetry of these new links. Consequently, we see that generally the extra overhead induced by our solution increases when the speed of nodes turns to larger and their pause times gets smaller.

As a conclusion, the cross layer attack being studied is more harmful for static networks and is less damaging in highly mobile networks. In the other hand, our solution works perfectly with static networks and maintains good results when the nodes start moving.

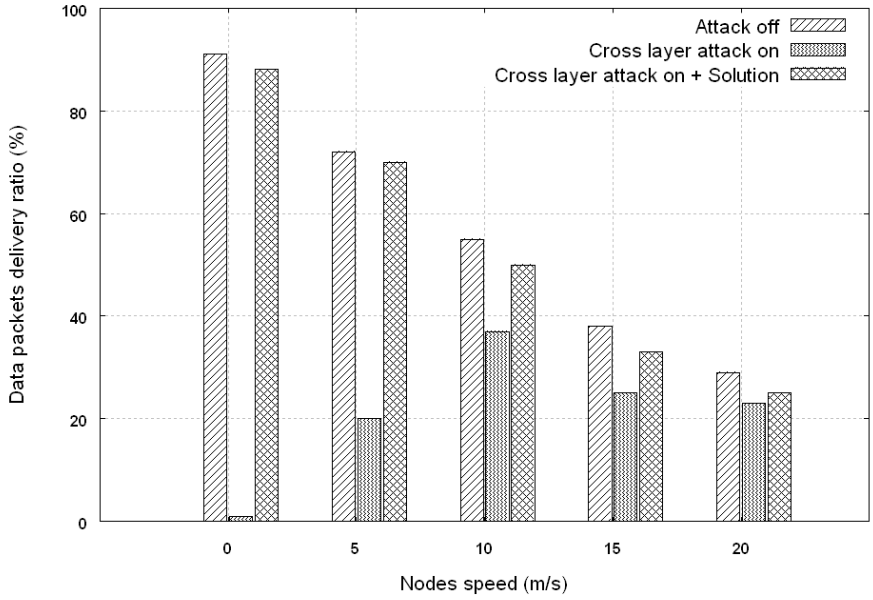


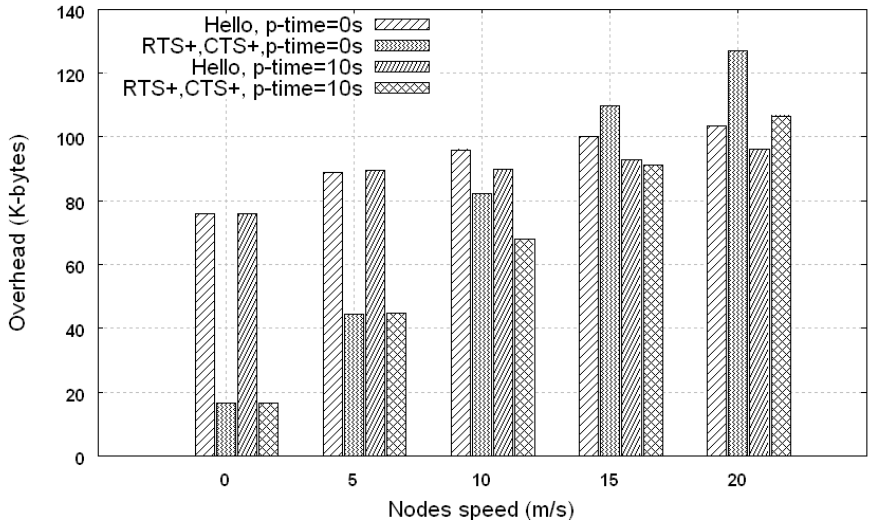
FIGURE 4.23: The proposed solution efficiency in terms of data packets delivery ratio under various nodes speed

4.4.4 Conclusion

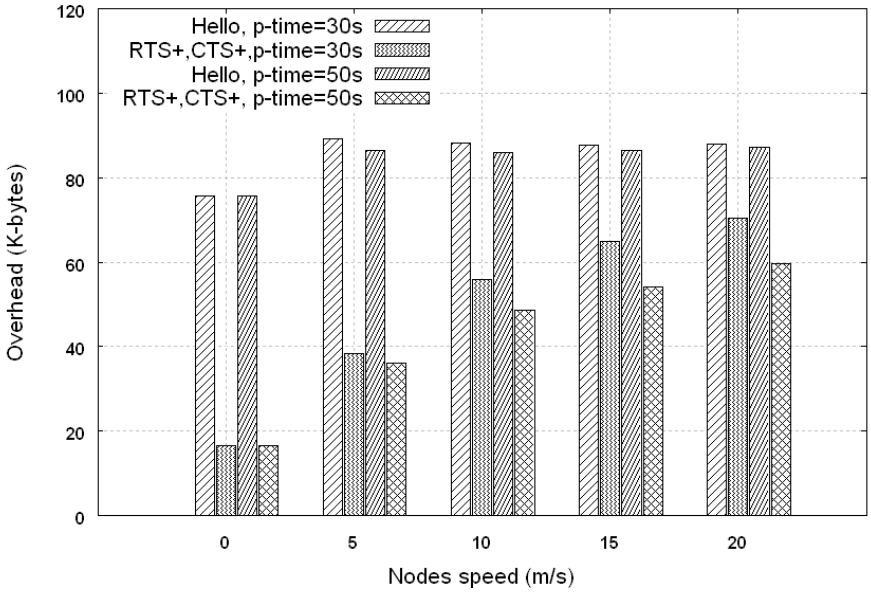
In recent years, VANETs are receiving tremendous attentions from the car manufacturers and the networking research community due to their flexibility and their importance of increasing safety of drivers and passengers. Inter vehicles communication (IVC) is set up using MANETs routing protocols for providing connectivity between remote vehicles. However, most of existing ad hoc routing protocols assume a cooperative and trusted environment, thus hostile environment may expose MANETs to several security attacks. One of the most destructive attacks is the black hole attack that we have investigated throughout this chapter and proposed adequate countermeasures to cope with it.

In this chapter, we have first analyzed the colluding packet dropping attack and proposed an acknowledgment based scheme to detect the attacker nodes and exclude them from the forwarding path. This scheme has been evaluated in IVC network and the obtained results show that it can keep high detection rate under various scenarios, and may trigger a negligible number of false alarms.

In the second part of this chapter, we have investigated a cross-layer scenario of black hole attack. Such attack is a combination of the virtual link attack launched at routing layer and the false validation attack launched at MAC layer. This attack can target any link state routing protocol in MANETs however its damage differs from one protocol to another. A cross-layer solution is then proposed to avoid the harm caused by this attack. In this solution, a node has to check the symmetry of each new established link by sending a special MAC frame dubbed RTS+ and waiting for the corresponding CTS+ frame. The reception of the frame CTS+ within the timeout period and with a valid authentication value confirms the symmetry of the checked link. Thus, we have carried out a solution at MAC layer to prevent the cross-layer attack that it is initially launched at routing layer. The simulation results have revealed that the impact of this attack is more severe in MANETs with low mobility as compared to that with highly mobile nodes. Despite that, our solution is shown to be efficient to struggle against this attack and can successfully prevent the establishment of the fake symmetric links in different scenarios, however its performance drops when the nodes move very fast.



(a) Case of node pause time equal to 0 and 10 seconds



(b) Case of node pause time equal to 30 and 50 seconds

FIGURE 4.24: Variation of the overhead added by RTS+ and CTS+ frames versus nodes speed and pause time

Part II

MAC Layer Misbehavior

Chapter 5

Greedy Behavior in Wireless Mesh Networks

Our focus in this chapter is to study the greedy behavior in Wireless Mesh Networks (WMNs), particularly the adaptive cheating technique, and propose adequate solutions to cope with it. In the first part, we present the FLSAC (Fuzzy Logic based Scheme to Struggle against Adaptive Cheaters) scheme and assess its performance and efficiency. In the second part, a Bayesian model will be introduced and integrated with FLSAC in order to ensure better efficiency and accuracy.

5.1 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 5.1. 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 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 (MCLs) 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).

The properties of WMNs, such as shared wireless medium, open network architecture and stringent resource constraints for MCLs make the security of communication a hard task to achieve. Due to these characteristics, WMNs are vulnerable to several types of security attacks at different layers. Particularly, at MAC layer where the security flaws of IEEE 802.11 MAC protocol [66] encourage the misbehaving MCLs to launch their attacks easily.

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 a considerable research attention leading to an in depth investigation and analysis of its root causes [67], [68]. 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 [69], [70] and [71]. 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 the detection system.

In this chapter, 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 MCL. 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. After describing the functioning of FLSAC and presenting the performance evaluation results, we introduce our second scheme which is based on a Bayesian statistical model and which will be later integrated with FLSAC in order to ensure higher accuracy and enhance the detection rate achieved by running FLSAC solely.

5.2 The adaptive cheating technique at MAC layer

As stated earlier in Chapter 2, adaptive cheating technique is a particular case of MAC layer misbehavior. In this technique, the cheater node tries to apply more than one misbehaving strategy; each of them is enabled during a short period and alternates between them either randomly or following deterministic scheme. 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 5.2. 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.

5.3 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 MCL attached to them will affect both of packets delivery towards these MCLs, and the forwarding of their packets towards far away MCLs or internet.

Let us now suppose that the carrier sensing range (R_{cs}) of a MCL is slightly larger than its transmission range, which is considered as the best case regarding the propagation of the greedy behavior impact in the network. As shown in Figure 5.1, when a misbehaving client (the red MCL) 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

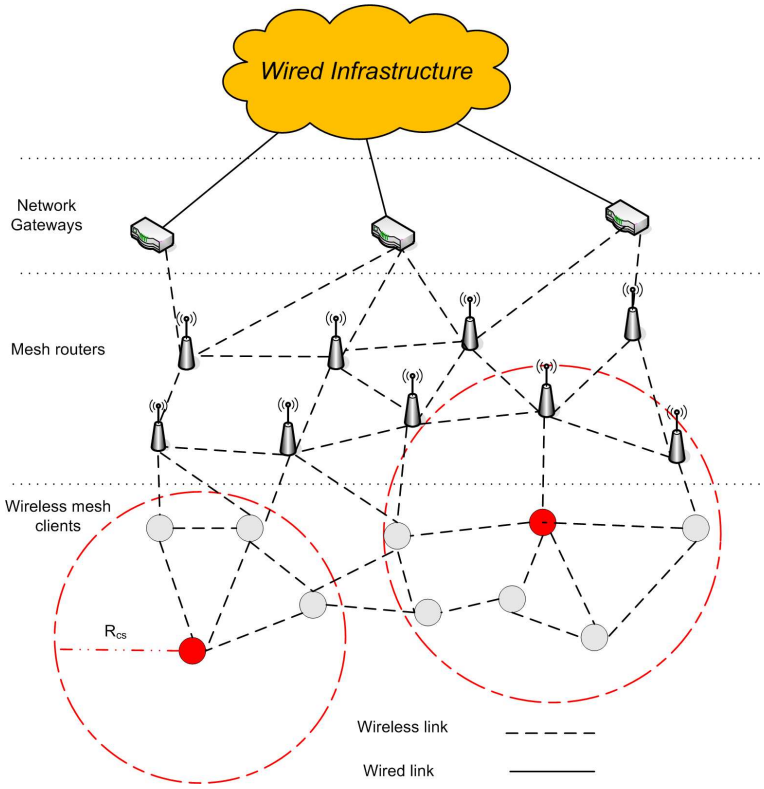


FIGURE 5.1: Wireless Mesh Networks model

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 wireless 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..

5.4 Related work

Many solutions have been proposed so far to deal with the greedy behavior at 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 add new components at 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.

5.4.1 Backoff algorithm modification based schemes

In order to handle the problem of MAC layer misbehavior, [70] has proposed to modify the BEB (Binary Exponential Backoff) algorithm specified in IEEE 802.11 MAC protocol [9], in order to facilitate the detection of cheater nodes. The main assumption of this approach is that the receiver node is trustworthy, and it is responsible

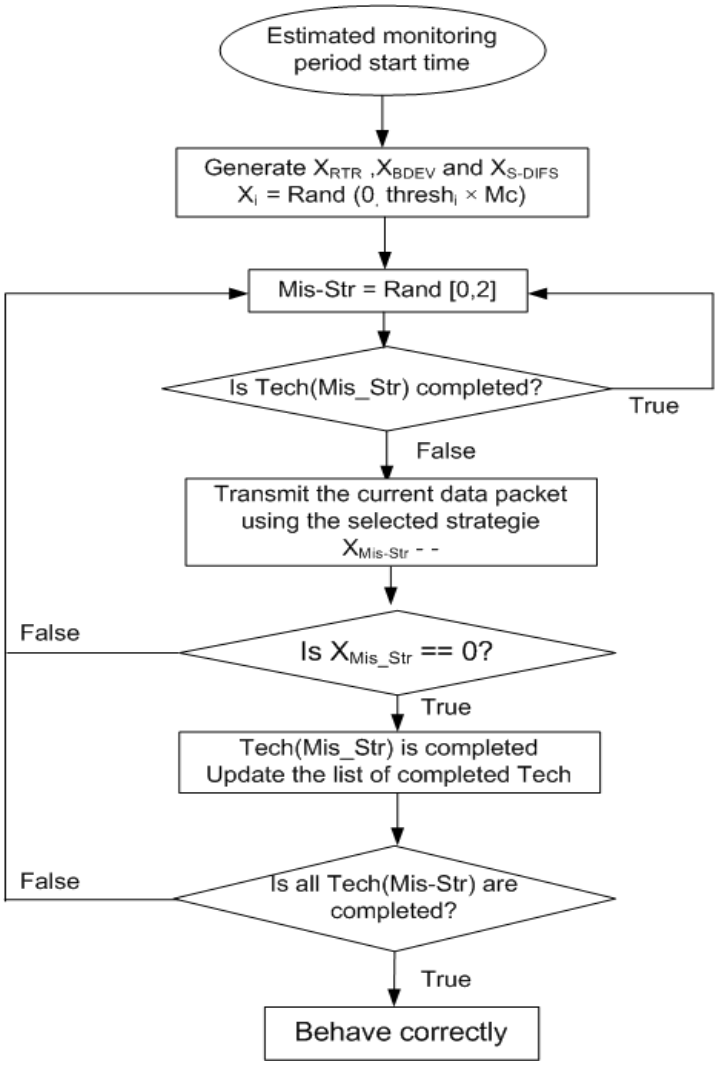


FIGURE 5.2: The switching scheme used by the adaptive cheater to switch over the cheating strategies

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 ongoing 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 value. If the total deviation of the sender throughout the last N observations exceeds a predefined threshold, then it 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.

In order to ensure the randomness of the backoff value, a new scheme has been proposed in [69]. This scheme assumes that at least one of the communicating nodes (sender or receiver) is honest and a reputation system, such as CONFIDANT [46], is deployed at MAC layer. Its main idea is to let both the sender and receiver agree on a random value of backoff through a public discussion. This discussion is based on a commitment scheme inspired from the protocol applying flipping coins over the telephone, which has been introduced in [72]. This scheme circumvents the misbehavior of one part of the communication, however it is still vulnerable to collusive attacks.

In order to guarantee a faster detection of the cheaters, the authors of [73] have developed the PRB (Predictable Random Backoff) algorithm. PRB is based on slight modification of the standard backoff algorithm by forcing each node to choose its 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} along with the number of failed transmissions. 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 it presents the following drawbacks:

- The backoff value observed by the receiver may be different from the one generated by the sender due to hidden terminal phenomenon, interference and inter frame delay of TCP traffic. Hence per frame detection may increase the probability of triggering false alarms and consequently punishing honest nodes.
- Since in PRB each node selects its backoff from a smaller interval as compared to the BEB (Binary Exponential Back off) algorithm, the number of collisions increases, leading to higher packet delay and low channel utilization.

5.4.2 Monitoring based schemes

The authors of [71] 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 5.2.

The sequential analysis concept introduced by Wald in [74] was widely used by researchers to struggle security attacks in wireless networks. The scheme presented in [75] is based on this concept; it describes an analytical model for the packets inter-arrival time distribution in saturated networks, representing an extension of Bianchi's stochastic model [76]. Based on this model the authors have developed an algorithm to detect the cheating nodes by observing the throughput earned by each node. These observations are further evaluated through a sequential probability ratio test to identify which node is not obeying the protocol rules. To ensure its correctness, this scheme assumes the knowledge of the exact value of the greedy factor (i.e. the interval from which the greedy node selects its back off value), however this information is not available in practice. Therefore this scheme cannot work in real environment.

In [77], a statistical framework is developed in order to detect selfish nodes which deliberately modify their contention window to increase their throughput. First, a sample of number of idle slots between successful transmissions of each node is collected. Subsequently, the Kolmogorov-Smirnov (K-S) [78] test is applied to distinguish the misbehaving nodes (using unpredictable strategy) from the legitimate ones. Notice that two detectors have been proposed by the authors, a batch detector based on Neyman-Pearson test and q sequential detector based

on Wald's test. The results have shown that both of the detectors successfully identify the cheaters for the majority of the applied strategies.

A framework aiming to cope with backoff rules violations has been proposed in [79]. This framework designs a deterministic scheme that allows nodes to identify any neighbor node disobeying the backoff rules. This scheme requires that each node uses its MAC address as a seed of the pseudo random number generator used to generate its backoff values. Thereby, each node became 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 a correct diagnostic, a statistical inference method is adopted wherein a series of tests [80] are applied.

A minimax [81] decision making formulation of MAC layer misbehavior is proposed in [82]. The advantage of this approach is that it provides detection rules requiring 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 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, a special case wherein an attacker delays the decision making, as long as possible, in order to prevent being detected is also studied.

5.4.3 Game theory based schemes

Game theory has been widely applied for assessing the impact of selfish behavior in CSMA/CA, and numerous contributions have been proposed to cope up with. In [83], the authors adopt a dynamic game based scheme to derive the conditions which lead the 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. Thus, 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. Therefore, the obtained throughput is significantly lower than that achieved in the case where the cheaters are minority, which is the common case in most wireless networks, especially in WMNs.

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 and alternatively switches between them. To address this issue, we have proposed a fuzzy logic based scheme, dubbed FLSAC, that ensures higher efficiency against such sophisticated misbehavior. The key differences between our solution and the other approaches are exhibited in table 5.1, from which we can see that FLSAC is the unique scheme able to detect the 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 remark also that the schemes proposed in [82], [70], [73], [79] and [75] consider that a cheater node behaves similarly in an Ad Hoc network or within a Wireless LAN, and develop their solutions based on 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 Ad Hoc networks before suggesting solutions.

	Schemes								
Features	FLSAC	DOMINO	PRB	[82]	[70]	[79]	[75]	[83]	[77]
Detection of adaptive cheater	Yes	No	No	No	No	No	No	No	No
Applies the standard BEB	Yes	Yes	No	No	No	No	Yes	No	Yes
Use the standard packet format (RTS/CTS/DATA)	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Keep the randomness property of BEB	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

TABLE 5.1: The key difference between FLSAC and the existing schemes

5.5 Fuzzy Logic based scheme to Struggle against Adaptive Cheaters (FLSAC)

5.5.1 Scheme description

The operations of our solution are described in Figure 5.6. These operations can be summarized as follows; first, the mesh router/gateway monitors the wireless MCLs 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.

5.5.1.1 Main idea

Our solution aims to extend the DOMINO by setting up a first defense line 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 rules 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 [84] which is proven to be suitable in such cases.

The advantage of such technique is to eliminate the decision making ambiguity regarding the behavior of an adaptive cheater which never reaches a threshold of one misbehavior technique, however it still gaining more bandwidth than its neighbors.

5.5.1.2 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 [85].

In what follows, we describe, in details, the functioning of FLSAC along with the role of each of its components, shown in Figure 5.7.

Inputs We have designed our fuzzy system to support four inputs where three of them represent the traces collected by the MR for each MCL. These traces represent the possible misbehavior techniques discussed in Chapter 2. The output of each monitoring period is used as the fourth input of the system in the next period in order to make the final decision more accurate.

Backoff DEViation (BDEV) In DOMINO, at the end of each monitoring period T the MR computes the average of the observed backoff values of a given wireless node 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 :

$$\frac{\sum_{i=1}^{N-m} B_i + \sum_{j=(N-m)+1}^{N} B_j}{N} \geq \alpha \times B_{MR} \tag{5.1}$$

Even if the detection system's parameters, such as T and B_{MR}, are not easy to guess, a cheater node 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:

$$BDEV = \frac{\frac{\sum_{i=1}^{N}(B_{MR} - B_i)}{N}}{B_{MR}} \tag{5.2}$$

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 because we are interested to the small values of backoff, and also to prevent the cheater from hiding its misbehavior by choosing some large backoff values to escape from detection.

Illustrative Example :

Let us suppose that during a monitoring period the MR has observed 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 estimates the following deviation percentage with respect to the Equation. 5.2.

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

This BDEV value will be processed further by the fuzzy controller together with the other inputs to assess the global deviation of the MCL, and then classify it accordingly.

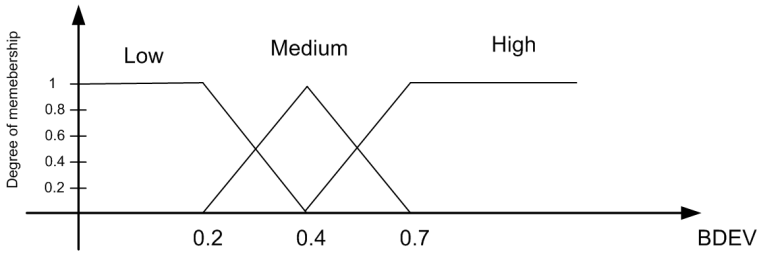


FIGURE 5.3: Example of the membership function for BDEV

ReTransmission Rate (RTR) This parameter is used to detect the MCL which scrambles other's frames in order to increase their contention windows. It is calculated according to the formula below.

$$RTR = \frac{num - rtx(S_i)}{AVG_{i \neq j}[num - rtx(Sj)]} \tag{5.3}$$

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

Frames sent after Short DIFS (S-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 that consumed for backoff. Therefore, the accurate measurement of this time is feasible only if the backoff value selected by the cheater is 0.

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 [84]:

$$\beta_F : U \rightarrow [0, 1] \tag{5.4}$$

such that for each $u \in U$, its degree of membership to F is given by $\beta_F(u)$.

The membership function In FLSAC, we use the membership functions presented in Figures 5.3, 5.4 and 5.5 to calculate the degree of membership of the inputs described above to each fuzzy set. As RTR represents the ratio of the number of retransmissions of each node to the average of retransmissions of its neighbors, then the higher the RTR is the lower observed deviation. Notice also that the value Max, used in Figure 5.5, refers to the maximum tolerable number of deviations from DIFS value.

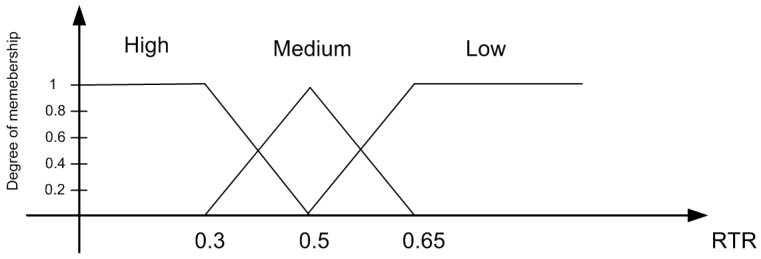


FIGURE 5.4: Example of the membership function for RTR

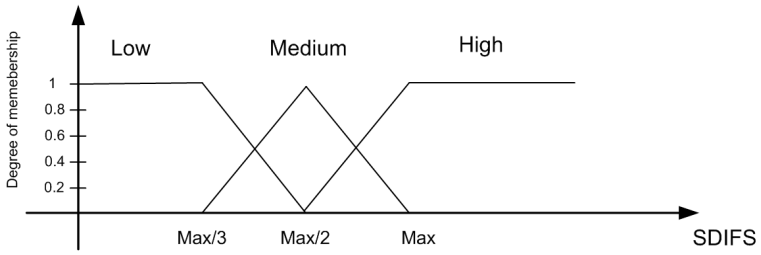


FIGURE 5.5: Example of the membership function for SDIFS

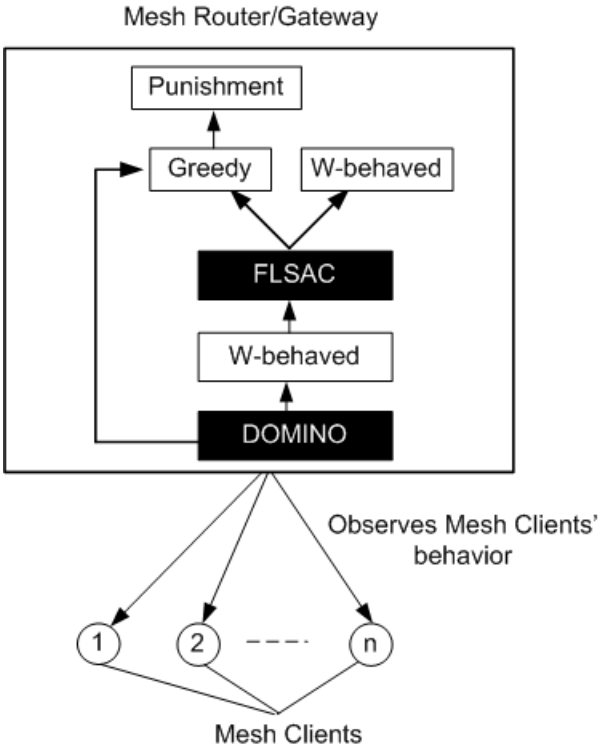


FIGURE 5.6: The overall functioning of our scheme

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

TABLE 5.2: Fuzzy rules of the formula : RES_1 = (BDEV $\wedge$ RTR)

Rule-based decision This step aims to use the rules established by the expert together with the knowledge acquired from the knowledge base (DCF protocol specifications) 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 [1] inputs/outputs and their fuzzy representation understood by the system.

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.

$$\begin{aligned} \beta_{P_1} \wedge \beta_{P_2} &\equiv min(\beta_{P_1}, \beta_{P_2}) \\ \beta_{P_1} \vee \beta_{P_2} &\equiv max(\beta_{P_1}, \beta_{P_2}) \end{aligned} \tag{5.5}$$

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, 0.5, **0.5**), RTR (0.7, **0.3**, 0) and S-DIFS (0, 0.4, **0.6**).

By applying the above rule, the result will be "the node is 20% Cheater" because the minimum of (0.5, 0.3, 0.6) is 0.3. The rules that fill our rule base are depicted in the tables 5.2 and 5.3, while the rules shown in table 5.4 are a combination of the decision of the last monitoring period with the output of the table 5.3.

These rules are inferred through 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 these two cheating techniques allow the cheater to increase its throughput significantly as compared to S-DIFS technique. Besides, they are more harmful in terms of the induced performance degradation, 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, which results in bandwidth under utilization.

[1] 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.

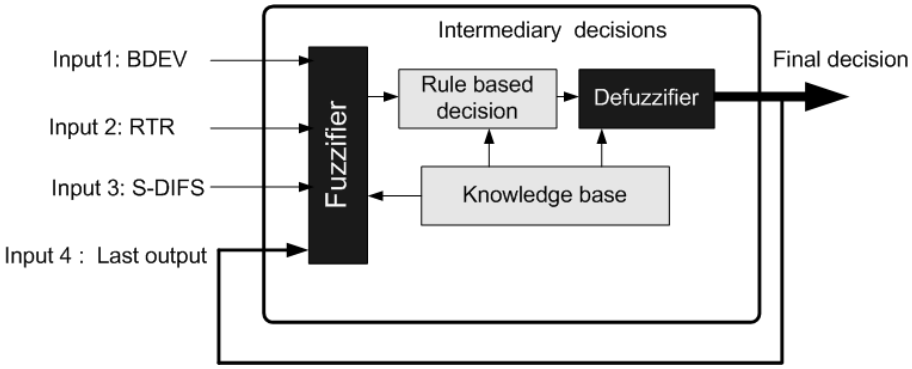


FIGURE 5.7: The main components of FLSAC

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

TABLE 5.3: 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 5.4: The final fuzzy decision of FLSAC : FDEC = (RES_2 $\wedge$ Last decision)

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 by computing the center of area of each fuzzy set. The CoG output value is given by the following formula.

$$\begin{aligned} Output_{crisp} &= \frac{\sum_{i=1}^{4} d_i \beta_i}{\sum_{i=1}^{4} \beta_i} \\ &= \frac{d_N \beta_N + d_{LS} \beta_{LS} + d_{HS} \beta_{HS} + d_C \beta_C}{\beta_N + \beta_{LS} + \beta_{HS} + \beta_C} \end{aligned} \tag{5.6}$$

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 3.

Algorithm 3 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 current 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 4 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

5.5.1.3 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; $m < N$). Moreover, the MR can also delay the delivery of packets

Parameters	Values
Area	1500m × 1000m
No. of Wireless MCLs	8 (light load) and 29 (heavy load)
MAC protocol	IEEE 802.11b
Transmission range	250 m
No. of Greedy nodes	1, 3, 7 and 10 (4 scenarios)
FLSAC	running on MR (Mesh Router)
Switching scheme	Random (see the flowchart in Figure 5.2)
Traffic type	CBR
Data rate	11mbps
CBR packets size	512 bytes
Monitoring period	10s
No. of Simulation epochs	20

TABLE 5.5: Simulation settings

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. In 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 4. 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 [9]. 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.

5.5.2 Simulation results

In this section, we report the simulation results of FLSAC which has been implemented in OPNET 14.0 network simulator [59]. The simulation scenarios and settings are summarized in table 5.5.

5.5.2.1 Simulation environment

For the simulation environment, we consider a WMN similar to the topology shown in Figure 2.2, 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 technique. The results are averaged over 20 simulations, with 120 seconds each. To outline the impact of these misbehavior strategies, we configure our simulation as follows; first, the cheater node launches each cheating strategy alone and then carries out an adaptive attack by switching dynamically between these different strategies.

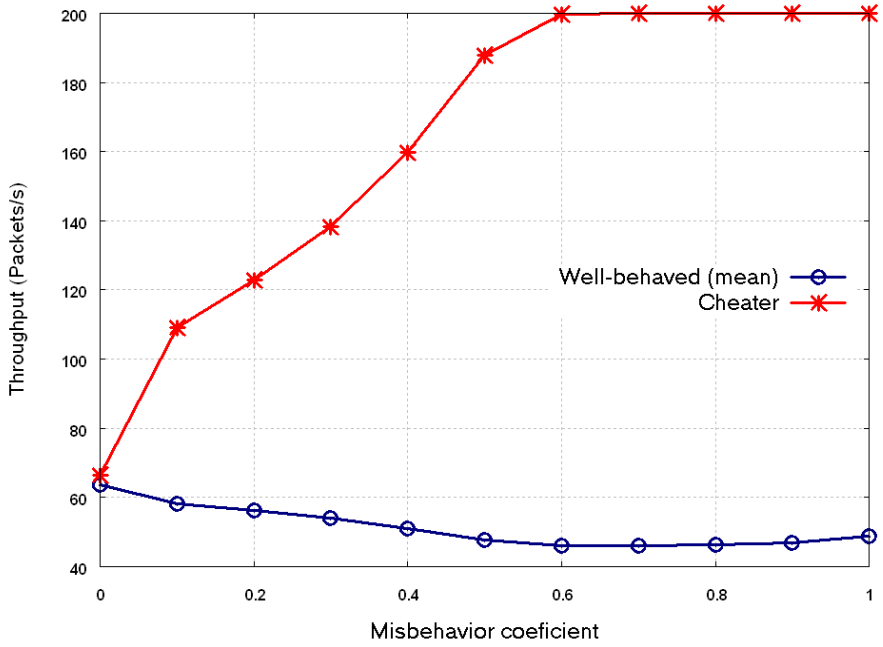


FIGURE 5.8: Impact of backoff manipulation on throughput

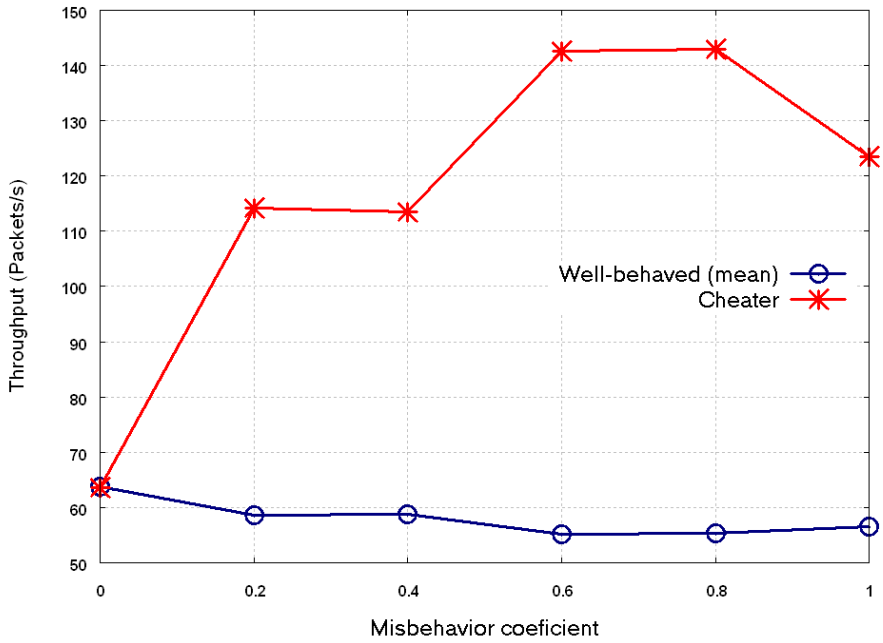


FIGURE 5.9: Impact of DIFS value reduction on throughput

Next, we compare the throughput of the cheater node with the mean [2] of well behaving nodes' throughput. 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. Notice that we use the misbehavior coefficient as a metric to measure the misbehaving level of the cheater node, in each strategy.

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

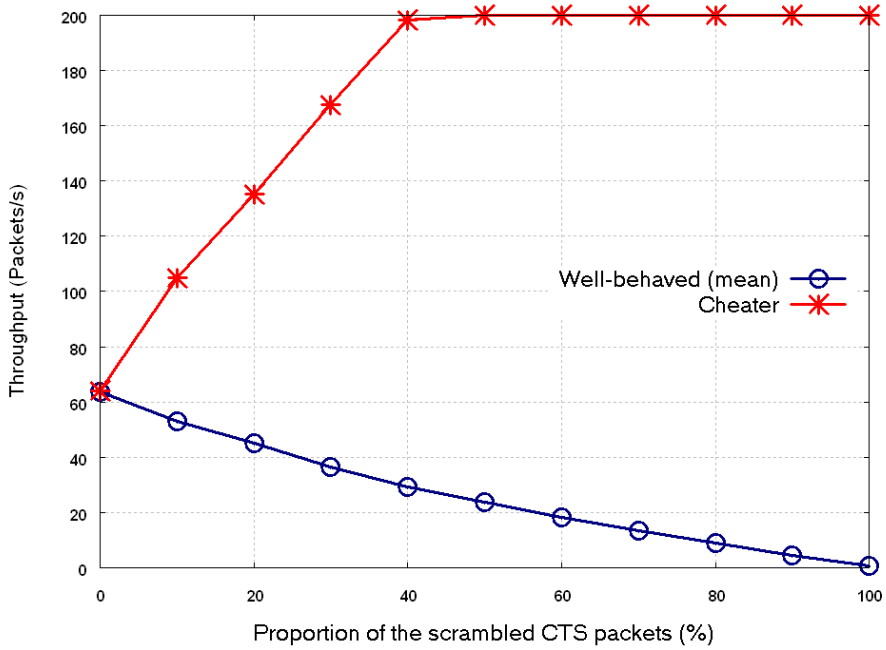


FIGURE 5.10: Impact of the proportion of scrambled CTS packets on throughput

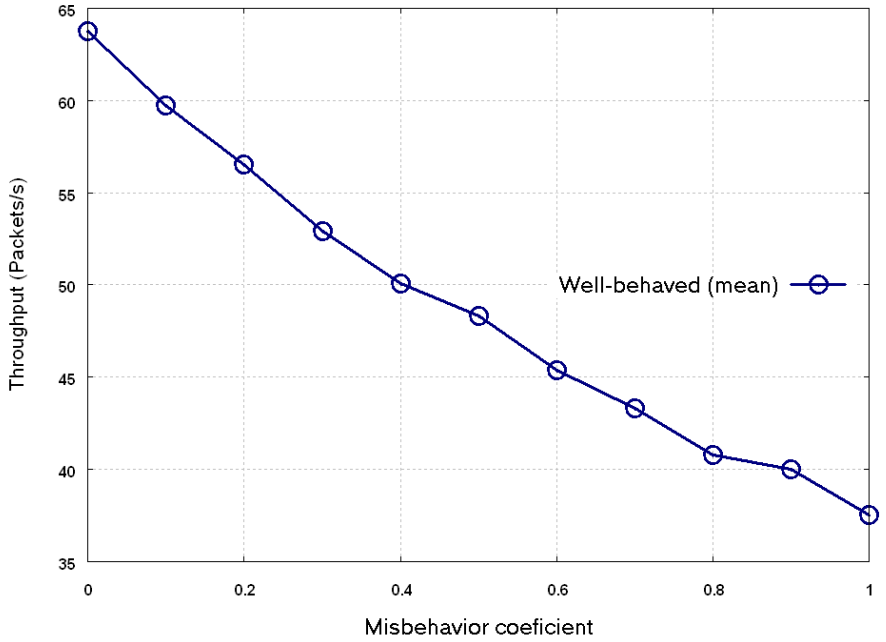


FIGURE 5.11: Impact of the NAV value inflation on throughput

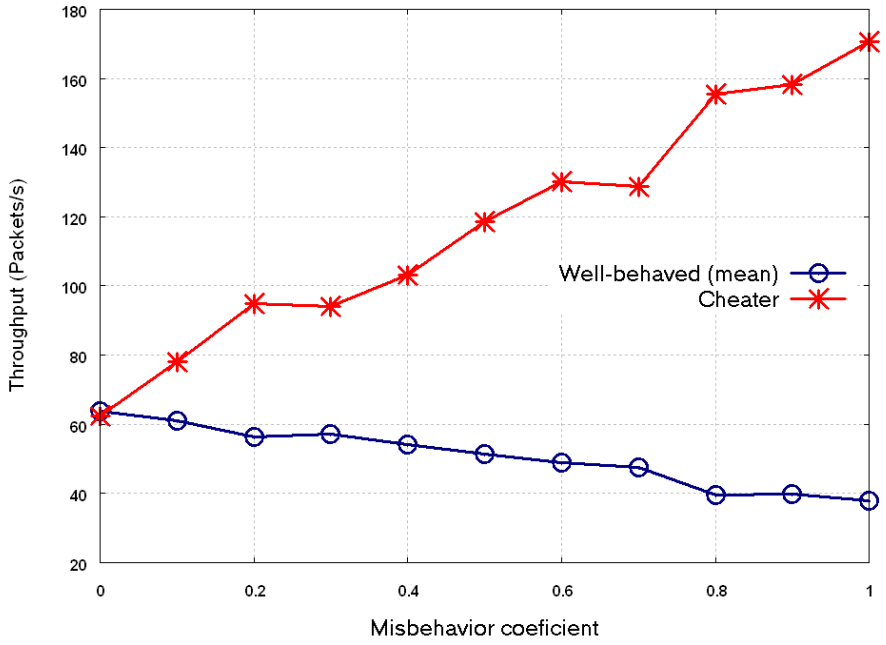


FIGURE 5.12: Impact of the adaptive cheater on throughput

5.5.2.2 Discussion of simulation results

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

As it is known, the backoff manipulation technique (BDEV) allows the cheater node to gain a considerable extra bandwidth. The throughput of the cheater increases with the increase of the misbehavior coefficient. The maximum load (200 packets) is achieved when the misbehavior coefficient is 0.6 as shown in Figure 5.8. 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 monopolize the medium and transmit all its buffered packets, since it chooses very small backoff values from the interval [0, $(1 - MC) \times CWmin$].

The Figure 5.9 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 $\times$ slot time}(equal to SIFS) to the higher limit of DIFS, as specified in IEEE 802.11b. The cheater's throughput increases with the increase of the 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 short frames (CTS or ACK). Similarly, 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 technique.

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 5.10. On the other hand, the bandwidth share of the cheater increases as the proportion of the scrambled packets increases. Moreover, 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 gaining sufficient bandwidth for its own traffic.

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

When the cheater amplifies the duration of its transmission, its neighbor 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 them can transmit because their NAV values are not expired yet. 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 less traffic load then a spatial reuse reduction is resulted. As depicted in the Figure 5.11, the throughput of the well-behaved nodes decreases when a cheater node launches this cheating technique and it is proportional to the value of misbehavior coefficient.

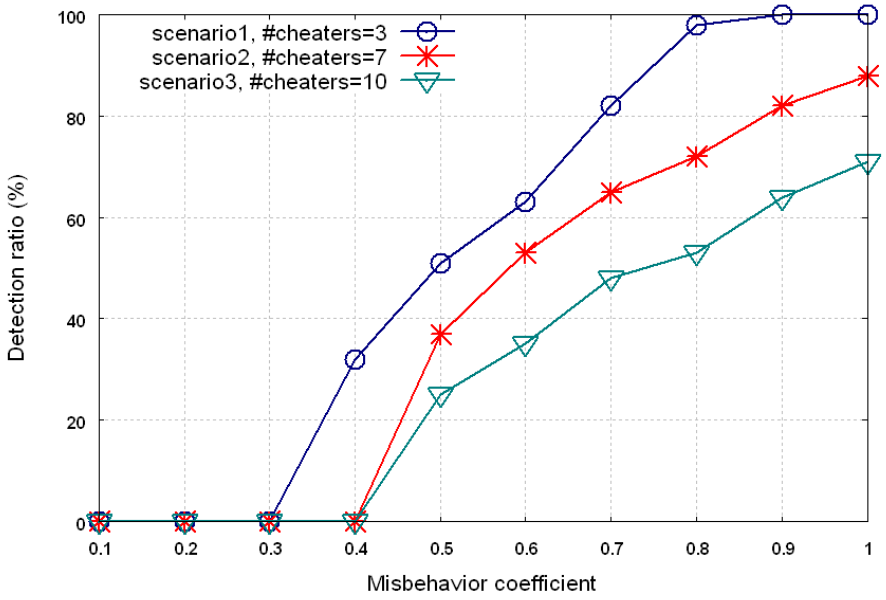
To implement the adaptive cheater behavior, we generate 3 random values within the interval [0, $thresh_i \times MC$], at the beginning of each monitoring period. These values represent the number of times the cheater can deviate from the protocol, in each strategy i, without being detected by DOMINO (see the flowchart depicted in Figure 5.2 for more details).

Figure 5.12 reveals that the cheaters applying this adaptive technique gain a considerable bandwidth as compared to well behaving nodes. The higher the misbehavior coefficient is, the higher the bandwidth that the cheater gains. However, this bandwidth is lower than that earned by a full cheater launching a single strategy solely.

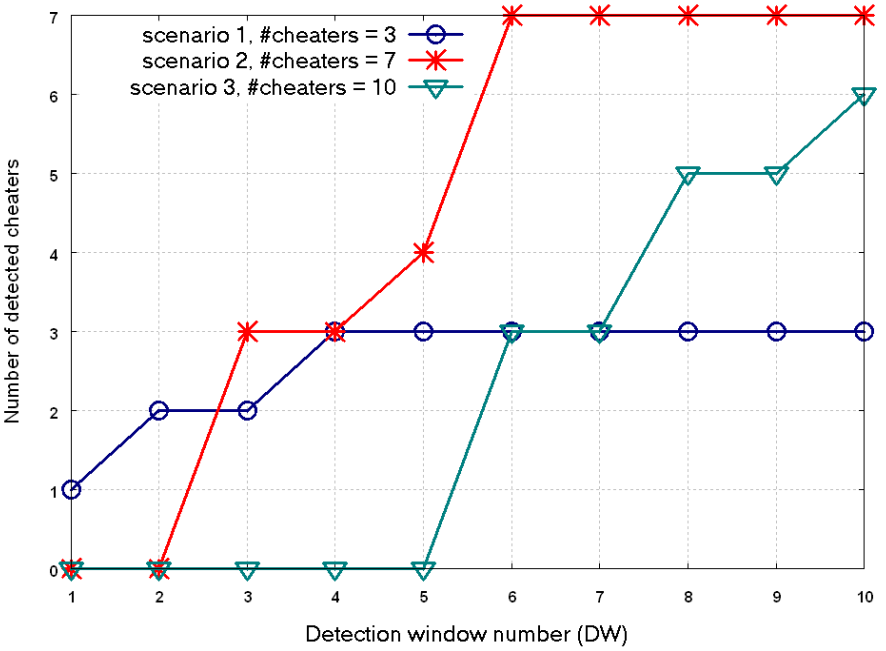
In what follows, we evaluate the efficiency of our scheme in terms of detection ratio and speed. To do so, we increase the number of wireless clients to 29 and the number of cheaters to 10. Moreover, we generate three scenarios through which we vary the number of cheaters to 3, 7 and 10 in order to figure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

Figure 5.13(a) reveals that the detection ratio is proportional to the MC of the cheater and 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 deviations are not large enough to accuse them as cheaters. However, their gain in bandwidth, in this case, is moderate as illustrated previously in the Figure 5.12. When the cheaters increase their MCs, 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 (i.e. detection speed) of FLSAC, we configure all the cheaters to start misbehaving simultaneously, and then record the number of detection windows after which they have been detected. The MC as well is set to the highest value. As shown in Figure 5.13(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, which prevents the MR from monitoring the BDEV and S-DIFS. Moreover, the number of retransmissions of cheaters as well as the well behaving nodes will appear close to each other since the cheater nodes also experience collisions of their own frames.



(a) Detection ratio versus misbehavior coefficient



(b) Detection speed of FLSAC

FIGURE 5.13: FLSAC's performance

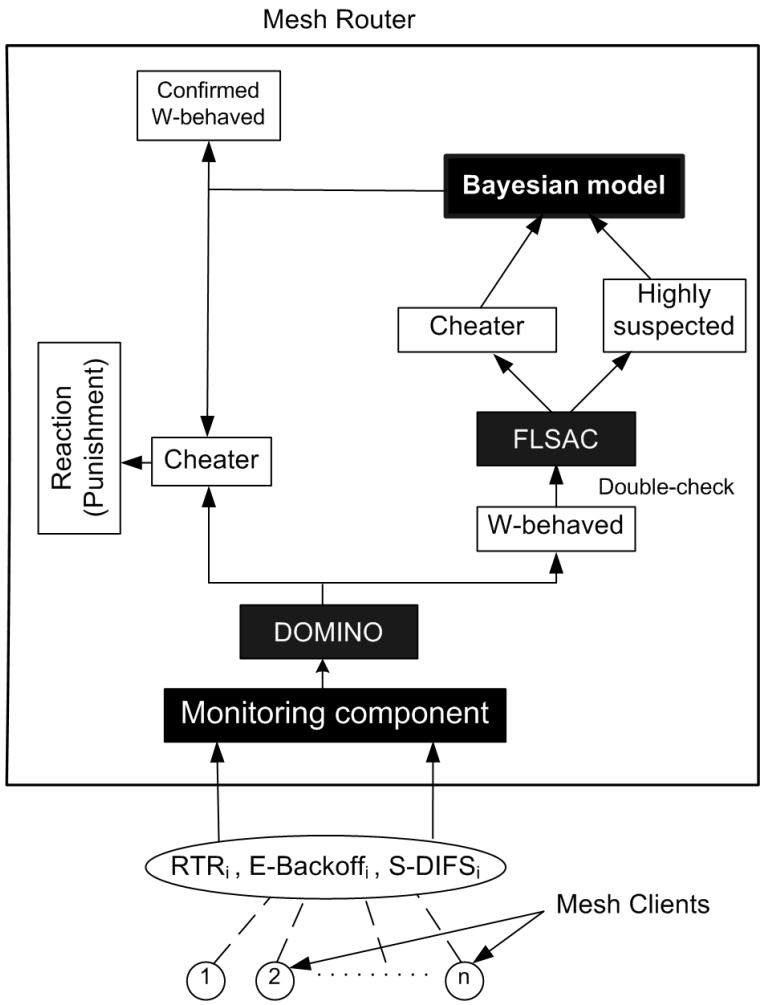


FIGURE 5.14: The integration of the Bayesian model with FLSAC

5.6 Enhanced FLSAC using Bayesian model

In what follows, we present, in details, our Bayesian model and provide a filter to evaluate its accuracy.

5.6.1 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 MCL is an adaptive cheater although DOMINO has classified it as well behaved, as depicted in Figure 5.14. To do so, we first classify the observed backoff values into two sets, legitimate and suspected, and then calculate a prior probability that this MCL is a cheater according to this classification. Afterwards, we use this probability along with the collected statistics regarding the different parameters that characterize the MAC protocol to compute the final cheating probability. This probability is then compared to a threshold value α, dynamically updated by the MR, in order to classify this MCL as cheater or honest. Notice that α is updated according to the variation of MCLs' density around the MR and the collision rate.

5.6.2 Model description

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

- the observed idle slots between consecutive transmissions from the same MCL.
- the observed idle slots between two transmissions from the same MCL interspersed by other transmissions from its neighbors.
- the observed retransmissions rate of each MCL: this metric is calculated through the comparison of the number of failed transmissions of a given MCL with the average of that of the other MCLs 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 MCL 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 MCL) that the cheater MCL has waited before decrementing its backoff.

In the following sections, 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 MCL (E-Backoff), whereas p_3 defines its retransmission rate (RTR).

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 MCLs; 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 ignore this parameter in our work.

Computation of the cheating probability

In our scheme, we did an earlier classification of the acquired observations of the parameters p_1 and p_2. For each observed value of these two parameters of a given MCL, 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 $\frac{\sum_{k=1}^{j-1} obs_k(p_i)}{(j-1)}$ - $obs_j(p_i) > \delta$ then the counter of the suspected observations cpt_1 (p_i) is incremented by 1, otherwise it is considered as legitimate observation. Notice that the initial value of j is 3 and the observations obs_1 and obs_2 are considered as legitimate. Notice also that the threshold δ is computed according to the confidence interval of the empirical average of the observed backoff values. We assume that at the end of a given monitoring window, the MR has collected N_1 and N_2 observations on the parameters p_1 and p_2, respectively. Thus, we calculate a prior probability that this MCL is suspected to be cheater ($\mathcal{P}$ (susp)) and a prior probability that it is a legitimate MCL that obeys the MAC protocol rules ($\mathcal{P}$ (leg)), as follows.

$$\mathcal{P}(susp) = \frac{\sum_{i=1}^{2} cpt_1(p_i)}{N_1 + N_2} \tag{5.7}$$

$$\mathcal{P}(leg) = 1 - \mathcal{P}(susp) \tag{5.8}$$

Notice that we didn't take into account p_3 and p_4 for calculating those prior probabilities because both of these two parameters are a number which is incremented each time a retransmitted frame is received or a S-DIFS value is observed, respectively.

How to calculate δ Let [A, B] be the confidence interval of the n observations taken by the MR, then

$$A = \overline{obs} - (1.96 \times \frac{\sigma(obs)}{\sqrt{n}}) \tag{5.9}$$

$$B = \overline{obs} + (1.96 \times \frac{\sigma(obs)}{\sqrt{n}}) \tag{5.10}$$

such that

$$\overline{obs} = \frac{1}{n} \times \sum_{k=1}^{n} \times p_k \tag{5.11}$$

and

$$\sigma^2(obs) = \frac{1}{n-1} \times \sum_{k=1}^{n} (obs_k - \overline{obs})^2 \tag{5.12}$$

Since our focus is on identifying the suspected backoff values which are smaller than the average of the previous values, then we only consider the lower boundary of the interval [A,B] to calculate the deviation threshold δ, as given below

$$\delta = \overline{obs} - A \tag{5.13}$$

Based on the two probabilities calculated in Equations 5.7 and 5.8, and by applying Bayes theorem, we calculate the probability that a given MCL is cheater as described below.

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

$$\mathcal{P}(A/B) = \frac{\mathcal{P}(B/A).\mathcal{P}(A)}{\mathcal{P}(B)} \tag{5.14}$$

gives the conditional probability of A knowing B. This formula can be applied to our problem as follows; if a certain MCL is represented by the vector $(p_1, \ldots, p_m)$, (m is equal to 4 in our case) then the probability that this MCL is an adaptive cheater is expressed as

$$\mathcal{P}(che/p_1, \ldots, p_m) = \frac{\mathcal{P}(p_1, \ldots, p_m/susp) \cdot \mathcal{P}(susp)}{\mathcal{P}(p_1, \ldots, p_m/susp)(\mathcal{P}(susp) + C)} \tag{5.15}$$

where

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

and p_i indicates that the MCL has been observed cheating by manipulating the MAC parameter of index i.

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

$$\mathcal{P}(che/p_1,\ldots,p_m) = \frac{\mathcal{P}(p_1/susp)\cdot\ldots\cdot\mathcal{P}(p_m/susp)\cdot\mathcal{P}(susp)}{C_1+C_2} \tag{5.16}$$

such that

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

Let us now consider the following rules issued from Bayes theorem

$$\mathcal{P}(p_i/susp) = \frac{\mathcal{P}(susp/p_i)\cdot\mathcal{P}(p_i)}{\mathcal{P}(susp)}$$
$$\mathcal{P}(p_i/leg) = \frac{\mathcal{P}(leg/p_i)\cdot\mathcal{P}(p_i)}{\mathcal{P}(leg)}$$

By applying those rules on the Equation 5.16, we obtain

$$\mathcal{P}(che/p_1,\ldots,p_m) = \frac{\frac{\mathcal{P}(susp/p_1)\cdot\mathcal{P}(p_1)}{\mathcal{P}(susp)}\cdot\ldots\cdot\frac{\mathcal{P}(susp/p_m)\cdot\mathcal{P}(p_m)}{\mathcal{P}(susp)}}{d_1\cdot\ldots\cdot d_m\cdot\mathcal{P}(susp)+\overline{d_1}\cdot\ldots\cdot\overline{d_m}\cdot\mathcal{P}(leg)} \tag{5.17}$$

with

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

As a result, we obtain the following formula

$$\mathcal{P}(che/p_1,\ldots,p_m) = \frac{\prod_{i=1}^{i=m}\mathcal{P}(p_i)[\mathcal{P}(susp/p_1)\cdot\ldots\mathcal{P}(susp/p_m)]}{[\mathcal{P}(susp)]^{(m-1)}\cdot\prod_{i=1}^{i=m}\mathcal{P}(p_i)[E+\overline{E}]} \tag{5.18}$$

where

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

and

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

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

$$\mathcal{P}(che/p_1,\ldots,p_m) = \frac{\mathcal{P}(susp/p_1)\cdot\ldots\cdot\mathcal{P}(susp/p_m)}{\mathcal{P}(susp/p_1)\cdot\ldots\cdot\mathcal{P}(susp/p_m)+R_1\cdot R_2} \tag{5.19}$$

such that

$$R_1 = (\frac{\mathcal{P}(susp)}{\mathcal{P}(leg)})^{(m-1)}$$

and

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

Therefore, the final cheating probability of a given MCL is calculated according to the Equation 5.19.

Now, this MCL which is represented by the set of parameters: $p_1, p_2, \ldots, p_m$ is classified as a cheater if

$$\frac{\mathcal{P}(che/p_1 \cdot \ldots \cdot p_m)}{1 - \mathcal{P}(che/p_1, \ldots, p_m)} > \lambda \tag{5.20}$$

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

Filter evaluation methodology

A filter performance is based on two parameters which are its accuracy (Acc), as defined in [86], and the error (Err = 1 - Acc), respectively defined as

$$Acc = \frac{N_{che \to che} + N_{hon \to hon}}{N} \tag{5.21}$$

$$Err = \frac{N_{che \to hon} + N_{hon \to che}}{N} \tag{5.22}$$

where $N_{x \to y}$ denotes the number of MCLs of class x which are erroneously classified in class y, and $N_{x \to x}$ represents the number of MCLs 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.

$$Wacc = \frac{\lambda N_{che \to che} + N_{hon \to hon}}{\lambda N_{che} + N_{hon}} \tag{5.23}$$

$$Werr = \frac{\lambda N_{che \to hon} + N_{hon \to che}}{\lambda N_{che} + N_{hon}} \tag{5.24}$$

where N_{che} and N_{hon} refer to the number of cheaters and honest MCLs, respectively.

To assess the performance of this filter we compare it to a non filtered network where every MCL is considered as honest, thus granting network access to every cheater MCL. 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:

$$Wacc^b = \frac{\lambda N_{che}}{\lambda N_{che} + N_{hon}} \tag{5.25}$$

$$Werr^b = \frac{N_{hon}}{\lambda N_{che} + N_{hon}} \tag{5.26}$$

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:

$$TCR = \frac{Werr^b}{Werr} = \frac{N_{hon}}{\lambda N_{che \to hon} + N_{hon \to che}} \tag{5.27}$$

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 MCLs correctly classified by the filter (i.e. Node Recall (NR)), and the precision of this filter when it classifies the MCLs as honest (i.e. Node Precision (NP)). These two metrics are defined as

$$NR = \frac{N_{hon \rightarrow hon}}{N_{hon}} \tag{5.28}$$

$$NP = \frac{N_{hon \rightarrow hon}}{N_{hon \rightarrow hon} + N_{che \rightarrow hon}} \tag{5.29}$$

The impact of NR and NP on the network performances is highly dependent on the filter context. Similarly to the parameter λ, 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.

Integration of the bayesian model with FLSAC

When we integrate our model with FLSAC, the output of this latter will define the classification criteria α used for identifying the cheater MCLs. Hence, for better understanding of the interaction between these two schemes we distinguish three different cases as follows:

- FLSAC's output reveals that the MCL 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 α is calculated as the addition of the current FLSAC's output and the third of the average of the last k outputs, where the decision was either normal or lowly suspected.
- if FLSAC judges that the MCL is cheater then α is assigned the value of FLSAC's output.

Algorithm 5 provides a pseudo code that explains how a MR combines both of FLSAC and the Bayesian model to make the final decision regarding the MCL's behavior.

5.6.3 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 MCL and MR are summarized in table 5.6.

We have conducted experiments using CBR traffic where a source MCL sends a CBR stream to a distant MCL that is attached to a different MR. In our configuration we vary the percentage of the cheater MCLs attached to the same MR from 20% to 40%. Notice that the switching scheme used by the cheater MCLs to alternate between the cheating techniques is similar to that used in [87]. We also vary the classification criteria α 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 5.7.

Algorithm 5 Interaction between FLSAC and our bayesian model

At the end of the monitoring window W_i;
if *(FLSAC_decision == (Normal* $\vee$ *L-susp))* **then**
  status(MCL_id)= honest;
else
  if *(FLSAC_decision == H-susp)* **then**
    $\beta = \frac{\sum_{j=1}^{k} FLSAC_{output}(Normal \vee L-susp)}{3 \cdot k}$;
    $\alpha = FLSAC_{output}$ (current window W_i) + β;
    if *(Bayes_prob(cheater)>=* α*)* **then**
      status(MCL_id) = cheater; //confirmed cheater;
    else
      Defer the decision till the end of the subsequent monitoring window (W_{i+1});
    end
  else
    $\alpha = FLSAC_{output}$;
    if *(Bayes_prob(cheater)*$\geq$ α*)* **then**
      status(MCL_id) = cheater;
    else
      Defer the decision till the end of the subsequent monitoring window (W_{i+1});
    end
  end
end

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 [59]

TABLE 5.6: Simulation settings

Figures 5.15,5.16,5.17 and 5.18 plot and compare the detection rates of DOMINO, FLSAC, B_Model and FLSAC+B_Model [3](FB_Model). Except DOMINO which, as expected, fails totally to detect the adaptive cheaters in all scenarios and with varying misbehavior coefficient, 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

[3]The notation FLSAC+B_Model refers to the scheme resulted from the integration of B_Model with FLSAC as described in section 5.6.2.

Scenario	Percentage of cheater MCLs per MR	α
1	20 %	0.55
2	30 %	0.65
3	35 %	0.75
4	40 %	0.85

TABLE 5.7: Scenarios setting

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 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.

From the curves plotted in Figure 5.15, we observe that the B_Model outperforms the other schemes as the value of α is small enough to let the MR easily recognize the cheater MCLs. So, it ensures that most of the cheater MCLs are detected even if they are slightly deviating from the standard, at the expense of some wrong accusation of well behaved MCLs.

The Figures 5.16, 5.17 and 5.18 divulge that the increase of α leads to a sharp decrease of B_Model's detection rate since the large value of α allows only detection of a small portion of the deviating MCLs, 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 α, (ii) the new defined classification criteria for the MCLs that have been classified as H-susp by FLSAC ensures their detection if they are misbehaving, hence those MCLs cannot escape from FB_Model as they have done with FLSAC (see Algorithm 5).

The histogram plotted in Figure 5.19 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 5.6.2. As we can see from this histogram, the higher the value of α is the lower accuracy of B_Model because when α rises the interval [0.5, α] gets larger. Thus, the cheater MCLs 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 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 MCLs are cheaters in our simulations).

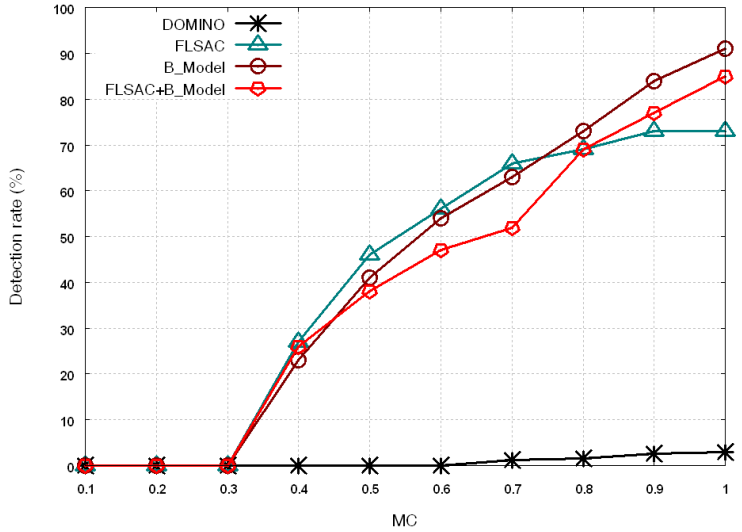


FIGURE 5.15: Detection rate versus the variation of the MC: scenario 1

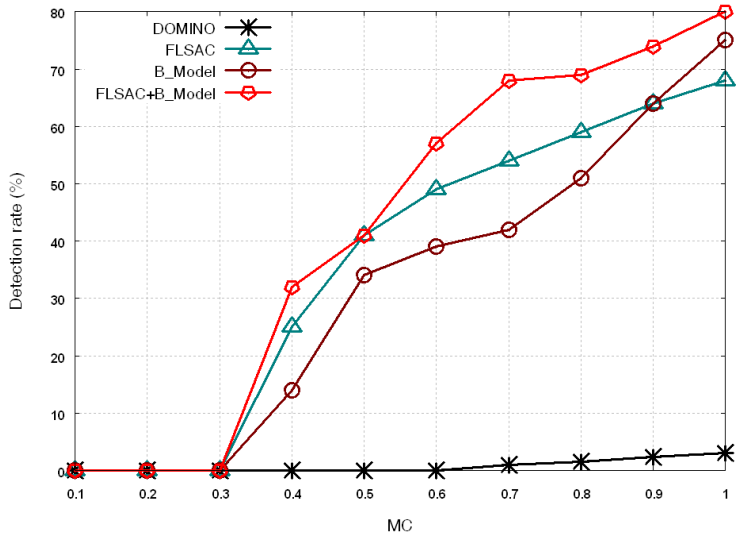


FIGURE 5.16: Detection rate versus the variation of the MC: scenario 2

5.6.4 Conclusion

In this chapter, we have studied the adaptive cheater behavior in WMNs and proposed two solutions to cope with it. The first solution, dubbed FLSAC, is based on a fuzzy logic controller that merges the observations of three different metrics to conclude whether a given MCL is greedy or not. The main advantage of FLSAC as compared to the existing schemes is its ability to detect the adaptive cheaters that apply a bunch of misbehavior techniques and switch intelligently among them to evade detection. According to the simulation results, FLSAC can reduce significantly the negative impact of the adaptive cheaters. Moreover, it is lightweight in terms of response speed. Despite that, FLSAC has a modest detection rate and accuracy, particularly in a WMN with large number of adaptive cheaters. To circumvent such weakness, we have developed a Bayesian scheme that allows us to compute the probability that a given MCL is being adaptive cheater and then make the right decision accordingly. This scheme is implemented at the mesh router which is the responsible for collecting the values of certain parameters used by the MCLs to access the wireless medium. Based on these observed values, our scheme calculates the

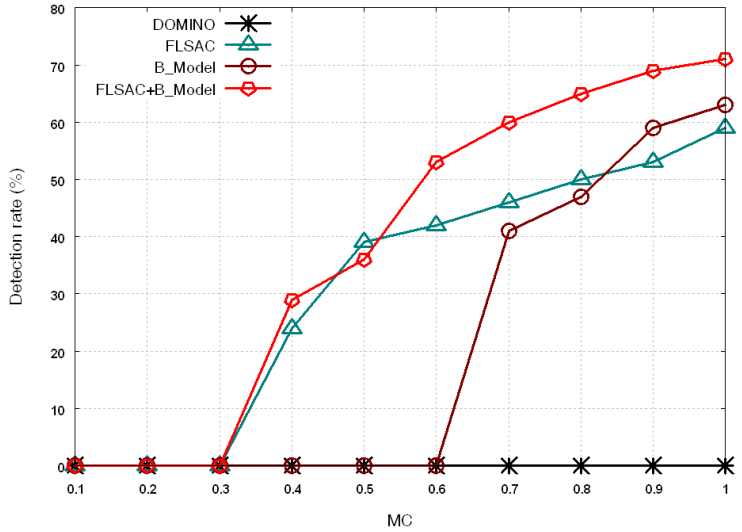


FIGURE 5.17: Detection rate versus the variation of the MC: scenario 3

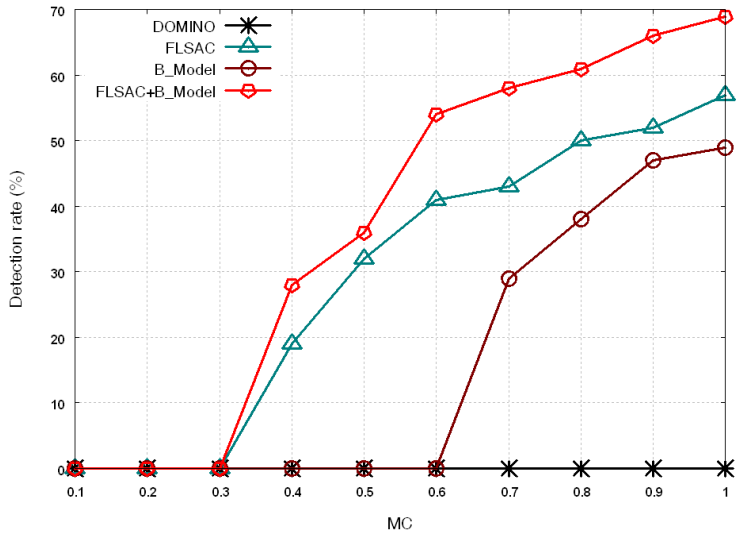


FIGURE 5.18: Detection rate versus the variation of the MC: scenario 4

cheating probability and then combines it with FLSAC's output to make the final decision regarding a given MCL's behavior. According to the simulation results, the Bayesian scheme combined with FLSAC shows better efficiency in terms of detection rate and accuracy even when the WMN is dominated by the cheaters, as compared to DOMINO, FLSAC and the Bayesian scheme solely. Therefore this proves that Bayes theory stills an efficient tool that can be exploited to defend against other misbehavior in WMNs.

Throughout this chapter we have investigated the adaptive cheater behavior in WMNs where we have always a trustworthy entity to which we can assign some crucial tasks, such as MCLs' behavior monitoring, to ensure fair-share of bandwidth and detect any deviating MCL. However, this property is absent in other wireless networks such as MANETs in which no predefined trustworthy entity exists. Therefore, the herein discussed solutions are not applicable to such networks. Hence, we devote the next chapter to first define new characterization of the greedy behavior in MANETs and then present a distributed countermeasure rather than the centralized ones proposed in this chapter.

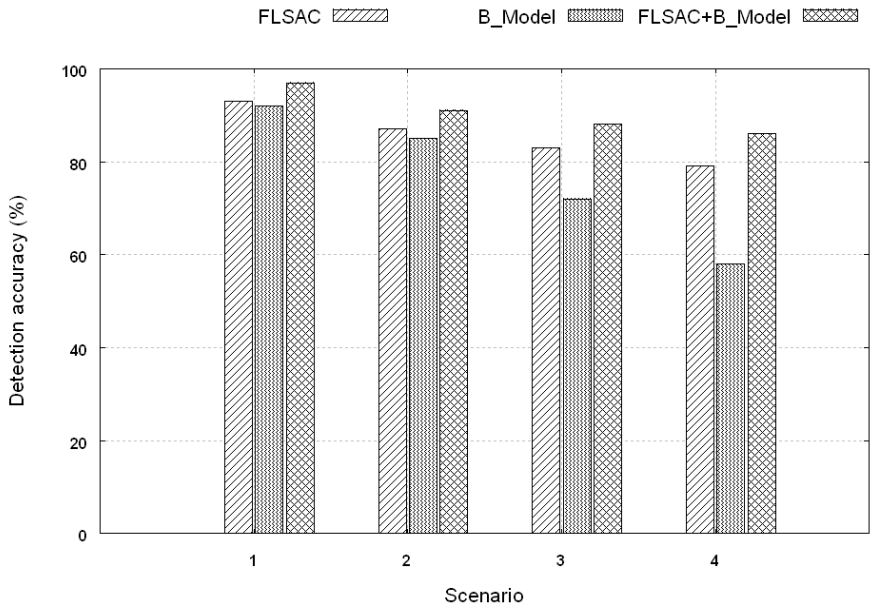


FIGURE 5.19: Detection accuracy in different scenarios

Chapter 6

Greedy Behavior in MANETs

As shown in the previous chapter, MAC layer misbehavior still an open issue that needs more investigation from the research community, in WMNs as well as in any other network based on IEEE 802.11 MAC protocol. It is obvious that the network characteristics and constraints may affect the way the greedy or cheater node behaves to increase its bandwidth share on the detriment of its neighbors. In a network like MANETs, the greedy node needs the service of its neighbors to relay its packets towards their destinations. Thus, it cannot monopolize the medium and prevent them from accessing it. Therefore, the greedy node is obliged to carefully design new technique to misuse the MAC protocol rules but still getting its packets forwarded and increasing its bandwidth share. So, what is the best strategy to be applied by the greedy node to satisfy its greediness and keep the performance of its applications reasonable? The answer to this question is provided throughout this chapter.

One of the techniques used by the greedy nodes is the selection of a small backoff value or setting it always to zero and thus accessing the medium without counting down any backoff timer. Hence, to face this misuse, particularly in the case of adaptive cheater where we cannot collect enough samples to make the correct decision, we propose in the second part of this chapter a new backoff selection scheme rather than the BEB scheme. This scheme will allow the neighbors of the greedy node to recalculate the backoff value that it has chosen and thus detect any deviation from this value.

6.1 New strategy of the greedy behavior in MANETs

6.1.1 Introduction

While the problem of greedy behavior at the MAC layer has been widely explored in the context of wireless local area networks, its study for multi-hop wireless networks still almost unexplored and unexplained problem.

Indeed, in a wireless local area network, an access point mostly forwards packets sent by wireless nodes over the wired link. In this case, a greedy node can easily get more bandwidth share and starve all other associated contending nodes by manipulating intelligently MAC layer parameters. However, in wireless ad hoc environment, all packets are transmitted in a multi-hop fashion over wireless links. In this case, an attempting greedy node, if it behaves similarly as in a WLAN, trying to starve all its neighbors, then its next hop forwarder will be also prevented from forwarding its own traffic, which leads obviously to an end to end throughput collapse.

In this chapter, we show that in order to have a more beneficial greedy behavior in wireless ad hoc network, a node must adopt a different approach than in WLAN to achieve a better performance of its own flows. Then,

we present a new strategy to launch such a greedy attack in a proactive routing based wireless ad hoc network. A detailed description of the proposed strategy is provided along with its validation through extensive simulations. The obtained results show that a greedy node, applying the defined strategy, can gain more bandwidth than its neighbors and keep the end-to-end throughput of its own flows highly reasonable.

6.1.2 Greedy nodes' classification

The misbehaving nodes in MANETs can be classified according to the adopted strategy to launch the attack and the extent of the induced harm. Hence, the following classes can be identified (see Figure 6.1):

1. **Indirect greedy node**: which aims to increases its bandwidth by launching a cross-layer attack targeting the routing protocols in order to decrease the number of contending nodes around it, and then increases its chance to frequently access the medium. To this end, it may either increase its SIFS value to cause RTS timeout at the sender node or deny response to the received RTS frames.
2. **Direct greedy node**: which manipulates the medium access parameters such as backoff, DIFS or jams the CTS/ACK packets of its neighbors. This class can be further divided into three sub-classes as follows:
   - **Malicious greedy node**: which aims to disrupt the ongoing communications in its neighborhood and cause damage to network performance without seeking for any benefits. It can even send fake data packets to monopolize the medium, under the assumption that it is equipped with a permanent energy supply.
   - **Rational & Selfish (uncooperative) greedy node**: the greedy node here wants to increase the throughput for its own traffic flow and decrease its end-to-end delay. However, it affects the performance of the crossing flow by delaying it and releasing the medium for the next hop of its flow to forward the transmitted packets.
   - **Rational & Cooperative greedy node**: in this case, the greedy node's behavior is similar to the previous category; however it chooses some crossing flows which are of direct interest or conveying critical information in order to favor them during forwarding. The following situations justify this behavior of the greedy node.
     - In a battlefield the orders issued from the group leader are critical and need to be prioritized than the other flows.
     - In an emergency area, the packets sent by the rescuers which are inside the area of incident are critical and should be given high level of importance by the forwarder node.

6.1.3 Greedy behavior impact on network performance: WLAN versus MANETs

In this section we emphasize the major difference between the greedy behavior in WLAN and MANETs. In other words, we try to answer the following question: Are the damages induced by greedy nodes in WLAN and MANETs similar?

As illustrated in Figure 6.2, the destination of a flow in WLAN can be either a far away node or the one attached to the same access point (AP). In the former case, the source node of the flow f_1 tries to gain the entire bandwidth regardless of the decrease in its neighbor's throughput. This is due to the fact that its next hop (AP)

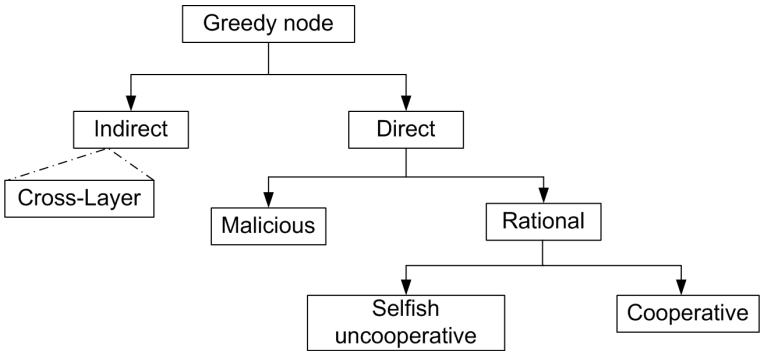


FIGURE 6.1: Classification of the greedy nodes's behaviors

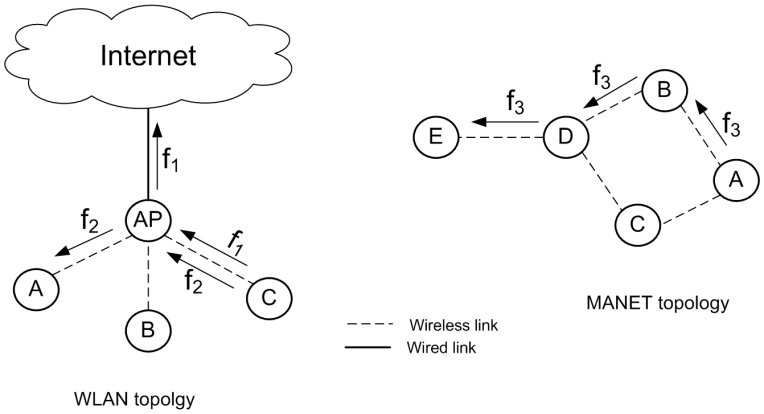


FIGURE 6.2: Greedy behavior: WLAN versus MANETs

forwards the packets of the flow f_1 through a wired link, independent from the wireless ones (no transmission conflict exist between those links). The flow f_2 is similar to the case of the flow f_3 in MANETs, any attempt of the flow's source node A or an intermediate node B to dominate the medium deprives its next hop from forwarding the received packets. Consequently, the flow's performance collapses sharply. Furthermore, the impact of this misbehavior may propagate to affect other flows crossing through the nodes in contention with the greedy node.

To illustrate this phenomenon related to radio wave propagation, let us consider the network topology given in Figure 6.3. In this figure, R_{tx} and R_{cs} represent the transmission and carrier sensing ranges of node A, respectively. The lightly shaded area represents the region which is not covered by RTS/CTS handshake between A and B. Note that any transmission initiated from a node within this region may not interfere with packet reception at node B as these nodes are out of its interference area, represented by the darker region which is delimited by the interference range R_I. Despite that, the nodes within the lightly shaded area have to differ their transmissions since they sense the medium busy due to node A's transmission. As a result, if the sender node A misbehaves and monopolizes the medium for a long duration, all the transmissions over the links where at least one node is within the lightly shaded area are delayed leading to an increase on the number of dropped packets and the end-to-end delay. Even the links (B,C) and (C,D) are negatively affected, which means that the greedy node A is increasing its throughput in the detriment of the quality of service requirements of its own traffic flow.

On the contrary of MANETs, the situation discussed above does not arise in WLAN environment since all the nodes are within the transmission range of the AP, therefore the increase of the greedy node's throughput does not affect the end-to-end delay of its traffic flow. As a conclusion, for a more effective greedy behavior the greedy node should choose an alternative strategy adapted to the constraints of MANETs environment.

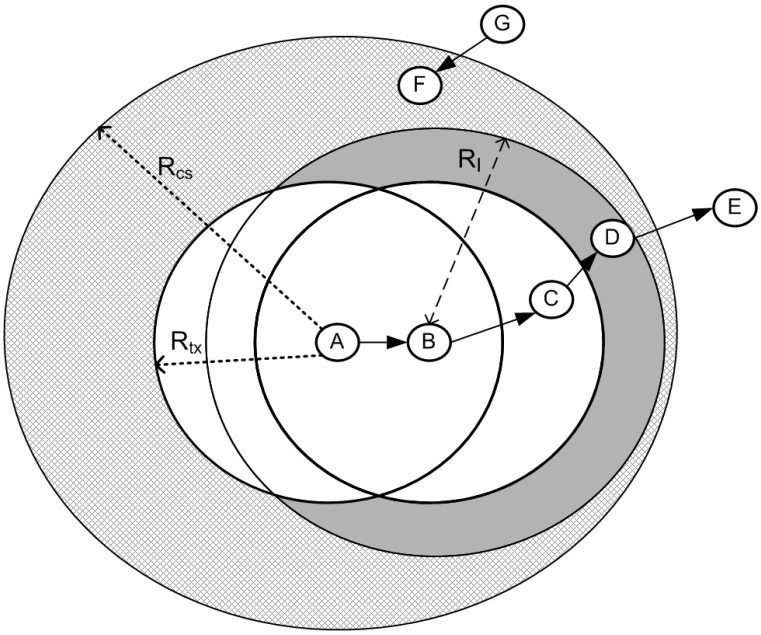


FIGURE 6.3: Propagation of greedy behavior's impact in MANETs

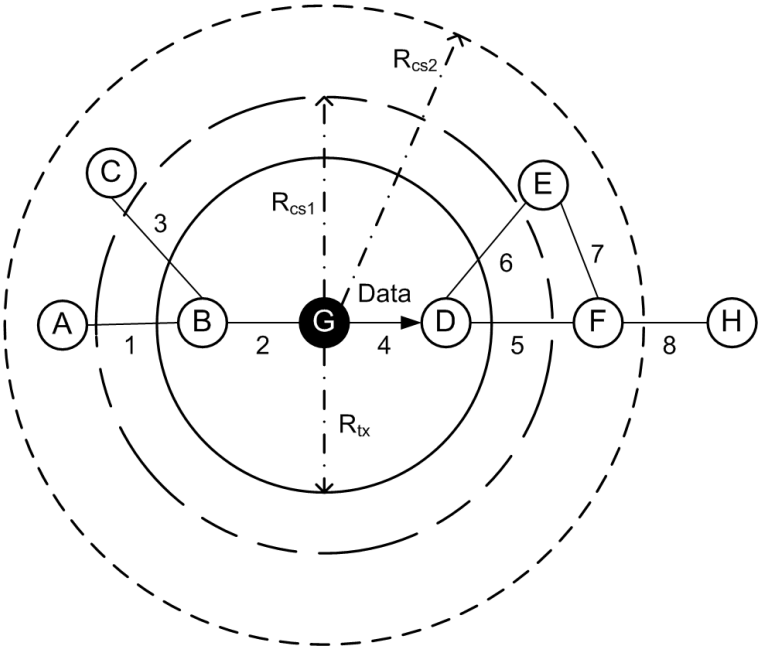


FIGURE 6.4: The connectivity graph

Illustrative example Figure 6.5 shows an example of the medium access frequency by a greedy node sending a traffic flow, its next hop node and the other neighbors in case where this greedy node tries to dominate the wireless medium. Therefore, this behavior leads to starving the greedy node's neighbors, including its next hop node, from retransmitting the received packets. If we consider the simple case where the destination node is two hops away from the sender (greedy) then the end-to-end delay (Ed_i) is computed as

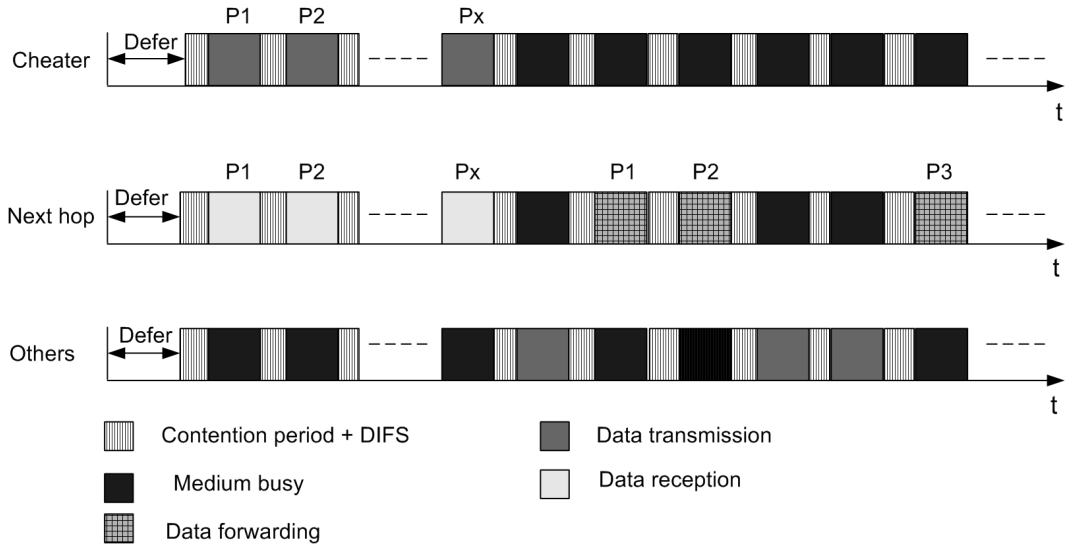


FIGURE 6.5: Example of bandwidth share among the greedy node, its next hop and the other neighbors nodes in the case where this greedy node is applying full greedy strategy (similar to WLAN case) in order to monopolize the medium

$$\begin{aligned} Ed_i &= 2T_{pi} + \sum_{j=i+1}^{x} (T_{pj} + CT_j) \\ &\quad + \sum_{j=1}^{i-1} (T_{pj} + CT_j) + T_1 \end{aligned} \tag{6.1}$$

where T_{pi} denotes the one hop transmission time of the packet pi, CT_j refers to the contention time spent by the node before accessing the medium and T_1 is the period during which the node is differing as one of its neighbors is transmitting. In the case where the flow's source node is behaving correctly the end-to-end delay Ed'_i is expressed as

$$Ed'_i = 2T_{pi} + \sum_{j=1}^{i-1} (T_{pj} + CT_j) + T_1 \tag{6.2}$$

then

$$Ed_i - Ed'_i = \sum_{j=i+1}^{x} (T_{pj} + CT_j) \tag{6.3}$$

This extra delay (Ed_i - Ed'_i) increases sharply as the number of packets (x) to be transmitted by the greedy node gets higher, leading to devastating impact on the traffic flow performance and violates all the QoS requirements.

6.1.4 Our greedy strategy description

In this section, we give the road map of the required steps for the greedy node to launch the greedy attack according to our strategy. First, we provide the basic assumptions of our scheme followed by a description of how the greedy

node constructs the conflict graph. Next, we show how to extract the bandwidth fair share of a node according to the conflict graph. Afterwards, we determine the maximum extra bandwidth that the greedy node can gain without negatively affecting its traffic flow performance. Finally, we present the algorithm used by the greedy node to launch the greedy attack and to ensure the accordance with the values computed in the previous step.

6.1.4.1 Main Assumptions

We give an overview of the assumptions used throughout the chapter. These assumptions constitute the core of our cheating strategy.

- A proactive routing protocol is used at the network layer to establish end-to-end routes such as OLSR [1][16] .
- Carrier sensing range (R_{cs}) is equal to more than twice of the transmission range (R_{tx}) [88].
- The nodes are distributed within the topology according to the Poisson process of parameter λ [89].
- We assume CBR traffic with fixed packets size S and the transmission rate offered by the underlying MAC protocol is β. Therefore, the number of time slots (γ) needed for the transmission of the packet payload is:

$$\gamma = \frac{(\frac{S}{\beta})}{\eta} \tag{6.4}$$

  where (η) is the duration of one time slot in μs.

- The length of a packet P_l is defined as the number of time slots required for its successful transmission and can be expressed as follows:

$$\begin{aligned} P_l &= \frac{(NAV - T_{DATA}) + DIFS + H_l}{\eta} + \gamma \\ P_l &= \frac{Duration + DIFS + H_l}{\eta} + \gamma \end{aligned} \tag{6.5}$$

  where

$$Duration = T_{RTS} + T_{CTS} + T_{ACK} + 3 \times SIFS$$

  Notice that T_{RTS}, T_{CTS} and T_{ACK} refer to the transmission time of RTS, CTS and ACK frames respectively, whereas H_l denotes the aggregate length of Physical, MAC and UDP headers defined as follows:

$$H_l = \frac{PHS + MHS + UHS}{\beta} \tag{6.6}$$

  where `PHS`, `MHS` and `UHS` are the size of PHY, MAC and UDP headers respectively.

[1] Notice that we can use any other routing protocol which provides the same topology view as OLSR.

6.1.4.2 Conflict graph construction

As a first step of our scheme, the greedy node constructs the contention flow graph with nodes within its R_{cs} to derive its predicted fair-share of bandwidth [90]. To this end, the greedy node analyzes the received information in Hello and topology control (TC) messages and constructs its conflict graph [91] accordingly. For example, node G in the topology shown in Figure 6.4 acquires the set of its 2-hop neighbors A, C, E and F from the Hello messages sent by nodes B and D, and it discovers its 3-hop neighbor H from the TC message sent by the node F which is multipoint Relay (MPR) of node H.

After acquiring the necessary information, the greedy node G constructs the conflict graph within its carrier sensing range, from which it extracts the set of maximal cliques. Since the topology information acquired from Hello and TC messages is partial, node G constructs this graph by considering the worst case scenario assuming the maximum number of contending links to compute the minimum bandwidth fair share. The number of maximal cliques is the key for determining the misbehaving threshold which will be discussed later. As shown in Figure 6.6, the conflict graph depends on the extent of the carrier sensing range of the greedy node, for which we distinguish two cases:

- R_{cs} is slightly larger than the transmission range R_{tx} (see Figure 6.4, R_{cs} = R_{cs1}), thus we have less contention between links and consequently the greedy behavior impact reduces. According to the set of maximal cliques shown in Figure 6.7(a), only a simultaneous transmission over the following pair of links is allowed:

(1,5)	(3,5)	(2,7)
(1,6)	(3,6)	(2,8)
(1,7)	(3,7)	
(1,8)	(3,8)	

- R_{cs} is greater than twice of the transmission range,
  $R_{cs} > 2 \times R_{tx}$ (see Figure 6.4, R_{cs} = R_{cs2}) which means that all the 2-hop neighbors of the greedy node G are within its carrier sensing range. Hence, only few links can be active for flow transmission at the same time as depicted in Figure 6.7(b), where only the pairs of links $(1,7)$, $(1,8)$, $(3,7)$ and $(3,8)$ are allowed to transport traffic flows simultaneously. As compared to the first case, the number of conflict between links raises leading to devastating consequences if one node doesn't obey the MAC protocol rules.

Time complexity for generating the maximal cliques Given that for N nodes we have at most $\frac{N(N-1)}{2}$ links which can be established between them. According to the algorithm by Tomita et al. [92], the worst-case time complexity for generating the set of maximal cliques from the graph constructed by those links is estimated to $O($ $3^{N/3})$.

6.1.4.3 Bandwidth fair-share estimation

Once the conflict graph is established and the set of maximal cliques is derived, the node G computes its fair share of bandwidth and the end-to-end throughput of its traffic flow. In order to compute these values, we assume each node has a nonempty buffer of packets ready to be transmitted at each time slot (saturation case). Hence, given a particular path relaying source and destination nodes, the end-to-end throughput capacity is defined as

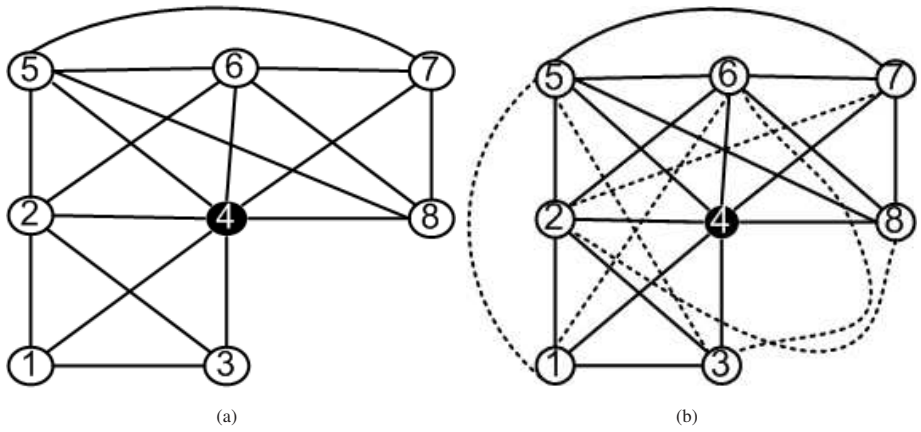


FIGURE 6.6: Conflict graph of the contending transmissions. (a) case of R_{cs} = R_{cs1}; (b) case of R_{cs} = R_{cs2}

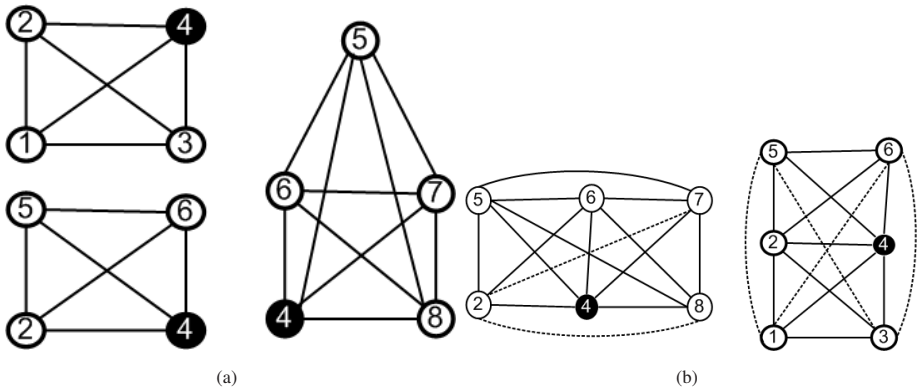


FIGURE 6.7: The set of maximal cliques. (a) R_{cs} = R_{cs1}, 3 maximal cliques whose sizes are 4, 4 and 5 respectively; (b) R_{cs} = R_{cs2}, 2 maximal cliques of 6 vertices each. Note that the dashed edges represent the new links created due to increase in R_{cs} from R_{cs1} to R_{cs2}

the minimum link throughput capacity B_i of this path. As the links in conflict with the link 4 ($G \rightarrow D$) are the bottleneck for any flow crossing them when G is cheating, we determine the end-to-end throughput E_{th} as the minimum capacity of these links. This means if the length of any path from the source node to destination node is n hops, the end-to-end throughput is computed only for the m hops ($m \leq n$) within the sensing range of the source node.

In order to calculate the throughput capacity of a single link we proceed as follows. From a sender node point of view, its sending link (link i) can be in one of the following three states: transmission state, channel busy state and channel idle state. the activities of the link i can be represented as follows:

- sc_t refers to the successful and collided transmission time.
- cl_t refers to the contending links' transmission time, which is the time during which the medium is busy.
- $idle_t$ represents the idle time of the link i, which is the time spent by the sender to count down its backoff timer.

We consider a long interval of time in [0, $Time$]. Let tr_i be the transmission time, within this interval, during which a node i (in steady state) transmits. tr_i includes the transmission time of RTS, CTS, DATA and ACK packets, and their corresponding SIFS periods. Moreover, the DIFS period and the times used up for retransmissions are also included. Therefore, the value of sc_t can be computed as follows:

$$sc_t = \lim_{time \to \infty} \frac{|tr_i|}{time} \tag{6.7}$$

where $|tr_i|$ refers to the length of this time interval. As a consequence of the carrier sensing property, any transmission within the R_{cs} of node i will change the state of the medium from idle to busy. The total time spent for these transmissions can be expressed as $\bigcup_{j \in S_1(i)} tr_j$ such that $S_1(i)$ denotes the set of the neighbors of the link i. Then cl_t can be defined as

$$cl_t = \lim_{time \to \infty} \frac{|\bigcup_{j \in S_1(i)} tr_j|}{time} \tag{6.8}$$

We consider that the channel is idle if there is no self transmission or neighbors' transmissions. Therefore, we have

$$idle_t = 1 - sc_t - cl_t \tag{6.9}$$

We assume that the sender node has nonempty queue of data packets (saturated load), therefore it counts down its backoff value whenever the channel is sensed to be idle. The DCF operation can be represented as a state diagram, where the transition between two states depends on channel state and the detection of any collision during transmission. Thus, the transmission of both of control or data packets is initiated only if the channel is confirmed to be idle, meaning that the DCF is an independent function running during the channel idle slots. So, if $Attempt_i$ represents the attempt rate per idle slots then sc_t can be formulated as

$$sc_t = idle_t \times Attempt_i \times \gamma \tag{6.10}$$

By knowing sc_t and the transmission probability τ, we can now calculate the link throughput B_i and the end to end throughput E_{th} as follows

$$B_i = sc_t \times (1-\tau)^{\lambda \pi R_{cs}^2 - 1} \times \beta \times \frac{\gamma}{P_l} \tag{6.11}$$

$$E_{th} = min(B_1, ..., B_{\lambda \pi R_{cs}^2}) \tag{6.12}$$

For more details about the computation of these values, the reader may refer to the works done in [93] and [89].

Since we construct a partial topology graph limited to links in which at least one of the extremity is within the carrier sensing range of node G, then the calculated fair-share B_i of a node i might be greater than the real one R_i. That is the reason why R_i can be expressed in function of Bi as

$$R_i = B_i \times \Phi \times \Psi \tag{6.13}$$

such that Φ is a factor used to adjust the estimated fair-share to the real one, where

$$0.5 \leq \Phi \leq 1$$

Moreover, as the greedy node constructs the topology graph by considering the maximum number of links relaying its 2-hop neighbors and between these nodes and its three hops neighbors, the computation of the fair-share is done based on links which might not exist. For that reason, the value Ψ, dubbed density factor, is used to increase this fair-share accordingly. This value is determined upon the following criterions:

- Nodes' density in the neighborhood of the greedy node.
- The number of MPR nodes selected by the greedy node; if this number is small and the density of nodes within its carrier sensing range is high then it is more likely to have more links between nodes, and consequently Ψ can be assigned a value close to 1. On the other hand, if the MPR set is large and the nodes' density is mediocre then the value of Ψ can be increased more than the previous case.

The value Ψ is expressed as

$$\Psi = 1 + \frac{|MPR\_set|}{|S_1 \cup S_2|} \tag{6.14}$$

where S_1 and S_2 denote the sets of node G's 1-hop and 2-hop neighbors respectively.

Remark In our proposed strategy, one may argue that a node can request its one or 2-hops neighbors for exchanging topology information in order to get the complete view of the network. However, such an action makes the node suspicious which may facilitate its detection if any anti MAC layer misbehavior system is deployed. Moreover, none of its neighbors will respond to these requests since they are not considered as proper operations of the routing protocols. As a consequence, our proposed algorithm is more secure and realistic since it depends only on the information gathered locally by the greedy node from the legitimate exchange of control packets.

6.1.4.4 Misbehaving Threshold Computation

In this section, we define an upper bound of the extra bandwidth earned by the greedy node, dubbed misbehaving threshold. Any greedy node overtaking this threshold will experience a decrease of its own flows' performance (in terms of end-to-end throughput and delay).

As known, in the case where fair-share is held amongst the contending nodes, the greedy node gets R_i of bandwidth. When the greedy node misbehaves, its share is R_i + B_g which means that it acquires B_g of extra bandwidth share as a result of its mischief. So, for a rational greedy node, the value B_g should satisfy the following condition

$$\begin{aligned} \frac{[B - (R_i + B_g)]}{(N-1)} &> \alpha \times E_{th} \\ (B - R_i - B_g) &> (N-1)\ \alpha \times E_{th} \end{aligned}$$

therefore

$$B_g \leq B - R_i - (N-1)\ \alpha \times E_{th} \tag{6.15}$$

such that, N is the average number of nodes within the carrier sensing range, R_{cs}, of the greedy node, which can be expressed as
$N = \lambda \pi R_{cs}^2$. B is the total bandwidth available and E_{th} is the estimated end-to-end throughput of the ongoing flow calculated according to the formula given in Eq. 6.12.

The reason of using the condition above is the fact that any adopted misbehaving strategy which reduces the mean [2] of the greedy node neighbors' throughput below the value of E_{th} has also a negative impact on its own flow's performance. Hence, the rational greedy node has to ensure that B_g fulfils the condition given in Eq. 6.15 in order to satisfy the QoS requirements of its flow.

Notice that the value α is used to adjust the extra bandwidth gain of the greedy node with respect to the topology of the bottleneck area (the area covered by R_{cs}) and the contention flow graph. It can be expressed as

$$\alpha = \frac{\sum_{i=1}^{N} MC_i}{N \times cl} \tag{6.16}$$

where MC_i denotes the number of maximal cliques to whom the link $i \leftrightarrow j$ belongs such that the node i is either sender or receiver over this link, and cl is the total number of the maximal cliques in the conflict graph.

6.1.4.5 How to launch our greedy strategy?

Once all the parameters defined in the previous steps are computed, the greedy node G carries out the two misbehavior techniques, described below, to achieve its goal.

It first selects a small Backoff value in order to gain more bandwidth within its allowed threshold. Then, it provokes collisions with its neighbors' frames except the frames of its ongoing flow's next hop, by simply scheduling a transmission of a small or empty packet whenever it receives an RTS which is not destined to it and not sent by its next hop node. This process is illustrated by the Algorithm 6.

These two steps needs to be adjusted according to the misbehaving threshold, B_g, which means that the greedy node must compute the bandwidth share acquired by its next hop and adjusts its jamming rate and contention window accordingly. This process is described in Algorithm 7. In this algorithm, the estimation of the next hop's bandwidth is periodically computed whenever the timer period is expired.

6.1.5 Energy constraints

The mobile nodes in ad hoc networks usually need to be autonomous and independent from any central fixed infrastructure, and thus powered by batteries providing limited energy supply. In order to establish routes towards far away destinations, each node have to participate in a distributed routing protocol by exchanging broadcast/unicast control packets, leading it to spend more energy. Since our proposed greedy strategy is based on a proactive routing protocol, known by its heavy control traffic, so an important part of the energy is consumed in sending and receiving this traffic. Therefore, for energy awareness perspectives, the greedy node needs to minimize the energy wasted in jamming the frames sent in its neighborhood, otherwise its energy depletes rapidly. In order to minimize the consumed energy, a periodic tuning of the jamming rate is applied as described in the Algorithm 7.

[2] We use the mean of throughput of the greedy node's neighbors as there is a common bandwidth fair-share for each of them.

Algorithm 6 Greedy node behavior

if *(RTS received)* **then**
  $attempt$ = $false$;
  if *(@Dest == my address)* **then**
    schedule CTS transmission;
  else
    if *(@source $\notin$ NH-set)* **then**
      if *($++ CPT < n_1$)* **then**
        schedule transmission of empty or small packet after SIFS;
      end
      CPT = $CPT \ mod \ n_2$;
    else
      $attempt$ = $true$;
    end
  end
end
/*where NH-set is the set of the greedy node's next hops for all the flows. n_1, n_2 and CPT are values used to adjust the jamming rate such that $n_1 < n_2$. */

Algorithm 7 Next hop bandwidth estimation and adjustment of the cheating parameters accordingly

if *($DATA$ received) && ($attempt$ == $true$))* **then**
  B_{NH} = B_{NH} + P_l;
  if *(Period elapsed)* **then**
    if *($B_{NH} < E_{th}$)* **then**
      increase jam rate;
      if *($B_{own} > R_{share}$ +B_g)* **then**
        decrease k;
      end
    else
      if *((E_{th}-B_{NH}) > $threshold$)* **then**
        decrease jam rate;
      end
    end
    B_{NH} = 0;
    B_{own} = 0;
  end
end
/*B_{NH} and B_{own} are the bandwidth gained by the next hop and the greedy nodes, respectively. They are expressed in terms of number of time slots, during each period. The value $Threshold$ is used by the greedy node to prevent wasting more energy in jamming whenever its goal is achieved. k is the misbehavior coefficient used to choose a small backoff value as described in Chapter 2. */

This algorithm shows that the greedy node jams its neighbors' CTS frames at a minimum rate in order to allow its next hop of the ongoing flow to achieve the appropriate throughput.

According to the study done in [94], the energy consumed by the network interface for sending, receiving or discarding a packet is expressed as a linear equation:

$$cost = m \times size + b \tag{6.17}$$

such that the linear coefficients m and b represent an incremental cost proportional to the packet size and the cost of medium acquisition, respectively. In our strategy, the cost of jamming one CTS packet is given as follows:

$$\begin{aligned} cost_{jam} &= m_{jam} \times size_{jam} + b \\ &+ \sum_{|S_1|}(cost_{recv} + cost_{disc}) \end{aligned} \tag{6.18}$$

where $cost_{recv}$ and $cost_{disc}$ reflect the cost of the reception of the small packet sent by the greedy node and its destruction by its neighbors, respectively. This emphasizes the importance of the proposed strategy in terms of minimizing the jamming rate (i.e. jams only if necessary) in order to provide a QoS guarantee for the running application.

To evaluate the amount of energy wasted for running our scheme, let's assume that during one second of network lifetime X RTS frames have been successfully sent and the greedy node has provoked collisions with $X \times jam\_rate$ CTS frames, where
$0 \leq jam\_rate < 1$. Therefore, the overall energy $E_{overall}$ consumed by the greedy node for jamming others' frames during the network lifetime T can be expressed as

$$E_{overall} = T \times X \times jam\_rate \times cost_{jam1} \tag{6.19}$$

where

$$cost_{jam1} = cost_{jam} - \sum_{|S_1|}(cost_{recv} + cost_{disc})$$

6.1.6 Experimental study

We now proceed to the experimental evaluation of our proposed greedy strategy. First, we illustrate the propagation of the greedy behavior's impact in ad hoc networks. Then, we emphasize the benefits gained by the greedy node in terms of throughput, end-to-end delay, and delivery ratio of its traffic flow when it behaves according to our strategy. The simulation parameters are summarized in Table 6.5.

6.1.6.1 Propagation of greedy behavior impact

In our experiments, we consider the same topology shown in Figure 6.4 where two traffic flows are generated, G → A and F → H. The traffic sources send 1000 bytes every 2 ms (500 packets/s each), which means that each source node has a packet ready for transmission at each time slot. Figure 6.8 plots the obtained throughput by the greedy node G, its next hop node B and the node F with different values of the contention window of node G. When node G behaves correctly or sets its contention window constantly to 31 (equivalent to the minimum contention window CW_m), the node F gets more bandwidth since it has less contention than node G. As we can see from the network topology, the location of node F favors it to seize the channel more likely than nodes G and B, leading to short term unfairness as well as long term unfairness. For example, during node B's transmission, the node F monopolizes the channel by transmitting continuously over the link 8 (all links in conflict with this link are inactive) and then increases its chance to transmit before node G which is deferring due to node B's transmission.

The throughput earned by the node F decreases slightly with the decrease of node G's contention window, whereas the throughput of nodes G and B is increasing, until it collapses sharply when node G's contention window is set to 1. When node G monopolizes the medium by choosing constantly a backoff value equal to 0,

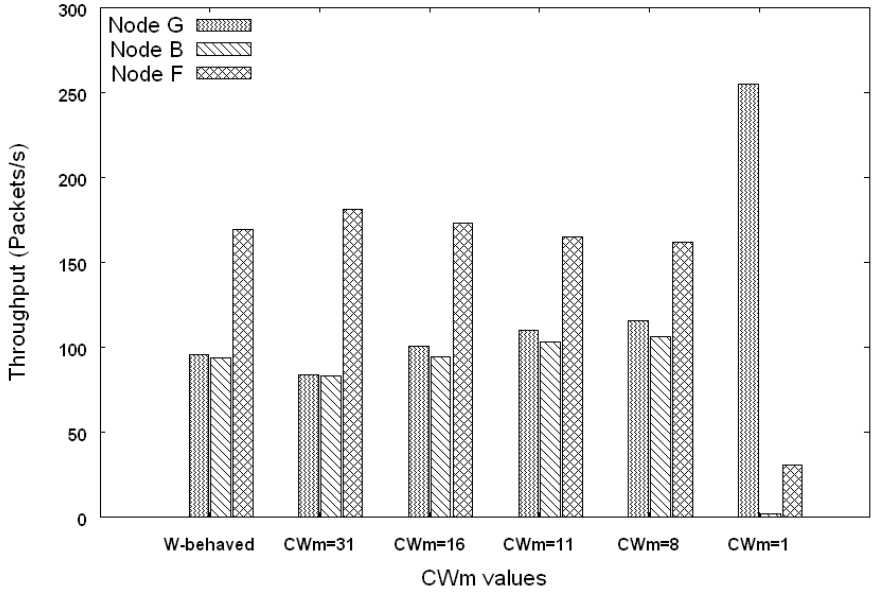


FIGURE 6.8: Propagation of greedy behavior's impact according to CW_m variation in MANETs, measured in terms of the acquired throughput.

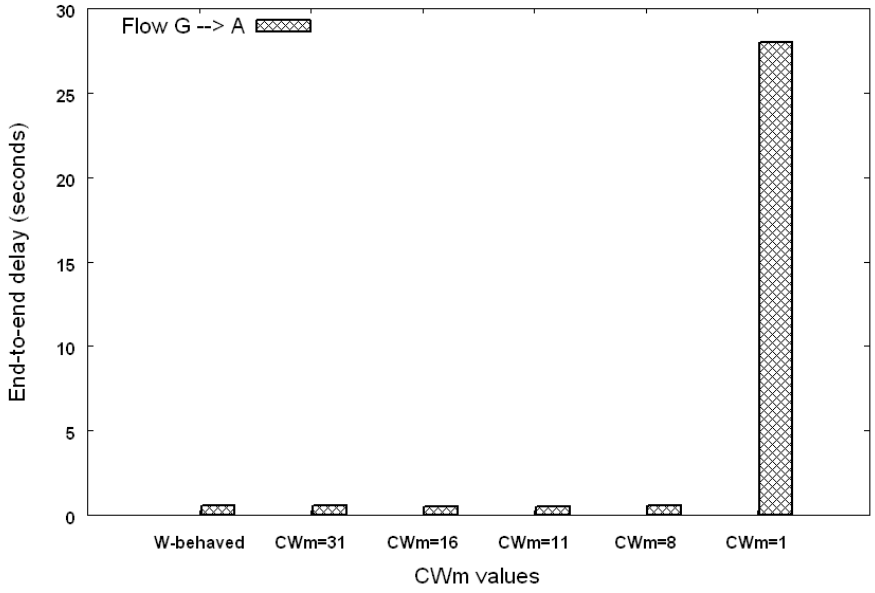


FIGURE 6.9: End-to-end delay of the greedy node's flow versus CW_m size.

Parameters	Values
Area	2000m × 1000m
Physical layer	direct sequence
Transmission range	250m
Carrier sensing range	550 m
Traffic type	CBR
Data rate	2 mbps
CBR packets size	500 bytes
Buffer size	64 packets
Simulation time	300 seconds
No. of simulation epochs	5
Network simulator	OPNET 14.0 [59]

TABLE 6.1: Simulation settings

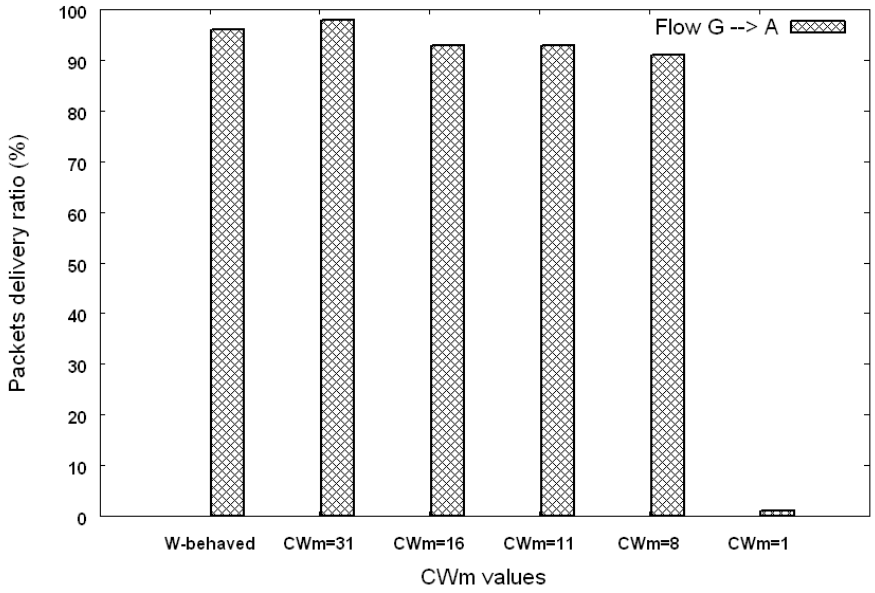


FIGURE 6.10: Variation of the packet delivery ratio of the greedy node's flow versus the chosen CW_m value

CW_m = 1, the throughput of its next hop, node B, drops sharply and consequently the delivery ratio of the flow G → A drops as well (see Figure 6.10.)

From Figures 6.8, 6.9 and 6.10, we note that the misbehavior of the node G has a devastating impact only when it constantly sets its CW_m to 1, where the end-to-end delay for the small portion of packets forwarded by node B becomes quite long leading to violation of the running application's QoS requirements. Moreover, this impact propagates to affect any other traffic flow within node G's carrier sensing range, which makes this area a bottleneck in the network.

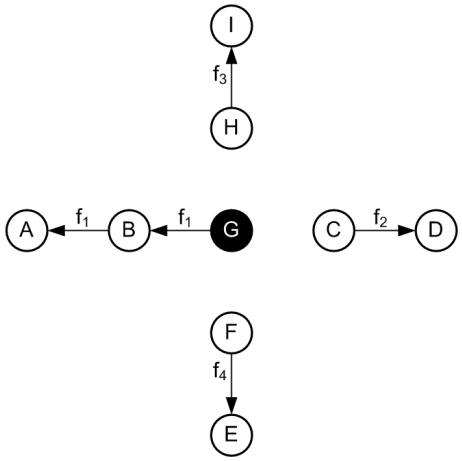


FIGURE 6.11: Topology used for evaluation of our proposed greedy behavior strategy.

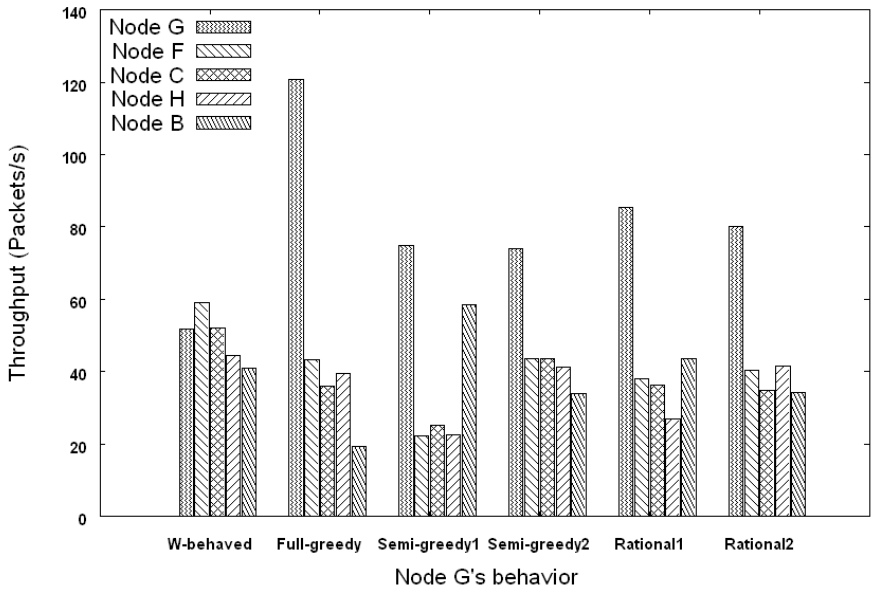


FIGURE 6.12: Variation of the traffic flows sources' throughput with the different cheating strategies adopted by the greedy node G.

6.1.6.2 Advantages of the proposed greedy behavior strategy

In this section, we highlight the advantage of adopting our strategy by the greedy node. We consider the topology shown in Figure 6.11, where four traffic flows f_1, f_2, f_3 and f_4 are generated in the network such that each source node sends 200 packets per second of 500 bytes each. In this scenario, we vary the node G's behavior among different strategies and observe the impact of each on the throughput, end-to-end delay and the packet delivery ratio. According to the results shown in Figure 6.12, we can see that when the node G tries to monopolize the medium for its own traffic (Full-greedy), its throughput gain is more than twice of the one earned in the W-behaved case and the bandwidth gained by its next hop node B is decreased to less than half. Consequently, the end-to-end delay of the flow f_1 is doubled along with the collapse of the packet delivery ratio as depicted in Table 6.2. Hence, as opposed to WLAN the Full-greedy strategy is inadequate in MANETs since it affects the performance of the traffic flow initiated by the greedy node itself.

		Full-greedy	Semi-greedy1	Semi-greedy2		
	W-behaved	CW_m = 1	CW_m = 31, jam_rate = 0.5	CW_m = 16, jam_rate = 0.2	Rational1	Rational2
End-to-end delay (seconds)	0.679	1.4295	0.554	1.1388	0.859	0.985
Delivery ratio (%)	79.11	15.95	76.59	46.04	50.59	45.93

TABLE 6.2: End-to-end delay and packet delivery ratio of flow f_1 under various greedy behavior strategies.

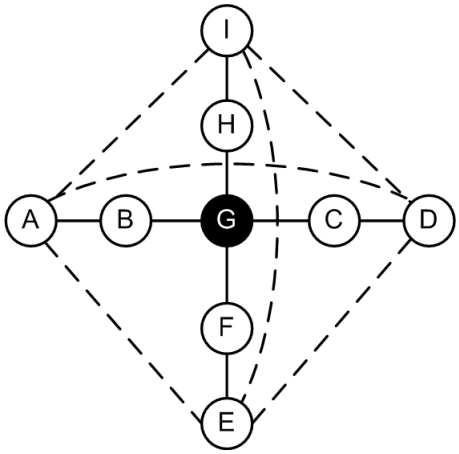


FIGURE 6.13: The topology perceived by the node G in the worst case, where the dashed lines denotes the extra links which are not acquired from Hello and TC message

As alternative strategies, we have implemented Semi-greedy1 and Semi-greedy2 in which the node G constantly sets its CW_m to 31 and 16, its jam-rate to 0.5 and 0.2, respectively. The results show that in the former strategy node G successfully increases its own throughput and the one of its next hop compared to the W-behaved strategy whereas the throughput of all its neighbors decreases to less than half. The drawback of this strategy is the energy necessary to jam 50 % of CTS packets sent by its 2-hops neighbors. Hence, due to the limited energy supply in MANETs, this strategy is unsuitable for adoption by the node G. For the latter strategy, node B's throughput is increased considerably along with the delivery ratio compared to the Full-greedy strategy; however, the rapidly changing topology of MANETs makes it inefficient since the chosen jam-rate and CW_m may not produce the same results in different network topologies.

Based on our discussion above, the main issues for choosing a suitable greedy strategy in MANETs are the energy constraints and the rapidly changing topology. To circumvent the limitations of the previous strategies regarding these issues, we apply our proposed method where we have implemented two scenarios Rational1 and Rational2. In Rational1, the greedy node G constructs the conflict graph according to the information acquired from HELLO and TC messages (i.e. the best case in terms of the obtained throughput), whereas in Rational2 it assumes the maximum number of contention among the links (i.e. the worst case for the estimated throughput). As shown in Figure 6.12 and Table 6.2, both scenarios give good results in terms of end-to-end delay and packet delivery ratio as compared to Full-greedy strategy, with a considerable increase of throughput where node B's throughput is almost equal to the one acquired in W-behaved strategy. Moreover, in both scenarios the greedy node G still gains more bandwidth than its neighbors and maintains a reasonable performance of its flow f_1. Therefore, these results prove the efficiency of our greedy strategy in MANETs.

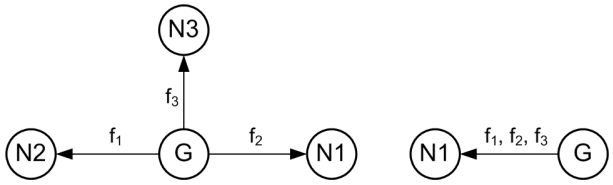


FIGURE 6.14: Multiple traffic flows issued from the greedy node G and forwarded either through one or several next hops

Scenario	Network density (D_N)	#flows
1	5.0×10^{-6}	5
2	1.5×10^{-5}	10
3	2.5×10^{-5}	15
4	3.5×10^{-5}	15
5	5.0×10^{-5}	15

TABLE 6.3: Scenarios setting

6.1.6.3 Impact of the mobility and network density on the efficiency of our greedy strategy

In order to assess the efficiency of our greedy strategy in dense and highly mobile network, we generate 5 random network topologies consisting of 10, 30, 50, 70 and 100 nodes, respectively. A number of traffic flows are also generated in the network such that each source node sends 100 packets per second of 500 bytes each. Some of these flows are generated by the greedy node and forwarded either through one or multiple neighbors.

In these scenarios the nodes move randomly within the area and their velocities vary from 0 m/s to 20 m/s. The network density (D_N) is estimated according to the following formula

$$D_N = \frac{N}{area} \tag{6.20}$$

where area is the network area in m^2 and N is the number of nodes in the network.

The different values of the network density and the number of traffic flows generated in each scenario are summarized in Table 6.3.

We define the effectiveness factor (Γ) in order to measure the efficiency of our greedy strategy with the variation of the following metrics: nodes speed, network density and the number of flows. This factor can be expressed as follows;

$$\Gamma = \frac{th_{ng} + \sum_{(i=1)}^{|NH-set|} th_{ni}}{|NH-set|+1} - \frac{\sum_{(i=1)}^{N1-|NH-set|} th_{ni}}{N1-|NH-set|} \tag{6.21}$$

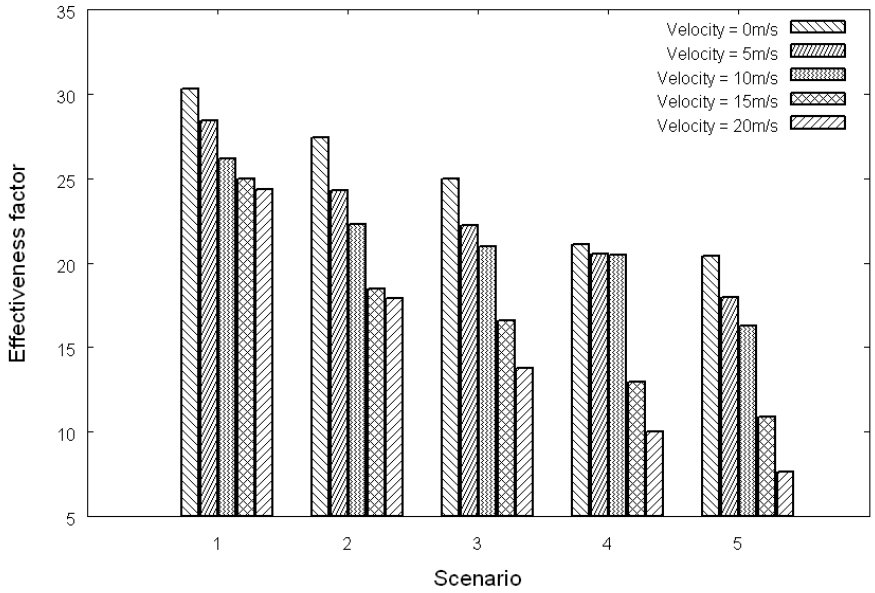


FIGURE 6.15: Variation of the effectiveness factor in different scenarios: case of greedy node sending multiple flows through only one next hop node

such that th_{ng} and th_{ni} are the throughput of the greedy node and its next hop nodes, respectively. $N1$ refers to the number of nodes within the sensing range of the greedy node which are generating traffic flows or forwarding data packets and $|NH - set|$ is the number of next hops of the greedy node.

As we can see from the curves plotted in Figure 6.15, the effectiveness factor reduces with the increase of nodes mobility and network density as well as the number of competing flows in the network. Moreover, it is observed that Γ decreases more in scenarios 4 and 5 where the nodes' speed are 15 m/s and 20 m/s respectively. Despite that our scheme is still effective since the worst value of Γ is 8 packets/s.

As expected, our greedy strategy fails if the greedy node is sending multiple flows simultaneously through several next hops nodes as depicted in Figure 6.14, hence even if the greedy node jams the other neighbors' frames its next hops' nodes have to compete with each others to gain access to the medium. Therefore, Γ reduces as the number of flows initiated or forwarded by the greedy node through different next hops's nodes goes to higher. The results graphed in Figure 6.16 show that Γ reduces sharply as compared to the results plotted in Figure 6.15, in which all the flows initiated from the greedy node are forwarded through one node, because the competition between the next hops nodes leads to a large number of collisions which makes the task of applying our scheme very difficult. Moreover, the increasing velocities of nodes and network density participate also to the failure of our scheme, especially in the scenario 5 where the value of Γ is equal to 0 when the velocity of nodes is 20m/s.

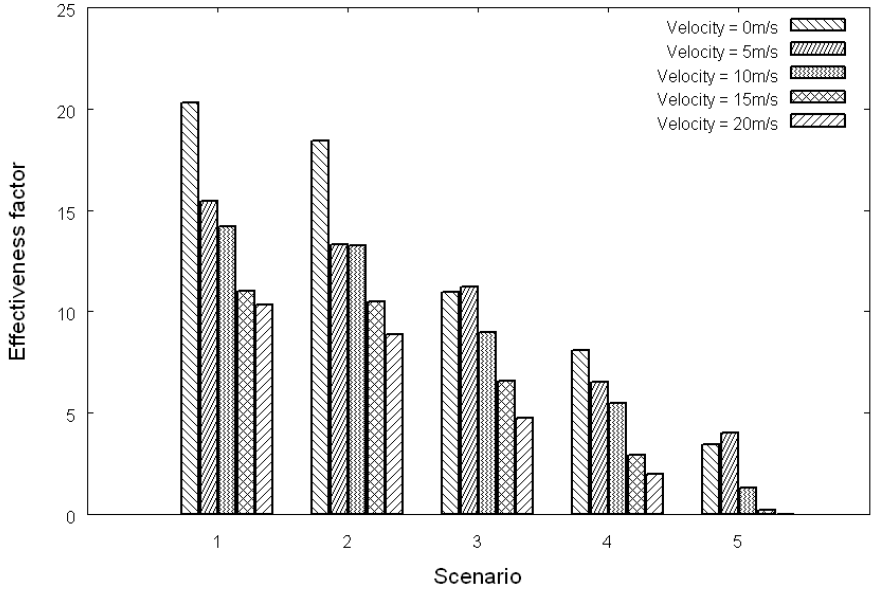


FIGURE 6.16: Variation of the effectiveness factor in different scenarios: case of greedy node sending multiple flows through several next hops

6.2 Thwarting backoff rules violation in MANETs

6.2.1 Introduction

Many schemes have been proposed in the literature to cope with the greedy behavior at MAC layer and particularly Backoff rules violation misbehavior. One of the common weaknesses of these schemes, discussed in the previous chapter, is that they delegate the monitoring of the sender node's behavior to the intended receiver (for example, the access point in WLAN scenario), thus any collusive misbehavior of these two nodes leads to a severe damage to the MAC protocol operations and makes them favorite to frequently access the medium. To circumvent this drawback, we propose to anticipate the defense technique and applying it at earlier stage in order to disable the ability of any node to either disobey the protocol rules or collude with its one hop receiver to evade detection. To this end, we suggest the modification of the standard backoff algorithm by using a one way function to generate the backoff values, with respect to the contention window. The main advantage of this scheme is that it preserves the randomness property of the standard backoff algorithm and makes all one hop neighbors of the sender aware of its backoff value. These neighbors are able to compute the backoff chosen by the sender node since the input parameters of the one way function are piggybacked with the RTS frame. This input is the combination of the number of failed transmissions and the CRC field's value of the data packet to be transmitted. This scheme prevents any node from fabricating its backoff value due to the specific properties of the one way functions. Likewise, the receiver node cannot collude with the sender by abstaining from launching a reaction towards it if this latter didn't follow our scheme. This is due to the fact that all their common neighbors can monitor the behavior of the sender and discover that it is disobeying the protocol rules.

6.2.2 The proposed scheme

Selfish or greedy behavior of nodes at MAC layer remains a hard problem to solve despite the numerous works done in the literature. To resolve this problem, we have devised a new scheme that allows the neighbors of any node to detect its attempt to deliberately disobey the MAC protocol rules, by either fabricating a small backoff

value or refusing to increase its contention window after an unsuccessful transmission. As opposed to the existing solutions, our scheme needs only few samples of observations as compared to detection windows based schemes like DOMINO and may trigger a negligible number of false alarms as compared to per frame monitoring based schemes. By proposing this scheme we aim to achieve the following objectives:

- Eliminating the assumption of trustworthiness of one part of the communication.
- Prevent any node from fabricating a small backoff value.
- Detect the nodes that refuse to double their CW after collision.
- Ensuring fast detection with less number of false alarms.
- Detect sender-receiver collusion.

Notice that in this scheme we consider that an anti-MAC spoofing mechanism is deployed at MAC layer.

6.2.2.1 Our Backoff computation scheme

In every RTS frame we add a new field dubbed $Attempt$ (Γ) which represents the number of times the sender has tried to transmit the RTS frame and its corresponding DATA packet. The $Attempt$ value is initialized to 1 after a successful transmission and incremented by 1 for each unsuccessful transmission of RTS or Data packet. The backoff value is computed according to the following formula:

$$bf = Hash(f(CRC, \Gamma)) mod\, 2^{\Gamma-1} CW_{min} \tag{6.22}$$

where

$$f(CRC, \Gamma) = (CRC \bigoplus \Gamma) \tag{6.23}$$

such that CRC is the cyclic redundancy code calculated over all the fields of the data packet to be transmitted and bf is the backoff value. After an unsuccessful transmission the sender node increases its transmission attempt Γ by 1 and doubles its contention window.

By adopting this scheme, the receiver can detect any deviation from the sender since it is able to recalculate the backoff value that has been chosen by this sender. This is feasible because the CRC and Attempt values, used by the sender to compute its backoff value, are known to the receiver.

The purpose of using hash function to generate the backoff value is to prevent any node from fabricating its backoff value since it is generated by one way function. Moreover, the use of the CRC field as input of the function f may reduce significantly the number of collisions since its value is different for each data packet. Finally, we have added Γ as a second input of f in order to deal with the retransmissions; as in this case the CRC is unchanged, then the new backoff value should be different from the previous one because Γ increases with every unsuccessful transmission. Hence, the randomness property of the backoff is ensured by the hash function whereas the likelihood of repetitive collisions is kept minimal due to the chosen inputs of the function f.

6.2.2.2 Detection of protocol rules violation

On receiving an RTS frame, the receiver node extracts the CRC and Γ values from the frame and uses them together to calculate the backoff value used by the sender. If the number of idle slots [3] observed by the receiver

[3] We consider only the number of the elapsed idle slots during which no collision is observed.

MAC address	#Attempts	CRC	Backoff check	Status
0A.34.9C.12.E2.48	3	0x1EDC6F41	No	Legitimate
DA.E4.9C.B2.92.4A	2	0x82F63B78	Compulsory	Suspicious
EA.10.90.BE.AD.B8	5	0x8F6E37A0	Compulsory	Suspicious
.....	...	...	...	...

TABLE 6.4: The monitoring table

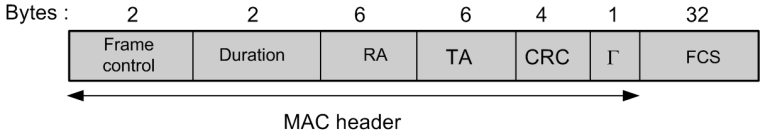


FIGURE 6.17: The new format of RTS frame

node since the last transmission of the sender is less than ((backoff + DIFS)- ϵ) then this indicates that the sender node has violated the protocol rules. Consequently, it will be punished. Notice that ϵ, called accuracy factor, is used to minimize the number of false accusations. This false accusation occurs when the receiver node estimates that the sender is counting down its backoff timer whereas this timer is frozen due to the sender being within an interference area of a third node.

Case of consecutive collisions In this case, the sender node may misbehave by keeping the Γ value unchanged and consequently doesn't double its contention window. To counter this misbehavior, we suggest piggybacking the CRC value of the DATA packet to be transmitted in the RTS frame. After several retransmissions due to the collision of either CTS or Data packets, if CRC values are similar and the Γ value remains unchanged then this indicates that the sender misuses the protocol rules (i.e. it is cheater).

Some works in the literature have proposed to deny response to one RTS frame and check whether the sender increases its Γ value or not in the subsequent frame. This scheme can lead to wrong accusation in the case of dense network with high collision rate where the receiver falsely deems the sender as cheater. Figure 6.18 depicts a scenario of false accusation that can be explained as follows; the receiver node successfully receives the first RTS frame and records the $Attempt\#$ of the packet1 but it refuses to send a CTS in order to check whether the sender is incrementing its $Attempt\#$ correctly or not. The sender tries again to retransmit the RTS frame however due to the high collision rate no frame is delivered to the receiver. Consequently, when the $Atttempt\#$ reaches the short retry limit (SRL) no further retransmission is allowed and the corresponding data packet is dropped. Afterwards the sender schedules the transmission of the subsequent data packet by sending a new RTS frame with $Attempt\#$ equal to 1. If it is successfully received by the receiver then this latter infers that the sender is a cheater node because it didn't increase its $Attempt\#$ although it has experienced a collision.

As a punishment scheme, the receiver node may deny response to the frame sent by the misbehaving node or for a less severe chastisement it delays the delivery of its packets. Therefore, the applications running at the sender side will experience severe performance degradation. Furthermore, a more efficient reaction can be launched at routing layer by refraining from relaying any control message sent by the detected cheater, such as RREQ (Route Request) message in AODV, TC (Topology Control) message in OLSR, etc. Likewise, the receiver node may also delete the identity of the cheater node from its MPR (MultiPoint Relay) selectors set when it broadcasts its TC

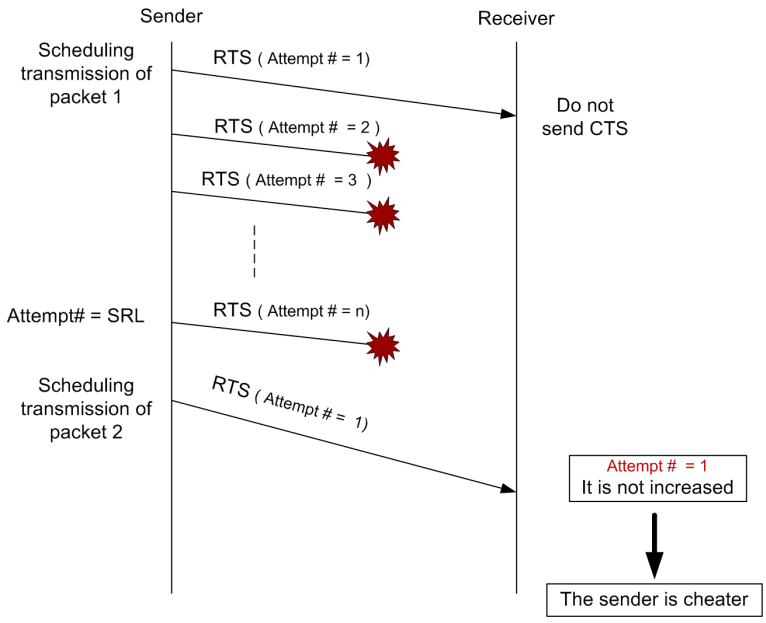


FIGURE 6.18: Scenario describing a case of false accusation of a legitimate node

message, if the cheater has chosen it as MPR. As a consequence of these punishment schemes the cheater node will be absolutely isolated from the network if all its neighbors detect its misbehavior.

6.2.2.3 Cheater identity dissemination (Reaction)

As opposed to the existing schemes, our solution gives a chance to the cheating node to repent and abide again to the protocol rules, and thereby it evades from the punishment. Notice that we have opted for this mechanism because we believe that it is better to incite a node to behave correctly rather than excluding it from the network. This belief is motivated by the fact that in MANETs every node is contributing significantly to make available all the network's services such as connectivity, routing and internet access. For example, if the cheater node is the only node in the network providing connection to some nodes or it is a gateway to the internet, then its exclusion from the network may lead to network partition and unavailability of some services like internet connection. Moreover, even if several nodes are providing the same service that the cheater node ensures, its punishment leads to overloading the other nodes if no load balancing mechanism is used. Furthermore, If the cheater node is the unique intermediate node to reach the leader of a group of soldiers in battlefield then its exclusion leads to devastating impact. Thus, it is more judicious to try to maintain any node in the network rather than its elimination.

One of the main issues of misbehaving nodes advertisement in MANETs is the prevention of false reports that may lead to harmful consequences as explained above. These false reports are usually issued from adversary nodes claiming that a given legitimate sender is not abiding the protocol rules in order to exclude it from the network. In what follows, we define a novel scheme to advertise the identity of the misbehaving nodes. Once a deviation of a sender node is observed by its corresponding receiver, this latter piggybacks the sender's identity and the observed deviation with its subsequent RTS frame, the resulting frame is dubbed RTS_{alarm}. On receiving this frame, each node sets the Backoff-check entry in its monitoring table, as illustrated in Table 6.4, to compulsory and the status entry to suspicious. Then, it monitors all the transmissions initiated by the advertised node in order to check its behavior and decide whether it is correctly applying the protocol rules, as described in the Algorithm 8. Notice that the advertisement of the sender's identity is considered as a warning message that aims to inform it that its direct receiver is aware of its misbehavior and thereby incites it to behave correctly. If it does so for

Parameters	Values
Area	2000m × 2000m
Physical layer	direct sequence
Transmission range	250m
Carrier sensing range	550 m
Topology	5..150 nodes
Mobility model	random waypoint
Node's speed	5 m/s
Traffic type	CBR
Data rate	2 mbps
CBR packets size	500 bytes
Hash function	MD5 (128 bits)
Simulation time	300 seconds
No. of simulation epochs	5
Network simulator	OPNET 14.0 [59]

TABLE 6.5: Simulation settings

its subsequent data packets then no punishment will be applied against it, otherwise all its neighbors that have received the advertisement frame will apply the punishment scheme described in section 6.2.2.2.

6.2.2.4 Detection of collusive nodes

Sender-receiver collusion leads to a severe performance degradation in the network. In this case, the neighbors of the colluding nodes experience a significant reduction of their acquired throughput. Once such throughput degradation is observed, each node starts supervising all the transmissions in its vicinity to distinguish which nodes are getting a higher throughput. Once this set of nodes is identified, the node supervises carefully every transmission initiated from them and checks whether they are respecting the backoff rules or not. If any node among them disobeys our proposed scheme and its intended receiver didn't inform its neighbors about this misbehavior, then the monitor node piggybacks both of their identities in its subsequent RTS frame, dubbed RTS_{alarm+}, along with the corresponding estimated deviation. The deviation of the sender is the difference between the estimated backoff and the observed one, whereas we associate a negative number (-1) to the receiver identity to precise that this latter is refusing to reveal the identity of a cheating node. Notice here that the monitor node should be neighbor of both of the sender and receiver. The receivers of this alarm frame behave similarly to its sender and monitor both of the advertised nodes until either their misbehavior is confirmed or they are repented and abide again to the MAC protocol rules, as illustrated in the Algorithm 9 .

6.2.3 Experimental study

We now proceed to the evaluation of the performance of our scheme. The configuration parameters of each node are summarized in Table 6.5. In order to highlight the strength of our scheme as compared to the standard BEB algorithm we have chosen to measure the following metrics.

Algorithm 8 Upon reception of RTS_{alarm}

```
Initialization:
NB-RTS_Alarm = 0;
confirm = 0;
if (status(id) ≠ Cheater) then
    NB-RTS_Alarm ++;
    if (NB-RTS_Alarm == 1) then
        Backoff-check(id) = Compulsory;
        status(id) = suspicious;
    end
    if (own-est-dev(id) ≪ max-dev) then
        Ignore RTS_Alarm;
    else
        if (max-dev - rec-dev ≤ thresh) then
            if (sender-id ∉ confirm-list) then
                confirm++;
            end
        end
        if (confirm == X) then
            status(id) = cheater;
        end
    end
end
```

own-est-dev: the deviation of the node id as observed by the receiver node.
max-dev: the maximum authorized deviation from the expected backoff value (i.e. due to channel conditions and hidden terminals phenomenon).
rec-dev: the deviation advertised in the RTS_{alarm} frame.
thresh: threshold indicating whether the received deviation can be considered as an attempt to misuse the protocol rules although it didn't overtake the max-dev.
X: this number is determined according to the accuracy level that the node wants to achieve. A larger value of X indicates that the node wants to reduce significantly the number of false alarms. Notice that this value is limited by the number of common neighbors of the supervisor and supervised node.
id, sender-id : are the identity of the advertised suspicious node and that of the sender of RTS_alarm, respectively.
confirm-list: the list of the neighbors that have already announced a significant deviation of the advertised node.

Algorithm 9 Upon reception of RTS_{alarm+}

```
extracts both of nodes' identities (id1, id2) and their corresponding deviations;
for the deviating sender node (id1) apply Algorithm 8;
set backoff-check(id2) to compulsory;
set status(id2) to suspicious;
monitors all the transmissions of both nodes id1 and id2;
if (status (id1) == cheater) then
    if (No RTS_Alarm has been received from id2) then
        status(id2) = cheater;
    end
end
```

- Normalized throughput: the normalized throughput of a node id is defined as:

$$N_{throughput} = \frac{Throughput_{id}}{Offered\ \ bandwidth} \tag{6.24}$$

this metric is computed for all scenarios under both IEEE 802.11 DCF protocol and our scheme. It is a good indicator to compare the performance of these two protocols.

- Fairness index (fi): The Jain's fairness index introduced in [95] is an important property that allows us to verify whether the proposed scheme still ensuring a fair share of bandwidth among the contending nodes. The fi is defined as follows:

$$fi(Th_1, Th_2, ..., Th_N) = \frac{(\sum_i Th_i)^2}{N \times \sum_i Th_i^2} \tag{6.25}$$

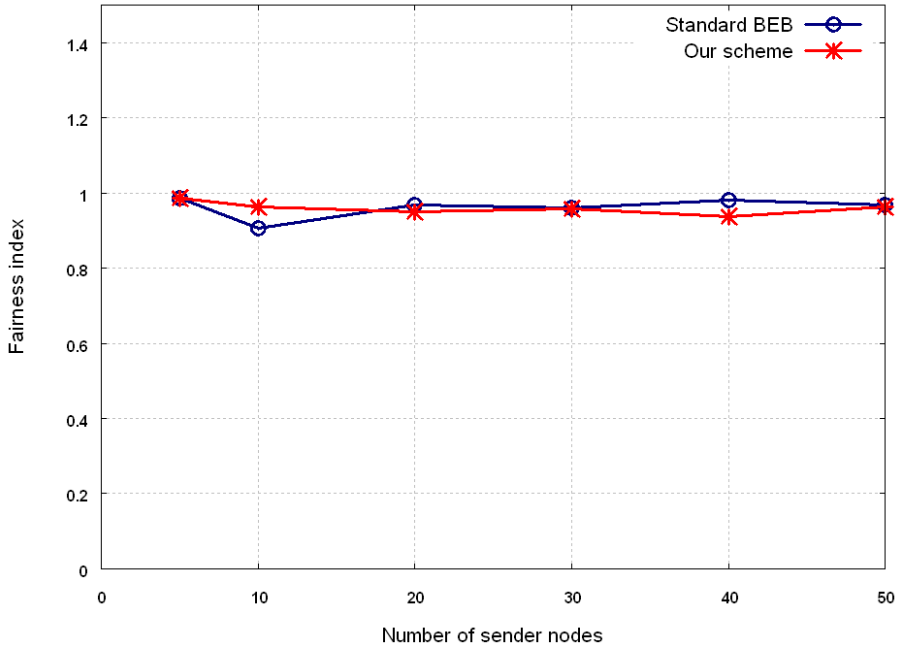
such that Th_i refers to the throughput of the traffic flow i and N is the number of the contending flows. Notice that if only M of N flows receive equal bandwidth (and others get none), then the fi is $\frac{M}{N}$. Therefore, a value of this index close to 1 indicates that our scheme is providing better fairness.

Figure 6.19(a) plots the fi values measured on a random topology with varying network size. Both our scheme and the standard BEB show tightly closed values which are always closed to 1. This proves that using a hash function to generate the Backoff value ensures fair share of bandwidth among the contending nodes. The normalized throughput of each sender node is graphed in Figure 6.19(b), where it shows that the achieved normalized throughput is much higher in case of small number of senders (dense network of 5 senders) and decreases to the half when the number of senders is doubled. This is due to the fact that in this latter case each node acquires half of the bandwidth gained in the previous scenario. Since the network topology is randomly deployed then it is composed of separated dense sets of nodes connected among them, therefore the throughput acquired in each set is independent from that of the other sets. Hence, when the number of senders overtakes 10 nodes the obtained normalized throughput for each node reduces slightly as compared to the values acquired in the scenario of 10 senders. Notice that the standard BEB outperforms our scheme in some scenarios however the gap is usually very small.

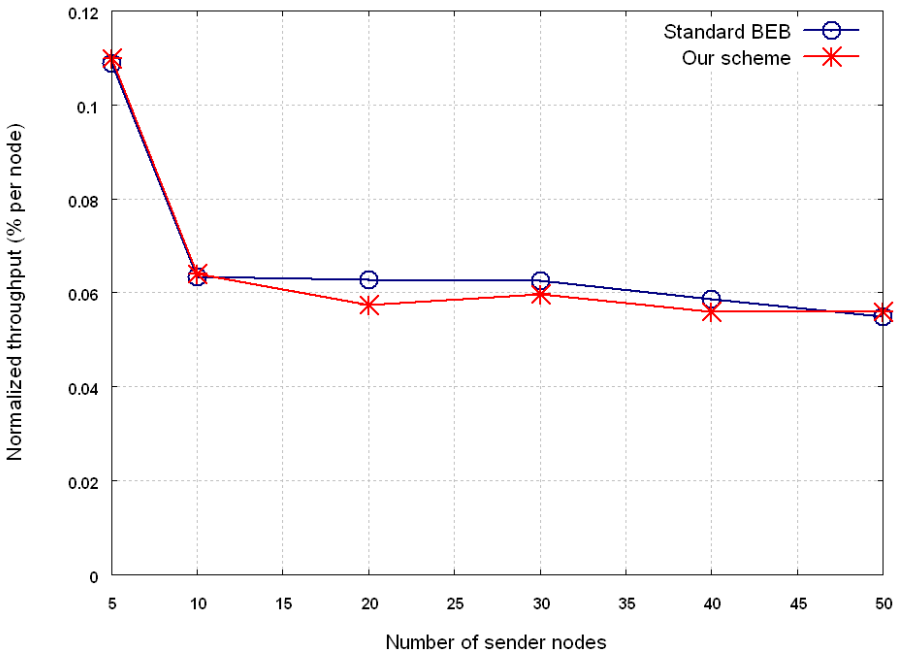
Figure 6.20(a) reveals that the fi of our scheme slightly decreases as the offered bandwidth increases and the same behavior is observed for the standard BEB. This decrease is due to the fact that when the offered bandwidth is low (1 and 2 mb/s or even 5.5 mb/s) the difference between nodes' throughput is very small or negligible, whereas when the offered bandwidth gets higher, such as 11mb/s or greater values like in IEEE 802.11 g, the gap between the gained bandwidth of the nodes augments and consequently fi experiences a slight decrease. Additionally, we remark that both of the protocols keep always tightly closed values and in some scenarios (in case of 2mb/s and 5.5 mb/s) our scheme slightly outperforms the standard BEB.

The curve plotted in Figure 6.20(b) highlights the impact of the percentage of the cheating nodes (PC) on the fairness property of the standard BEB. According to this curve, we remark that fi experiences a sharp decrease when the percentage of the cheater nodes is low, PC $\leq 20\%$; however it starts to rise once the PC gets higher (PC $> 20\%$) till it reaches the highest value when the percentages of cheater and honest nodes are identical. The variation of fi values can be explained as follows; initially the cheater nodes seize higher throughput at the expense of the honest nodes leading to reduction of fi. Afterward, when more cheaters join the network the extra throughput previously earned by the existing cheaters is reduced due to the high collision rate. As a result, fi gets higher. Notice that the high collision rate is caused by the small CWs chosen by the cheater nodes.

Now we assess the efficiency of our scheme in terms of detection rate and accuracy. The detection accuracy is calculated based on the percentage of the triggered false alarms in the network. To calculate this metric, we

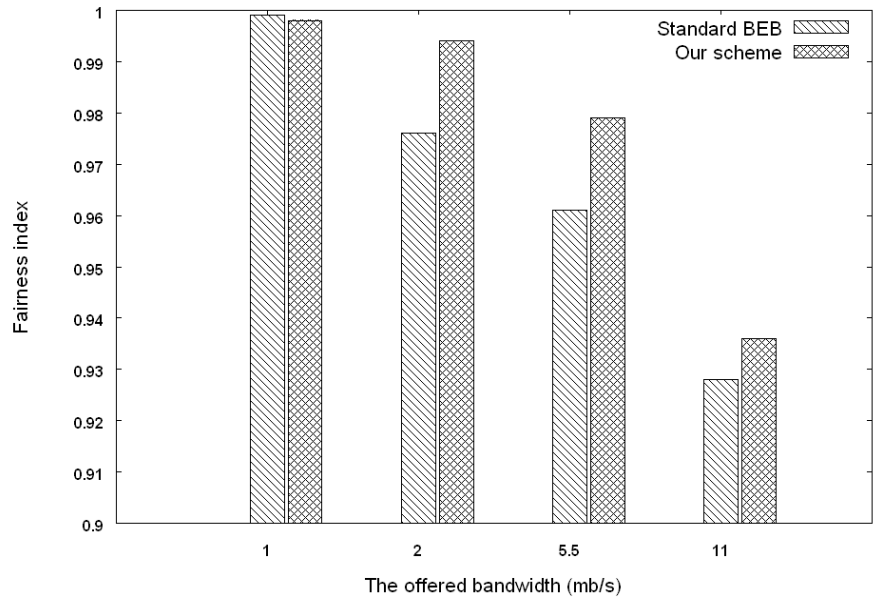


(a) Comparison between standard BEB and our scheme: Fairness-index versus the number of sender nodes in the network

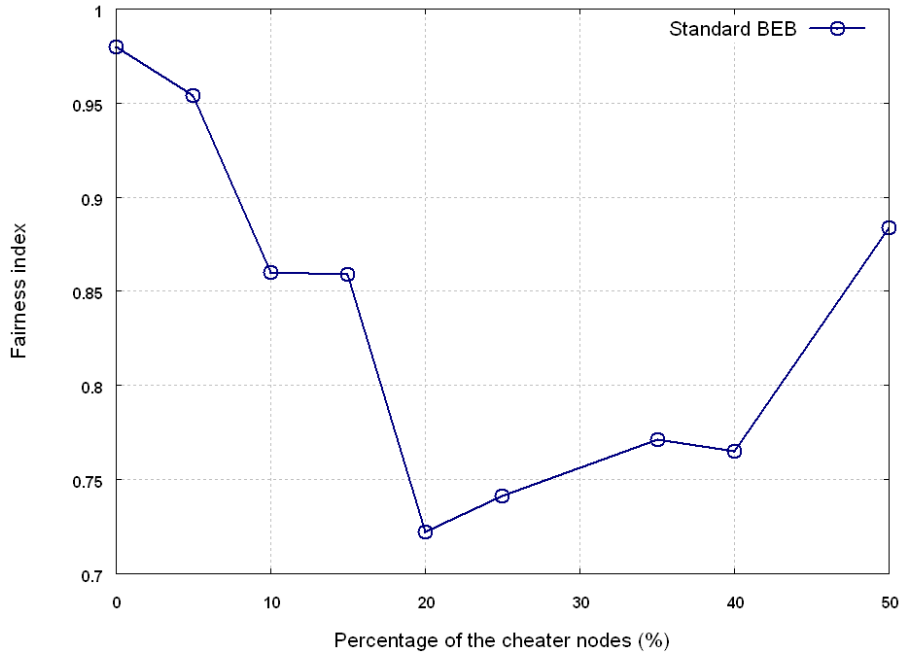


(b) Comparison between standard BEB and our scheme: Normalized throughput versus the number of sender nodes in the network

FIGURE 6.19: Impact of the number of senders on the fairness index and the normalized throughput: case of random topology



(a) Impact of the variation of the offered bandwidth in the network on the Fairness index



(b) Impact of the number of cheaters in the network on the Fairness index of the Standard BEB

FIGURE 6.20: Impact of the offered bandwidth and MC on the fairness index: case of ring topology of 21 nodes

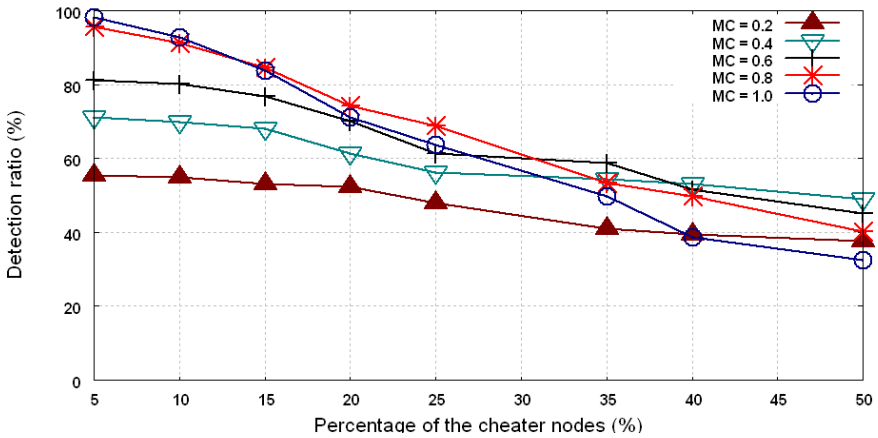


FIGURE 6.21: Detection ratio versus the percentage of the cheater nodes: case of random topology

consider a random topology with varying size and percentage of cheater nodes along with their misbehavior levels. Figure 6.21 shows the variation of the detection ratio of our scheme in function of the percentage of cheaters in the network and the applied misbehavior coefficient. As we can see from the plotted curves, the detection ratio reaches its highest values when the number of cheaters is low and the MC is high (the detection ratio is close to 100 % when the MC is equal to 0.8 and 1). Then it decreases as the PC increases, where we observe that the higher the MC is the more reduction of detection ratio. This decrease is justified by the rise of the collision rate among the cheating nodes and the interferences that prevent the receiver node from estimating the correct deviation.

The false alarms triggered by a detection scheme are an important metric that outlines the drawbacks or the harm induced from the deployment of this scheme in the network. This metric represents the percentage of the well-behaved nodes that are wrongly classified as cheaters by our scheme. As shown in Figure 6.22, no false alarms are triggered by our scheme in the case of network characterized by a ring topology for different values of MC. This is due to the absence of interferences that may disrupt the observations taken by the nodes to compute the backoff values of their neighbors. However, in the case of random topology the percentage of false alarms is quite high and increases as the size of the network rises. We remark that the extent of the misbehavior level (MC) doesn't affect the percentage of false alarms since this latter is caused by the interferences due to the hidden terminal phenomenon. Despite the high detection rate of our scheme it still suffering from the wrong accusation of some well-behaved nodes due to the triggered false alarms. Therefore, in order to overcome this drawback an adequate increase of the accuracy factor (discussed in section 6.2.2.2) remains a good choice to alleviate the impact of these false alarms.

6.3 Conclusion

Facing the greedy nodes at MAC layer still a hot topic that requires the use of many tools to prevent definitively this devastating problem. From fuzzy logic to game and Bayes theories, all these techniques aim to detect any deviating greedy node that doesn't obey the MAC protocol rules. However, the main question here, is there any efficient scheme that guarantees the prevention of such misbehavior? In fact, this is still an open question that we hope to can answer it in the near future.

Throughout the first part of this chapter, we have analyzed the greedy behavior problem in wireless ad hoc networks and proven that its impact can be more devastating compared to that in wireless local area networks. The propagation of the effect of this misbehavior is illustrated through conflict graphs analysis. As a result of

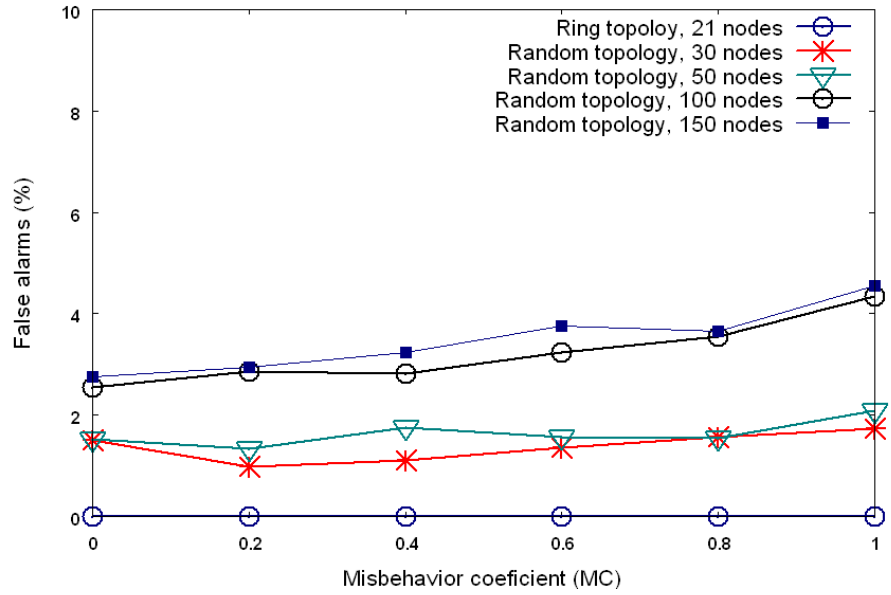


FIGURE 6.22: Impact of the MC, network size and topology on the percentage of the false alarms

this investigation, an effective greedy behavior strategy is proposed, which is suitable for ad hoc networks. This method allows the greedy node to gain more bandwidth share compared to its neighbors and keeps the performance of its ongoing flows reasonable by maintaining its extra bandwidth share within the misbehaving threshold. Our algorithm is evaluated through extensive simulations and the obtained results highlight its advantage in terms of the increase in delivery ratio and the reduction of the end-to-end delays compared to the Full-greedy strategy applied in WLAN.

Subsequently, we have presented a novel solution to prevent and detect MAC layer misbehavior resulting from violation of the backoff computation rules. The proposed scheme aims to circumvent any misuse of the backoff algorithm at MAC layer, and thereby ensuring a fair share of bandwidth among the contending nodes. To this end, the nodes are required to use a one way function to generate their backoff values and make the state of their transmission attempts known to their neighbors. Thus, any receiver node is made aware of the backoff value used by the sender to access the medium and consequently it is able to detect any deviation from this value. Furthermore, the proposed scheme is resistant to sender-receiver collusion and provides a novel reaction technique. The simulation results show that our scheme fulfills the fairness property and ensures high detection rate of the cheaters in various scenarios with minimum false alarms.

Chapter 7

Conclusion and Future Work

We conclude our major contributions in this chapter, and point out future work.

Wireless multi-hop networks, for example MANETs, VANETs and WMNs, have significantly facilitated our daily life. However, many challenging issues must be tackled before their widespread deployment. The foremost issue is to secure routing and MAC layers protocols as the performance of any application depends on the reliability of the services provided by these two layers. Moreover, the architecture of these networks requires explicit cooperation between the nodes to fulfill routing functionalities, and implicit cooperation at MAC layer by respecting the correct rules of IEEE 802.11, thus it is a key issue to ensure this cooperation by developing appropriate schemes.

In this dissertation, we dealt with two misbehaviors at both routing and MAC layers. Most of the existing research works which investigate the potential threats against routing protocols in MANETs, such as OLSR, and IEEE 802.11 MAC protocol, assume simple attack scenarios and design solutions accordingly. These solutions cannot deal with more sophisticated and collusive attacks. We, therefore, are motivated to carefully reexamine those protocols for defining new attack scenarios, like the colluding and cross layer black hole attack, and the adaptive greedy behavior, along with effective countermeasures.

We consider black hole attack in that it may lead to sharp network performance deterioration. Even worse, it may cause network partitioning if control packets, such as TC packets in OLSR, are compromised. On the other hand, our choice of IEEE 802.11 is justified by the increasing attention devoted to develop robust MAC layer protocols. Since all the recent versions of MAC protocols are extensions of the basic model of IEEE 802.11, thus we believe that identifying new misbehaviors that threaten the functioning of this model will significantly help to secure theses new extensions.

To secure OLSR against black hole attack, we have made a comprehensive investigation of its root causes, its consequences and highlighted the advantages and drawbacks of the existing countermeasures. Afterwards, we have identified a new collusive attack scenario and proposed three hop acknowledgement based scheme to cope with it. This scheme achieves good results with minimum number of false alarms. Additionally, another cross layer collusive attack is also identified. Then, a solution that requires slight modification to RTS and CTS frames format is designed to prevent such attack. This solution is efficient however it still unable to identify the attacker nodes.

To secure MAC layer protocol, we have proposed two schemes, namely FLSAC and its extension that uses bayesian model, to struggle against the adaptive greedy behavior in WMNs, and shown that they can be combined together to ensure better efficiency. The resulted scheme reduces significantly the impact of this misbehavior and

shows high accuracy. Moreover, a new characterization of greedy behavior in MANETs is introduced and its effectiveness is proven in different simulation scenarios. Finally, we have also proposed an alternative backoff scheme that preserves the randomness property of BEB while it allows fast detection of the greedy nodes.

We believe that the contributions presented in this dissertation constitute a significant input to the existing research works on routing and MAC layers misbehaviors in Wireless multi hop networks. We hope also that the findings of these contributions will be the starting point for Ph.D students pursuing research in the same research topic.

As part of our future work, we plan to pursue our research activity in the following directions:

Investigating Jamming Attack: for the greedy behavior addressed in chapters 5 and 6, we have considered a greedy strategy in which the node scrambles or jams the CTS/DATA/ACK frames of its neighbors; however we didn't explain, in details, how can it does that and what will be the consequences if it jams all the frames in its neighborhood. Moreover, to design a robust MAC layer protocol we must address the jamming attack in addition to the greedy behavior. To achieve our ultimate goal which consists in developing a framework that encompasses both of the above misbehaviors, we need to enhance our contributions to take into account the jamming attack with all its possible techniques. Hence, this is our main focus for the future works.

Testbed based evaluation of the proposed schemes: as most of our contributions have been evaluated through simulation using OPNET network simulator, we intend to implement them into real testbed and assess their performance in such real network environment.

Publications List

Journal papers

1. S. Djahel, F. Naït-Abdesselam and Z. Zhang. **Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges**. *IEEE Communications Surveys and Tutorials*, available online .

2. S. Djahel, F. Naït-Abdesselam and D. Turgut. **Characterizing the Greedy Behavior in Wireless Ad Hoc Networks**. *Journal of Security and Communications Networks (SCN), Wiley InterScience Publisher*, available online.

3. 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, Num. 10, pp. 1245-1266, june 2009.

4. S. Djahel, F. Naït-Abdesselam, Z. Zhang and A. Khokhar. **Defending Against Packet Dropping Attacks in Vehicular Ad Hoc Networks**. *Journal of Security and Communications Networks (SCN), Wiley InterScience Publisher*, Vol. 1, Num. 3, pp. 245-258, May-June 2008.

Conference papers

5. S. Djahel, Y. Begriche, and F. Naït-Abdesselam. **A Bayesian Statistical Model to Alleviate Greediness in Wireless Mesh Networks**. *IEEE Global Communications Conference (IEEE GLOBECOM)*, Miami, Florida, USA, December 6-10, 2010.

6. S. Djahel and F. Naït-Abdesselam. **Thwarting Back-off Rules Violation in Tactical Wireless Ad hoc Networks**. *IEEE Symposium on Computers and Communications (ISCC'10)*, Riccione, Italy, June 22-25, 2010.

7. S. Djahel, F. Naït-Abdesselam and D. Turgut. **An Effective Strategy for Greedy Behavior in Wireless Ad hoc Networks**. *IEEE Global Communications Conference (IEEE GLOBECOM)*, Hawaii, USA, November/December 2009.

8. F. Ahsan, S. Djahel and F. Naït-Abdesselam. **Neighbor based Channel Hopping Coordination: Practical against Jammer?**. *The 34th IEEE Conference on Local Computer Networks (LCN'09)*, Zurich, Switzerland, 20-23 October, 2009.

9. S. Djahel, F. Naït-Abdesselam and Ashfaq Khokhar. **A Cross Layer Framework to Mitigate a Joint MAC and Routing Attack in Multihop Wireless Networks**. *The 34th IEEE Conference on Local Computer Networks (LCN'09)*, Zurich, Switzerland, 20-23 October, 2009.

10. S. Djahel and F. Naït-Abdesselam. **A Fuzzy Logic based Scheme to Detect Adaptive Cheaters in Wireless LAN**. *IEEE International Conference on Communications (IEEE ICC)*, Dresden, Germany, June 2009.

11. S. Djahel, F. Naït-Abdesselam and F. Ahsan. **Highlighting the Effects of joint MAC Layer Misbehavior and Virtual Link Attack in Wireless Ad Hoc Networks**. *ACS/IEEE International Conference on Computer Systems and Applications (ACS/IEEE AICCSA)*, Rabat, Morocco, May 2009.

12. Z. Zhang, F. Naït-Abdesselam and S. Djahel. **ARSoS: an Adaptive, Robust and Sub-optimal Strategy for Automated Deployment of Anomaly Detection System in MANETs**. *International Wireless Communications and Mobile Computing Conference (IWCMC)*, Crete Island, Greece, August 6-8, 2008.

13. S. Djahel, F. Naït-Abdesselam and A. Khokhar. **An Acknowledgment-Based Scheme to Defend against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol**. *IEEE International Conference on Communications (IEEE ICC)*, Beijing, China, May 2008.

14. S. Djahel and F. Naït-Abdesselam. **Avoiding Virtual Link Attacks in Wireless Ad Hoc Networks**. *ACS/IEEE International Conference on Computer Systems and Applications (ACS/IEEE AICCSA)*, Doha, Qatar, April 1-4, 2008.

Bibliography

[1] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. *RFC 2501*, Jan. 1999.

[2] I. F. Akyildiz and X. Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9), Sep. 2005.

[3] Ieee standard 802.16-2004, ieee standard for local and metropolitan area networkspart 16: Air interface for fixed broadband wireless access systems. *IEEE*, 2004.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayici. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug. 2002.

[5] T. Monahan. war rooms of the street: Surveillance practices in transportation control centers. *The Communication Review*, 10(4):367–389, Oct. 2007.

[6] D. Reichardt, M. Miglietta, L. Moretti, P. Morsink, and W. Schulz. Cartalk 2000 safe and comfortable driving based upon inter-vehicle-communication. *IEEE Intelligent Vehicle Symposium (IEEE IVS)*, 2002.

[7] F. Naït-Abdesselam, B. Bensaou, and T. Taleb. Detecting and avoiding wormhole attacks in wireless ad hoc networks. *IEEE Communications Magazine*, 46(4):127–133, Apr. 2008.

[8] J. R. Douceur. The sybil attack. *In proc. of the* 1^{st} *International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA*, Mar. 2002.

[9] Ieee 802.11 wireless lan media access control (mac) and physical layer (phy) specifications, ansi/ieee std 802.11. 1999.

[10] D. K. Y. Yau X. Wu. Mitigating denial-of-service attacks in manet by incentive-based packet filtering: A game-theoretic approach. *In Proc. of the* 3^{rd} *international Conference on Security and Privacy in Communications Networks, Nice, France*, Sep. 17-20 2007.

[11] Y. Hu and A. Perrig. A survey of secure wireless ad hoc routing. *IEEE Security & Privacy*, 2(3), May 2004.

[12] L. Khelladi D. Djenouri and N. Badache. A survey of security issues in mobile ad hoc and sensor networks. *IEEE Communications Surveys & Tutorials*, 7(4), Fourth Quarter 2005.

[13] P. Argyroudis and D. O'Mahony. Secure routing for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 7(3), Third Quarter 2005.

[14] T. R. Andel and A. Yasinsac. Surveying security analysis techniques in manet routing protocols. *IEEE Communications Surveys & Tutorials*, 9(4), Fourth Quarter 2007.

[15] Y. Sun W. Yu and K.R.Liu. Hadof: Defense against routing disruptions in mobile ad hoc networks. *In Proc. of the* 24^{th} *IEEE INFOCOM, Miami, USA*, Mar. 2005.

[16] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). *IETF RFC 3626 (Experimental)*, Oct. 2003.

[17] E. Belding-Royer C. Perkins and S. Das. Ad hoc on-demand distance vector (aodv) routing. *IETF RFC 3561 (Experimental)*, Jul. 2003.

[18] D. B. Johnson, Y. C. Hu, and D.Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. *IETF*, Feb. 2007.

[19] M. Amitabh. Security and quality of service in ad hoc wireless networks. *Cambridge University Press,* 1^{st} *edition*, Mar. 2008.

[20] S. Djahel, F. Naït-Abdesselam, and A. Khokhar. An acknowledgment-based scheme to defend against cooperative black hole attacks in optimized link state routing protocol. *In Proc. of the International Conference of Communication (ICC 2008), beijing, China*, May 19-23 2008.

[21] W. Mao. Modern cryptography: Theory and practice. *Prentice Hall publisher*, Jul. 2005.

[22] P. C. van Oorschot A. J. Menezes and S. A. Vanstone. Handbook of applied cryptography. *CRC Press*, 24, Oct. 1996.

[23] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, Nov. 1979.

[24] A. Shamir. Password authentication with insecure communication. *Communications of the ACM*, 24(11): 770–772, Nov. 1981.

[25] D. Hankerson J. Hernandez M. Kirkup M. Brown, D. Cheung and A. Menezes. Pgp in constrained wireless devices. *In Proc. of the* 9^{th} *USENIX Security Symposium, Denver, Colorado*, Aug. 14-17, 2000.

[26] M. Fischlin. Fast verification of hash chains. *In Proc. of the RSA Security Cryptographer's Track, CT-RSA 2004 : topics in cryptology, San Francisco CA*, Feb. 2004.

[27] W. Li H. Deng and D. P. Agrawal. Routing security in wireless ad hoc networks. *IEEE Communication Magazine*, 40(10), Oct. 2002.

[28] J. Chen B. Sun, Y. Guan and U. W. Pooch. Detecting black- hole attack in mobile ad hoc networks. *In Proc. of the* 5^{th} *European Personal Mobile Communications Conference, Glasgow, UK*, Apr. 2003.

[29] H. Nakamaya B. Kannhavong and A. Jamalipour. *In Proc. of the Global Communications Conference (GLOBECOM '06)*, .

[30] P. Martini M. Jahnke E. Gerhards-Padilla, N. Aschenbruck and J. Tolle. *In Proc. of the* 33^{rd} *IEEE Conference on Local Computer Networks (LCN).*

[31] M. Bussmann M. Jahnke, J. Tolle and S. Henkel. Components for cooperative intrusion detection in dynamic coalition environments. *In Proc. of NATO/RTO IST Symposium on Adaptive defense in Unclassified Networks, Toulouse, France*, Apr. 19-20 2004.

[32] N. gentschen Felde Tolle, M. Jahnke and P. Martini. *In Proc. of the* 25^{th} *Military Communications Conference (MILCOM 2006).*

[33] secrets B. Schneider and lies. Digital security in a networked world. *John Wiley & Sons, inc, 1th edition*, 2000.

[34] L. Buttyan and J. P. Hubaux. Nuglets: A virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. *Swiss Federal Institution of Technology, Lausanne, Switzerland, Tech. Rep. DSC/2001/001*, Jan. 2001.

[35] W. Liu Y. Zhang, W. Lou and Y. Fang. A secure incentive protocol for mobile ad hoc networks. *Wireless Networks journal*, 13(5), Oct. 2007.

[36] A. Perrig Y. C. Hu and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *In Proc. of the* 8^{th} *ACM international Conference on Mobile Computing and Networking, Westin Peachtree Plaza, Atlanta, Georgia, USA*, Sep. 2002.

[37] B. Dahill B. Levine C. Shields K. Sanzgiri, D. LaFlamme and E. Belding-Royer. Authenticated routing for ad hoc networks. *IEEE Journal in Selected Areas in Communications*, 23(3), Mar. 2005.

[38] K. Lai S. Marti, T.J. Giuli and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. *In Proc. of the* 6^{th} *annual international conference on Mobile computing and networking (MOBICOM '00, Boston, Massachusetts, USA)*, Aug. 2000.

[39] A. Patcha and A. Mishra. Collaborative security architecture for black hole attack prevention in mobile ad hoc networks. *In Proc. of Radio and Wireless Conference (RAWCON '03), Boston, Massachusetts, USA*, Aug. 2003.

[40] N. Nasser and Y. Chen. *In Proc. of the Internationl Conference od Communication (ICC 07).*

[41] D. Djenouri and N. Badache. Anew approach for selfish nodes detection in mobile ad hoc networks. *In Proc. of Workshop of the* 1^{st} *International conference on Security and Privacy for Emerging Areas in Communication Networks*, Sep. 2005.

[42] D. Djenouri, N. Ouali, A. Mahmoudi, and N. Badache. Random feedbacks for selfish nodes detection in mobile ad hoc networks. *In Proc. of the* 5^{th} *IEEE International Workshop on IP Operations and Management (IPOM'05), Barcelona, Spain*, Oct. 2005.

[43] P. K. Varshney K. liu, J. Deng and K. Balakrishnan. An acknowledgment-based approach for the detection of routing misbehavior in manets. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 6(5), May 2007.

[44] S. S. Ramaswami and S. Upadhyaya. Smart handling of colluding black hole attacks in manets and wireless sensor networks using multipath routing. *In Proc. of the Workshop on information Assurance, United States Military Academy, West Point, NY*, Jun. 21-23 2006.

[45] Y. Nemoto N. Kato B. Kannhavong, H. Nakamaya and A. Jamalipour. *In Proc. of the International Conference of Communication (ICC 2008)*, .

[46] S. Buchegger and J. Y. Le Boudec. *In Proc. of the* 3^{rd} *ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC'02).*

[47] H. Miranda and L. Rodrigues. Friends and foes: Preventing selfishness in open mobile ad hoc networks. *In Proc. of the* 23^{rd} *international Conference on Distributed Computing systems Workshops (ICDCSW'03), Providence, RI, USA*, May 2003.

[48] P.-H. Ho Z. H. Zhang, F. Nait-Abdesselam and X. Lin. *In Proc. of the IEEE Wireless Communications and Networking Conference (WCNC2008).*

[49] D. Djenouri T. Fahad and R. Askwith. On detecting packets droppers in manet: A novel low cost approach. *In Proc. of the* 3^{rd} *international symposium on information Assurance and security, Manchester, UK*, Aug. 2007.

[50] S. M. Yoo M. Al-Shurman and S. Park. Black hole attack in mobile ad hoc networks. *In Proc. of the* 42^{nd} *annual Southeast regional conference (ACMSE'04), Huntsville, ALabama, USA*, Apr. 2004.

[51] N. Kato A. jamalipour S. Kurosawa, H. Nakayama and Y. Nemoto. Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamuic learning method. *International Journal of Network Security*, 5(3), Nov. 2007.

[52] C. Chigan Z. Li and D. Wong. Awf-na: A complete solution for tampered packet detection in vanets. *In Proc. of the Global Communications Conference (IEEE GLOBECOM 08) New Orleans, LA, USA*, Nov./Dec. 2008.

[53] R. K. Ghosh P. Agrawal and S. K. Das. Cooperative black and gray hole attacks in mobile ad hoc networks. *In Proc. of the* 2^{nd} *international conference on Ubiquitous information management and communication (ICUIMC 2008), SKKU, Suwon, Korea*, Jan./Feb. 2008.

[54] V. Varadharajan V. Balakrishnan and U. K. Tupakula. Fellowship: Defense against flooding and packet drop attacks in manet. *In Proc. of the* 10^{th} *IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, Canada*, Apr. 2006.

[55] Z. Kalbarczyk C. Basile and R. K. lyer. Inner-circle consistency for wireless ad hoc networks. *IEEE TRANSACTIONS ON MOBILE COMPUTING*, 6(1), Jan. 2007.

[56] Y. R. Tsai and S.J. Wang. Two-tier authentication for cluster and individual sets in mobile ad hoc networks. *International Journal of Computer and Telecommunications Networking*, 51(3), Feb. 2007.

[57] P. Michiardi E. Altman, A. Kherani and R. Molva. Non cooperative forwarding in ad hoc networks. *In Proc. of the* 4^{th} *IFIP International Conferences on Networking, Waterloo, Canada*, May 2-6 2005.

[58] W. Yu Z. Ji and K. J. Ray Liu. Cooperation enforcement in autonomous manets under noise and imperfect observation. *In Proc. of the* 3^{rd} *Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (IEEE SECON), Reston, VA, USA*, Sep. 25-28 2006.

[59] OPNET Technologies. Opnet modeler. URL http://www.opnet.com/.

[60] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A performance comparaison of multi-hop wireless ad hoc network routing protocols. *4th ACM/IEEE International Conference on Mobile Computing and Networking (ACM MOBICOM), Dallas, Texas, USA*, Oct. 1998.

[61] J. Haerri, M. Fiore, F. Filali, C. Bonnet, C. F. Chiasserini, and C. Casetti. A realistic mobility simulator for vehicular ad hoc networks. *EURECOM Technical Report*, 2005.

[62] K. T. Wong M. K. Awad and Z. Li. An integrated overview of the open literatures empirical data on the indoor radiowave channels delay properties. *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION*, 56(5), May 2008.

[63] S. Djahel and F. Naït-Abdesselam. Avoiding virtual link attacks in wireless ad hoc networks. *In Proc. of ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-2008), Doha, Qatar*, Mar./Apr. 2008.

[64] S. Setia S. Zhu, S. Xu and S. Jajodia. Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach. *In proc. of the* 11_{th} *IEEE International Conference on Network Protocols(ICNP'03), Atlanta, Georgia, USA*, Nov. 4-7 2003.

[65] L. Buttyan S. Capkun and J. P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1), 2003.

[66] A. Boukerche. Handbook of algorithms for wireless networking and mobile computing. *CRC Chapman Hall*, 2005.

[67] V. Gupta, S. Krishnamurthy, and M. Faloutsos. Denial of service attacks at the mac layer in wireless ad hoc networks. *Military Communication Conference (MILCOM), Anaheim, CA, USA*, Oct. 2002.

[68] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. *12th USENIX Security Symposium, Washington, DC, USA*, Aug. 2003.

[69] A. A. Cardenas, S. Radosavac, and J. S. Baras. Detection and prevention of mac layer misbehavior in ad hoc networks. *ACM Workshop on Security of Ad Hoc and Sensor Networks (ACM SASN), Washington DC, USA*, Oct. 2004.

[70] P. Kyasanur and N. H. Vaidya. Selfish mac layer misbehavior in wireless networks. *IEEE Transactions on Mobile Computing*, 4(5), Sep. 2005.

[71] 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*, 12(5), Dec. 2006.

[72] M. Blum. Coin flipping by telephone: a protocol for solving impossible problems. *24th IEEE Spring Computer Conference (IEEE COMPCON)*, 1982.

[73] L. Guang and C. Assi. Mitigating smart selfish mac layer misbehavior in ad hoc networks. *2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (IEEE WiMob), Montreal, Canada*, Jun. 2006.

[74] A. Wald. Sequential analysis. *John Wiley & Sons, New York*, 1947.

[75] Y. Rong, S. K. Lee, and H. A. Choi. Detecting stations cheating on backoff rules in 802.11 networks using sequential analysis. *IEEE INFOCOM,Barcelona, Spain*, Apr. 2006.

[76] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications (JSAC)*, 18(3), Mar. 2000.

[77] A. L. Toledo and X. Wang. Robust detection of selfish misbehavior in wireless networks. *IEEE Journal of Selected Areas in Communications (JSAC)*, 25(6), Aug. 2007.

[78] F. Massey. The kolmogorov smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), Mar. 1951.

[79] V. N. Lolla, L. K. L. Krishnamurthy, S. V. Ravishankar, and C. Manjunath. Detecting mac layer backoff timer violations in mobile ad hoc networks. *26th IEEE International Conference on Distributed Computing Systems (IEEE ICDCS), Lisboa, Portugal*, Jul. 2006.

[80] V. Rohatgi and A. Saleh. An introduction to probability and statistics. *Wiley*, 2001.

[81] S. Verdu and H. V. Poor. On minimax robustness: a general approach and applications. *IEEE transactions on Information Theory*, 30(2), Mar. 1984.

[82] S. Radosavac, J. Baras, and I. Koutsopoulos. A framework for mac protocol misbehavior detection in wireless networks. *4th ACM Workshop on Wireless Security, Cologne, Germany*, Sep. 2005.

[83] M. Cagalj, S. Ganeriwal, I. Aad, and J-P. Hubaux. On selfish behavior in csma/ca networks. *IEEE INFOCOM, Miami, USA*, Mar. 2005.

[84] K. M. Passino and S. Yurkovich. Fuzzy control. *Addison-Wesley*, 1998.

[85] G. Klir and B. Yuan. Fuzzy sets and fuzzy logic: Theory and applications. *Prentice-Hall PTR*, 1995.

[86] S. Heckerman M. sahami, M. Dumais 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.

[87] S. Djahel and F. Nat-Abdesselam. Flsac: A new scheme to defend against greedy behavior in wireless mesh networks. *International Journal of Communication Systems (IJCS), Wiley InterScience Publisher*, 22(10): 1245–1266, Jun. 2009.

[88] M. Gerta K. Xu and S. Bae. Effectiveness of rts/cts handshake in ieee 802.11 based ad hoc networks. *Ad Hoc Networks Journal*, 1(1), Jul. 2003.

[89] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3), Mar. 1984.

[90] Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworksfor wireless ad-hoc networks. *In Proc. of IEEE INFOCOM 04, Hong Kong*, Mar. 7-11, 2004.

[91] V. Padmanabhan K. Jain, J. Padhye and L. Qiu. Impact of interference on multi-hop wireless network performance. *In Proc. of the* 9^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom 03), San Diego, California, USA*, Sep. 14-19, 2003.

[92] A. Tanaka E. Tomita and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science Journal of Elsevier*, 363(1), Oct. 2006.

[93] J. J. Garcia-Luna-Aceves Y. Wang. Performance of collision avoidance protocols in single-channel ad hoc networks. *In Proc. of the* 10^{th} *IEEE International Conference on Network Protocols (ICNP'02), Paris, France*, Nov. 12-15, 2002.

[94] L. M. Feeney and M. Nilson. Investigating the energy consumption of wireless network interface in an ad hoc networking environment. *In proc. of IEEE INFOCOM 01, Anchorage, Alsaska*, Apr. 22-26, 2001.

[95] B. Nagendra R. Jain, G. Babic and C. Lam. Fairness, call establishment latency and other performance metrics. *Technical Report ATM_Forum/96-1173, ATM Forum Document*, Aug. 1996.