Cooperation techniques to improve peer-to-peer wireless Networks security

MANUEL DAVID SERRAT OLMOS

EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

UNIVERSIDAD
POLITECNICA
DE VALENCIA

DEPARTMENT OF COMPUTER ENGINEERING

Ph. D. Thesis

COOPERATION TECHNIQUES TO IMPROVE PEER-TO-PEER WIRELESS NETWORKS SECURITY

Manuel David Serrat Olmos

Ph.D. Advisors:
Dr. Enrique Hernández Orallo
Dr. Juan Carlos Cano Escribá

Valencia, September 2013

First Edition, 2013

ISBN: 978-84-9048-136-3 (printed version)
Publishing reference: 5675

Acknowledgements

Thanks to everyone who has been supporting and encouraging me during this time to attain the Ph. Doctorate. Specially I would like to thank my Ph.D. advisors, Enrique and Juan-Carlos, for their efforts, enthusiasm and patience. And I must not forget to acknowledge my family for their indulgence and support.

Abstract

Computer networks security is a topic that has been extensively researched. This research is fully justified due to the dimensions of the problem faced. Different kinds of networks and a large quantity of network protocols and applications conform a vast research field, where it is possible for a researcher to set his (or her) interests over a set of threats, vulnerabilities, or types of attacks, so devising mechanisms to prevent the attack, mitigate its effects, or repair the final damages, based upon the specific characteristics of each particular scenario.

The Computer Networks Research Group from the Technical University of Valencia (Universidad Politécnica de Valencia) has been researching on computer networks security risks, specially those affecting wireless networks. In previous doctoral works, detection and exclusion methods for dealing with malicious nodes in Mobile Ad hoc Networks (MANETs) had been proposed, from the point of view of every individual network node, using a technique called Intrusion Detection Systems (IDS) based on Watchdog methods. In this scope, we pretend to optimize network throughput removing misbehaved nodes from the network communication processes, a task performed specifically by Watchdog systems.

A way to improve the whole network performance is to use mechanisms for cooperatively sharing information between well-behaved nodes to speed up misbehaved node detection and increase accuracy. Obviously, these mechanisms will have a cost in terms of network transmission overhead and also a small computing time overhead needed. The key issue here is to adequately balance the costs and the benefits related to these cooperation techniques to ensure that the overall network performance is increased if compared with a non-collaborative one.

In this doctoral thesis, we have designed a mechanism to allow individual watchdogs to share reputation information about their neighbour nodes to characterize them as soon as possible. We call this method Collaborative Bayesian Watchdog, which is based on a non-collaborative bayesian version

of a watchdog. We have evaluated our approach through simulation methods, and also by proposing an analytical model to reduce the time and effort needed to evaluate this kind of solutions in different scenarios.

These evaluations showed that the use of the adequate collaboration mechanisms between well-behaved nodes could improve the performance of the watchdog techniques at an affordable cost in terms of computational and messsage overhead.

Resumen

La seguridad en redes de computadores es un tema que ha sido extensamente investigado. Esa investigación se justifica al observar las dimensiones del problema que se afronta. Diferentes tipos de redes y una gran cantidad de protocolos de red y aplicaciones conforman un vasto campo de investigación, donde es posible para un(a) investigador(a) fijar su interés en un conjunto de amenazas, vulnerabilidades o tipos de ataques, diseñando mecanismos para prevenir el ataque, mitigar sus efectos, o reparar los daños causados, basándose en las características específicas de cada escenario en particular.

El grupo de investigación en Redes de Computadores de la Universidad Politécnica de Valencia ha estado trabajando en ciertos tipos de riesgos para la seguridad de las redes de computadores, especialmente aquellos que afectan a las redes inalámbricas. En trabajos doctorales previos, se han propuesto métodos de detección y exclusión para enfrentarse a nodos maliciosos en redes móviles ad hoc (MANETs), desde el punto de vista de cada nodo de la red por separado, utilizando una técnica llamada Sistema de Detección de Intrusiones (IDS, de Intrusion Detection Systems) basada en Watchdogs. En este ámbito, se pretende optimizar la productividad de la red excluyendo de la misma a aquellos nodos cuyo comportamiento no sea considerado adecuado por sus nodos vecinos, de forma que no participen en los diferentes procesos de comunicación de la red. Esta tarea la desarrollarán específicamente los sistemas basado en Watchdogs.

Una posible manera de mejorar el rendimiento puede ser el establecimiento de un mecanismo de cooperación entre nodos legítimos para intercambiar información, de forma que se acelere la detección de nodos maliciosos y se incremente la exactitud de la detección. Obviamente, un mecanismo de este tipo tiene unos costes en términos de información transmitida por la red, y en necesidades de computación en el nodo para el análisis de la información recibida y la obtención de una opinión sobre un nodo concreto. La clave es equilibrar adecuadamente la sobrecarga que estos mecanismos introducen con las mejoras obtenidas si se les compara con mecanismos no colaborativos.

En esta tésis doctoral, se ha diseñado un mecanismo que permita a los watchdogs individuales intercambiar información de reputación sobre sus nodos vecinos, de forma que se puedan caracterizar lo antes posible. Hemos llamado a este método un Watchdog Bayesiano Colaborativo, porque se basa en una versión no colaborativa de un watchdog bayesiano. Hemos evaluado nuestra propuesta no sólo mediante simulación, sino también se ha propuesto un modelo analítico que permita reducir el tiempo y el esfuerzo necesarios para evaluar este tipo de soluciones en diferentes escenarios.

Estas evaluaciones han mostrado que el uso de un mecanismo adecuado de colaboración entre nodos legítimos puede mejorar el rendimiento de las técnicas basadas en watchdogs con un coste asumible en términos de carga computacional y de transferencia de mensajes.

Resum

La seguretat en xarxes de computadors és un tema que ha estat extensament investigat. Aqueixa investigació es justifica a l'observar les dimensions del problema que s'afronta. Diferents tipus de xarxes i una gran quantitat de protocols de xarxa i aplicacions conformen un vast camp d'investigació, on és possible per a un(a) investigador(a) fixar el seu interès en un conjunt d'amenaces, vulnerabilitats o tipus d'atacs, dissenyant mecanismes per a prevenir l'atac, mitigar els seus efectes, o reparar els danys causats, basant-se en les característiques específiques de cada escenari en particular.

El grup d'investigació en Xarxes de Computadors de la Universitat Politècnica de València ha estat treballant en certs tipus de riscos per a la seguretat de les xarxes de computadors, especialment aquells que afecten a les xarxes sense fils. En treballs doctorals previs, s'han proposat mètodes de detecció i exclusió per a enfrontar-se a nodes maliciosos en xarxes mòbils ad hoc (MANETs), des del punt de vista de cada node de la xarxa per separat, utilitzant una tècnica anomenada Sistema de Detecció d'Intrusions (IDS, de Intrusion Detection Systems) basada en Watchdogs. En aquest àmbit, es pretén optimitzar la productivitat de la xarxa excloent de la mateixa a aquells nodes el comportament dels quals no siga considerat adequat pels seus nodes veïns, de manera que no participen en els diferents processos de comunicació de la xarxa. Aquesta tasca la desenvoluparan específicament els sistemes basat en Watchdogs.

Una possible manera de millorar el rendiment pot ser l'establiment d'un mecanisme de cooperació entre nodes legítims per a intercanviar informació, de manera que s'accelere la detecció de nodes maliciosos i s'incremente l'exactitud de la detecció. Òbviament, un mecanisme d'aquest tipus té uns costos en termes d'informació transmesa per la xarxa, i en necessitats de computació en el node per a l'anàlisi de la informació rebuda i l'obtenció d'una opinió sobre un node concret. La clau és equilibrar adequadament la sobrecàrrega que aquests mecanismes introdueixen amb les millores obtingudes si se'ls compara amb mecanismes no col.laboratius.

En aquesta tési doctoral, s'ha dissenyat un mecanisme que permeta als watchdogs individuals intercanviar informació de reputació sobre els seus nodes veïns, de manera que es puguen caracteritzar com més prompte millor. Hem anomenat a aquest mètode un Watchdog Bayesià Col·laboratiu, perquè es basa en una versió no col·laborativa d'un watchdog bayesià. Hem avaluat la nostra proposta no només mitjançant simulació, sinó també s'ha proposat un model analític que permeta reduir el temps i l'esforç necessaris per a avaluar aquest tipus de solucions en diferents escenaris.

Aquestes avaluacions han demostrat que l'ús d'un mecanisme adequat de col·laboració entre nodes legítims pot millorar el rendiment de les tècniques basades en watchdogs amb un cost assumible en termes de càrrega computacional i de transferència de missatges.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Objectives, Contributions and Organization of the Thesis

Periodically, certain research topics catch the attention of a significative percentage of researchers in a particular research field. Once the major advances on the research topic are achieved, only minor advances can be obtained, and the task becomes more difficult. There is a time when, instead of evolving previous solutions, researchers must bridge a major gap to explore other possible paths to address an unresolved issue or to climb a big step in the performance or realiability of existing solutions. Additionally, exploring the applicability of existing solutions to new issues is also a common research activity.

When we started this work, MANETs were a well known research field, with lots of routing protocols and security schemes introduced by very active researchers worldwide. Nevertheless, some small issues in this field remained unaddressed, like the selfish node problem. A selfish node is a MANET node that makes use of the cooperative nature of the MANET for its own benefit but it does not cooperate for the MANET benefit. There are different motivations for this behaviour, like battery savings or additional fees on the use of WAN networks. But once started the study of this topic, we noticed that there was a bigger set of misbehaved nodes in MANETs whose behaviour must be addressed. Of course, not all misbehaviours can be addressed by the same technique, so our focus was set in those nodes which do not forward packets in behalf of other nodes, called generically black hole nodes, a behaviour which could deeply impact the MANET performance.

Some work had been done with certain routing protocols and with some security mechanisms to detect and isolate this misbehaved nodes, but we

detected that there were room for improvements and extensions that could lead to better solutions in terms of performance and/or scope.

1.1 Objectives of the Thesis

The main research objective of this thesis is to explore the applicability and performance level that cooperative (or collaborative) mechanisms could attain when used in wireless networks nodes to deal with certain types of security threats, like nodes behaving selfish or nodes developing a black hole attack. Aligned with previous researches from the Technical University of Valencia Computer Networks Research Group, we focus our attention over these security risks in the scope of Mobile Ad hoc Networks (MANETs). This types of attacks are easily carried out, thus they are expected to severely affect network performance in community-built MANETs' deployment scenarios. If a set of selfish or malicious nodes enter the MANET, their effect over the quality of the service provided by the network could be remarkable, so strategies to cope with this potentially disrupting behaviours should be designed, implemented and evaluated.

An Intrusion Detection System [O'L92], "*refers to those systems which are designed to monitor an agent's activity to determine if the agent is acting as expected or if the agent is exhibiting unexpected behaviour[...]*". Intrusion Detection Systems, or IDS, are generally based on statistical data collection to perform their task, and one of the basic IDS forms is known as **Watchdog** [HCCM10]. Watchdog systems analyze network traffic and detect misbehaviours. Using watchdogs in infrastructureless wireless networks to detect misbehaviours usually leads to obtain a large amount of wrong detections, due to the channel characteristics and node mobility. Wrongly detected misbehaved nodes could imply a network partition. Undetected misbehaved nodes could damage the communication processes between the remaining nodes. So, research path in this area should focus on enhancing two metrics of the detection process: accuracy, in the sense of reducing the presence of wrong detections; and speed, struggling to obtain a characterization of every suspect node as soon as possible. We expect that cooperative techniques will help us in achieving our objectives over these two metrics while increasing as less as possible the network overhead due to message-passing mechanisms.

So the objectives of this thesis are:

- Study the previous works on this research field.
- Confirm the watchdog as an adequate mechanism for detecting misbe-

haviours in MANETs and other networks.

- Introduce collaboration techniques to improve the watchdog performance and its applicability to different networks and protocols.
- Evaluate the watchdog results through simulation.
- Detect watchdog weaknesses and limitations.
- Develop an analytical model to speed up the performance evaluation of the watchdog in changing scenarios.

Summing up, this work is aimed at increasing the security level of MANET deployments by reducing the negative effect that misbehaved nodes could introduce in the MANET performance. So, we do not aim our approach at solving a big set of MANET security threats, because we only will pay attention to black holes. Additionally, we should accomplish this objective in a cost-effective way in terms of time, computational requirements, and communication overhead.

1.2 Contributions

During this doctoral work we have attained the following major achievements:

- We introduce a Collaborative Bayesian Watchdog to detect black hole attacks in Mobile Ad hoc Networks. This approach proves that simple but well designed cooperative mechanisms between individual detection techniques perform better than its non-collaborative equivalents, and the cost of this cooperation is worth in terms of network overhead vs. network performance ratio. This technique has been implemented and validated through simulation, as a common research methodology. Additionally, this implementation could be a suitable initial step to implement this mechanism in a real testbed or commercial product.
- Our proposal has been able to reduce the time required to detect misbehaved nodes while sending a limited amount of packets over the network, resulting in an affordable cost in terms of overall computational and message overhead.
- We propose an analytical model to evaluate the performance of Collaborative Bayesian Watchdogs. Our model has become an efficient and

accurate method to obtain the required results without using a time-consuming simulation process and an expensive post-simulation data analysis. This model is also suitable for other types of networks, such as Delay Tolerant Networks (DTNs). With this model, it is possible to change different parameter values to quickly obtain the detection time and message overhead in different network scenarios.

1.3 Organization of the Thesis

This thesis is organised as follows. In Chapter 2 we introduce the different concepts, technologies, and protocols involved in the problem we want to solve, and we describe the different approaches proposed in literature in the research area.

In Chapter 3 we present our proposal, the Collaborative Bayesian Watchdog. We also present in this chapter an evaluation of this technique through simulation, and we discuss the obtained results.

Chapter 4 shows a basic version of our analytical model to evaluate the performance of a simple collaborative watchdog.

Chapter 5 is dedicated to improve the previous version of the analytical model to better suit the complete Collaborative Bayesian Watchdog approach.

Finally, in Chapter 6 we present a summary of the main results of this thesis, along with some concluding remarks. We also include a list of the publications related to the thesis, and we comment on possible future research works that can derive from the work here presented.

Chapter 2

Related Work and Definitions

This chapter reviews the state-of-the-art on the topics related to wireless peer-to-peer networks, focusing on Mobile Ad hoc Networks, the most common types of attacks they could suffer, and the different proposals in the literature to deal with this kind of attacks. We also introduce some security-related concepts in order to clearly define the scope and area of this thesis.

2.1 Mobile Ad hoc Networks

A Mobile Ad Hoc Network, usually known as MANET, consists of a set of wireless mobile nodes that functions as a multi-hop mesh network in the absence of any kind of networking infrastructure and centralized administration. In these networks, nodes have the auto-organization capacity, and they cooperate to achieve each one's objectives. MANETs have attracted research efforts in the last years because of the increase on the number of available wireless devices and its increasing computing capabilities. In a tipical MANET, a node is a wireless mobile computing device, like a smartphone or a tablet, running the appropriate software, which allows the node to join the network, identify neighbouring nodes, configure the routing protocol, and participate in forwarding activities. Due to their mobile nature, these nodes are battery operated, so energy saving is a key design parameter, which could encourage the implementation of techniques aimed at saving individual node resources against network performance maximization. As a result of their deployment, MANETs rely on cooperation schemes between nodes for a correct operation, that is, every network node involved in a data communications flow not only generates and sends its own packets, but it also forwards packets on behalf of other nodes. Although there are not many

MANET implementations in real testbed scenarios, these networks have been a popular research topic in the last decade, and some technologies developed for them are applicable to other types of networks, such as Vehicular Ad Hoc Networks (VANET), which are a type of MANET formed by vehicles with the aim of implementing Intelligent Transport System platforms [YMF06].

We can characterize a MANET as:

- A network which is based on wireless technologies, forming a non-regular mesh structure of bidirectional communication links between nodes which cooperate to achieve communication among nodes.
- A network which is usually built, without any central authority, by battery-powered mobile nodes with different technical specifications and different speeds and movement patterns, so it has a dynamic topology.

In a MANET, a node labelled as A is a neighbour of node B, or viceversa, if both nodes are in range of their respective antennas, they publish their presence and detect the presence the other one. Obviously, nodes in a MANET can only send packets to their respective neighbours. So, MANET nodes must use a multi-hop route formed by cooperative nodes which relay packets in behalf of other nodes to its final destination. We can define the network diameter as the maximum number of hops between any pair of nodes in the network. MANET communication protocols implement mechanisms to detect new neighbours, to discover routes to distant nodes, and to check whether a packet must be retransmitted to the next hop in a route (Figure 2.1).

As a possible MANET deployment scenario, let's imagine a big park, where a wireless network deployed by the City Council provides Internet access to the citizens who are passing through it only in particular locations, like *hotspots*. In that scenario there is no wireless coverage in the major part of the park. If citizens using their wireless devices near the access points build a MANET with other citizens who are outside of the access points coverage, and these aforementioned citizens forward packets from/to farther ones, and so on, soon almost all the citizens can access the Internet even if they are far from any of the hotspots. Of course, there is a requirement that all the participants must meet in their wireless devices: they have to run compatible MANET protocols which allow this kind of cooperation. But it is even more important that all the citizens must be willing to share its resources for the whole network benefit. In return, when a user moves outside of the direct coverage from any access point, other cooperative nodes, which will be nearer

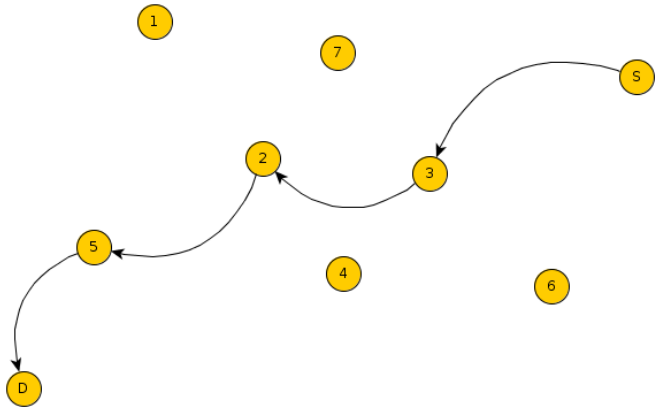


Figure 2.1: Example of Multi-hop packet transmission from node S to node D.

to a hotspot, will forward his (or her) packets, allowing Internet access this way.

The advantages of this kind of networks come up directly from their deployment:

- MANETs do not depend on any pre-existing infrastructure, so they can be deployed indoors or outdoors, wherever they should be necessary.

- MANETs are cheap and easy to deploy. In the case of a community-built MANET, it will be enough with switching on the nodes and letting them to autoorganize.

- MANET tolerate a certain number of node failures due to their mesh topology. Depending on the number of nodes, on the number of possible routes between every source-destination pair, and on the speed and path of each one of them, MANETs are able to readapt them steadily to maintain ongoing communication between distant nodes even in the case of some nodes' failure.

Of course, there are some disadvantages and problems related to the use of MANETs:

- Mobile Ad hoc Networks are not general purpose networks, nor high throughput ones.

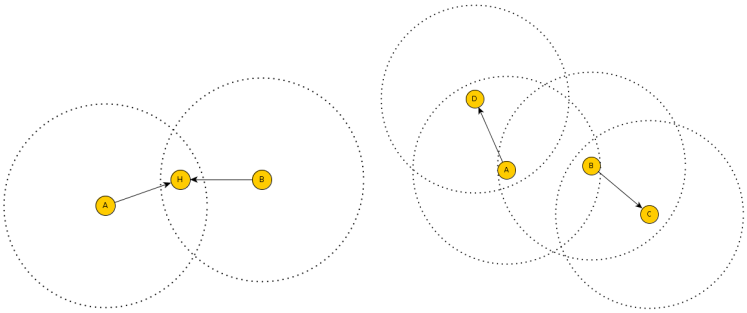


Figure 2.2: (a) Hidden node problem (b) Exposed node problem

- As wireless networks, MANETs are affected by certain physical and medium access level problems, like packet loss, due to interferences or node mobility, and reduced antenna ranges. Two typical problems in wireless networks which could affect performance are known as 'hidden node problem' and 'exposed node problem':
  - The hidden node problem appears when transmissions from two nodes which do not know each other collide at one common neighbour of those nodes. As these nodes can not detect the other one and the communication with the common neighbour has not been acknowledged, they retry to send the packet, colliding again repeatedly. In Figure 2.2(a), a simultaneous transmission from nodes A and B will collide in their common transmission range area near node H.
  - The exposed node problem appears when two pairs of nodes try to send a packet simultaneously and one node of each pair is in trasmission range of the other one. In Figure 2.2(b), a simultaneous transmission from nodes A and B will collide in their common transmission range area, thus nodes D and C wont receive the packets sent.
- Node mobility is also a problem for MANET, specially at high speeds, because it could easily lead to network partitioning. VANETs are very prone to suffer from this problem.

In the previous big park example, each node belongs to a different user, they almost certainly have different hardware, and they could be running different configurations and user software. The absence of a centralized authority has its advantages, but obviously raises some concerns:

- If the nodes belong to different users, this means that everyone could rightfully configure his (o her) own node(s) as he (or she) wants, i.e., prioritizing battery power savings instead of network throughput. Analogously, one can configure a node(s) to be stricter or more relaxed when accepting certain protocols or packet classes, based on security policies or expert advice. The extreme effect of non-centralized administration or network property is the use of incompatible protocols between nodes, resulting in a network partition and a whole network poor performance. In addition to normal people who uncounsciously adds difficulties to the best network performance, we also have to take into account those people who explicitly act maliciously, deploying their nodes in ways that affect the network in terms of lose of information security properties (see section 2.3).

- Different configurations between nodes could also lead to an easier way to disseminate some types of malicious software, like viruses, using those low secured nodes to thrive and to oportunistically infect other nodes.

- Different hardware implementations could mean different computing capabilities, different antenna ranges, different battery capacities, and different driver implementations. In one word, heterogeneity. Every single aspect has its own influence on the node's behaviour and it is able to introduce flaws in some part of every MANET node.

- There are certain drawbacks related to nodes which are near certain popular resources, because they will forward a lot of traffic from other nodes, resulting in an excesive power or processor cycles consumption.

All these issues could pose, *per se,* a wide range of security risks for the individual node, but also for the whole network. In section 2.3, we will discuss some computer security-related concepts, in order to clarify the problem definition and the proposed solutions.

2.2 MANET routing protocols

2.2.1 Taxonomy

One of the most important component of the MANETs' autoorganization feature is the routing protocol. There are lots of them in the literature (see [Mis09]). In fact it has been one of the most popular research topic in this

area. Some of them were designed specially for this type of networks, while others were adapted from protocols developed in other areas. In general, we can classify routing protocols by several criteria [Bou08, Fee99, Gio02, BK09]:

- Depending on the possibilities of changing the message route once the message has been injected into the network, a routing protocol could be:
  - **Adaptative**, if the route can be changed in response to link failures or congestion while the message is on path to its destination.
  - **Deterministic**, if those changes are not allowed once the initial routing decision has been made.
- Depending on where the routing decision is made, a routing protocol could be:
  - **Source** routing, if the decision is made at the source node for the complete message route, explicitly indicating which nodes the message should be forwarded to.
  - **Distributed** routing, if the decision is made in every route's node from the source node to the destination node. In this case, depending on the information stored in every node to allow the decision making process, a distributed routing protocol could be:
    - a **distance vector** routing protocol, if every node stores the next hop and distance (number of hops) for every known destination
    - a **link state** protocol, if every node shares the state of all its active links with all the network nodes. In this case, every node is able to build a complete network topology map.
- Depending on the update mechanism of best routes from source to destination, a routing protocol could be classified as:
  - **Reactive**, if it requires the node to search for the best route when it is necessary to send a message.
  - **Proactive**, if it enforces the node to continuously update its routing table to assure the immediate provision of the best route for a message.

  - **Hybrid**, where some routes are obtained proactively and others are obtained reactively. Obviuosly, the criteria to select one technique or the other is based on the time needed to obtain a new route (for reactive routes) or the network overhead introduced to discover new routes (for proactive routes).

- Depending on the parameter used to decide the best route between two nodes, a routing protocol could be characterized as:

  - **Shortest path**, if it selects the route with the minimum quantity of hops between source and destination nodes.
  - **Shortest time**, if it selects the route with the minimum latency between source and destination nodes.
  - **Shortest weighted path**, if it selects the route which present the shortest weighted path between source and destination nodes, taking into account battery consumption, or available bandwith for every link.

- Depending on the use of topological information about the network, we can find:

  - **flat routing** protocols, if they do not use any topological information.
  - **hierarchical routing** protocols, if they use that information about the network topology.

- Depending on the number of destination nodes, there are:

  - **unicast routing** protocols, if there is only one destination node for a data communication flow.
  - **multicast routing** protocols, when the destination node set count is greater than one.

In any case, a good routing protocol for MANETs must fulfill three requirements:

- The routing dependences graph for every combination of source-destination nodes must not contain cycles. To match this requirement several techniques can be used, like Spanning Tree (Breadth-First Search or Depth-First Search) mechanisms.

- It must operate in a distributed and self-contiguring way, that is, no entity must act as route server or something similar. Every node must be able to obtain a route to every destination node by itself, querying the rest of the nodes in the MANET.

- It must be efficient in front of the dynamic nature of the network, which triggers that nodes and links appear or dissapear frequently.

Of course, there is no protocol which fulfills in a satisfactory manner all those requirements, so the MANET routing protocol selection will usually depend on the number of nodes, its mobility and the amount of traffic which is expected to be injected into the network. It is interesting to refer to the existence of a IETF working group on routing protocols for MANETs [IET]. Although there are a lot of routing protocols for MANETs, in the following sections we will only present the three most important routing protocols for this kind of networks: Dynamic Source Routing (DSR) [DJH07], Ad-hoc On-demand Distance Vector routing (AODV/DYMO) [CPD03, CP10], and Optimized Link State Routing (OLSR) [CJ03]. We must remark here that our proposal for a Collaborative Bayesian Watchdog (see Chapter 3) has been implemented over AODV, although it is protocol-agnostic and can be built over other routing protocols with minor modifications.

2.2.2 DSR: Dynamic Source Routing

As its name denotes, Dynamic Source Routing is a routing protocol where the decision about the route is made in the source node. We can also classify this protocol, using the taxonomy defined in section 2.2.1, as a flat, reactive, deterministic and shortest path routing protocol. In this section we will introduce the basic functionality of this protocol.

Every node in the network periodically sends a small HELLO message to announce its presence to the nodes which are in wireless transmission range, its one-hop neighbours. With this information, every node builds a one-hop neighbour table, which will be useful for certain protocol activities. Of course, every route to a distant node will begin in a neighbour node.

In DSR, once established by the route discovery/maintenance mechanisms, the route is formed by the addresses of the nodes which define the path to reach the target node. In this protocol, the complete route is inserted in every packet's header by this source -or initiator- node. In every hop, the node involved removes its own address from the packet before forwarding the packet to the next hop. This process is repeated until the packet arrives to its destination node, or if an intermediate node can not forward the packet

due to a route failure caused by a node's movement, misbehaviour or failure. In this case, the issue will be notified to the initiator node.

The interesting part of this protocol is the route discovery and maintenance mechanisms. Route discovery allows a node in the MANET to dynamically discover a route to any other node, whether it is directly reachable within its antenna range or it is reachable through one or more hops. The initiator node broadcasts a *route request* (RREQ) packet, identifying the target node, which may be received by those nodes within its wireless transmission range. If the route discovery process is successful, then the initiator receives a *route reply* (RREP) packet which includes a sequence of network hops through which it may reach the target node. Each route request packet also contains a route record, in which the sequence of hops taken by the route request packet is registered as it is propagated through the MANET. When a node receives a route request packet, there are four options:

1. The node has recently received another copy of this RREQ, due to the broadcast nature of this process, so it must discard the current route request packet without further processing.

2. This node's address is already included in the route record in the recently received request. In this case, the node also has to discard the RREQ packet and do not process it further, to avoid route cycles.

3. If the target of the request matches this node's own address, then it must return a copy of this route in a RREP packet to the initiator. In order to do it, the target node must have a route to the initiator. If the target has an entry for this destination in its route cache, then it may send the route reply packet using this route in the same way as is used in sending any other packet. Otherwise, the target may reverse the route in the route record from the route request packet, and use this route to send the route reply packet.

4. Otherwise, the node appends its own address to the route record in the RREQ packet, and re-broadcasts it.

This basic route discovery process has some improvements included in the standard protocol, related to the route maintenance and the use of a route cache. For example, if an intermediate node knows a route to the target node, it completes the route record and sends the route reply packet to the initiator. This mechanism speeds up the process, but it may introduce route errors if the cache information is not fresh and routes are stale. Another optimization included for the route maintenance and discovery process is a

route learning technique for intermediate nodes. When these nodes forward a data packet in behalf of other node, they may include the route to the target node in its own route table. On the same way, when a node participates in a route discovery process, it may include the routes from it to the initiator and the target, processing both the route discovery and route reply packets. The protocol also allows a node to learn new routes by overhearing packets which are transmitted inside its reception range. Finally, when a node forwards a route error packet, it takes care of updating its routing table to delete any route affected by this topology change.

In-transit routing simplicity is the main advantage of DSR. For data flows, at intermediate nodes there is no need of route decision-making, because the route is established and explicitly specified in the data packet itself by the initiator node. So, the nodes can quickly forward the packets with reduced computational requirements. In the other hand, DSR presents some disadvantages[HBT+03]:

- The data packet's header size grows as the network diameter does. A big network could induce big headers, decreasing the valid data sent/total bytes trasferred ratio, thus reducing the network throughput.

- It is not adaptative nor proactive, so a broken link once the packet is in transit to the target node requires the reporting of the broken route, a new route discovery process, and the packet to be re-sent through the new route, all from the source node, not from the node where the link failure has been detected. Thus, this increases the end-to-end latency for a particular packet.

- Additionally, stale route cache information could also result in inconsistencies, but the protocol includes a mechanism to drop routes which have not been used for a particular predefined time.

- At high speeds, the network throughput could degrade due to the proliferation of route error packets, bloating the network with them instead of using the bandwith to transfer valid data.

2.2.3 AODV/DYMO: Ad-hoc On demand Distance Vector/Dynar MANET On demand routing

Ad-hoc On demand Distance Vector is a routing protocol for MANETs that could be characterized as a flat, adaptative, reactive, and shortest path rout-

ing protocol. Unlike DSR, AODV/DYMO implements Distributed On demand routing instead of source routing, so every node stores the next hop and distance for every known destination in its routing table. Like DSR, every node in the network periodically sends a small HELLO message to announce its presence to the nodes which are in its wireless transmission range. Unlike DSR, AODV requires that the communication channel betwen every node must be bidirectional, which commonly represents no problem for wireless nodes.

The explicitly declared design objective of AODV was to improve DSR, trying to avoid its disavantages while keeping its strengths. The first improvement implemented in AODV was the packet header size reduction, because the complete route to the target node is no longer needed in the packet header. In AODV, every node in the path source-destination decides the next hop by reading its routing table. The route discovery process in AODV is very similar to the one outlined for DSR, but must be undertaken whenever a node needs a next-hop to forward a packet to a destination. In AODV, the routing tables do not contain complete routes to destinations, because the only information needed is the next hop and distance for every known destination. As in DSR, every entry in the routing table has been assigned a life time. Once exhausted, the routing table entry will be deleted.

AODV has some advantages if we compare it with DSR. The first one is the reduction in the packet header size, as said previously. The second one is related to the fact that AODV is an adaptative routing protocol instead of a deterministic one, so its reliability is bigger. And finally, the routing table cleaning process is cheaper in terms of computational costs. But, as DSR, AODV does not perform well in scenarios where nodes move at a high speed, and it could have problems dealing with stale routes in its cache.

The IETF MANET working group has proposed some improvements to AODV, mostly in the area of route discovery, and this AODV enhanced version has been called Dynamic MANET On demand (DYMO) routing protocol [CP10]. The standard specification for DYMO is expected to be the main body for the IETF *Reactive MANET Protocol* (RMP) which is currently under development.

2.2.4 OLSR: Optimized Link State Routing

Optimized Link-State Routing (OLSR) [CJ03] is a MANET routing protocol that could be characterized, using the taxonomy introduced in section 2.2.1, as a flat, adaptative, proactive, distributed and shortest path routing protocol whose routing table is built using the link state information, obtained

through HELLO messages. The IETF MANET working group has selected OLSR as the *Proactive MANET Protocol* (PMP), and they have proposed it as a standard for this kind of networks. The proactive protocols periodically flood the network with topological information, which is needed by every node to build a complete network topology map in order to correctly route data packets. The most remarkable OLSR characteristic is its aim to minimize the overhead introduced in the network by the need of periodically flooding it with that topological information.

OLSR uses the concept of **Multipoint Relay** (MPR) to minimize the overhead of flooding messages in the network by reducing redundant retransmissions in the same region. The Multipoint Relays of a node are those neighbours through which two-hop neighbours are reachable from the initial node. As HELLO messages in this protocol include the complete list of neighbours from each node, each one knows its two-hop neighbours. After every HELLO reception, every node must update its MPR set in consequence. This protocol has the consideration of optimized if the MPR set for every node is as small as possible, while guaranteeing that every one of its two-hop neighbours is reachable through each MPR set. The process to build the optimized MPR set for node X is outlined in Algorithm 2.1. Every node includes its MPR set in its HELLO messages.

Algorithm 2.1 OLSR: Building the MPR set for node X

Select one-hop node X's neighbours (namely set $N_1(X)$) which can be used to reach isolated two-hop node X's neighbours (namely $N_2(X)$) which can not be reached through any other node.

Repeat

Select, from $N_1(X)$, the node which has not been selected in Step one and from which the maximum number of $N_2(X)$ set members are reachable.

Until every $N_2(X)$ member has been reached through a selected MPR.

The other concept used in OLSR is the **Multipoint Relay Selector** (MS). Node X is Multipoint Relay Selector for node Y if Y has chosen X as MPR. A broadcast message, intended to be diffused in the whole network, coming from any of the MPR selectors of node Y is assumed to be retransmitted by node Y, if Y has not received it yet. So when a node wants to publish topology information, it sends Topology Control (TC) messages only to the nodes pertaining to its MPR and MS sets. Every node builds its MS set based on the MPR set information received through HELLO mes-

sages from its neighbours. When node Y, which is a one-hop neighbour of node X, receives a TC message, it will process this message and it only will forward it to its own neighbours if X pertains to Y's MS set. This guarantees the minimum amount of retransmissions to propagate link states to the whole network. When this propagation ends, every node has a complete list of available links between nodes, useful to update its own routing table selecting the shortest paths to every possible destination in the network.

The main advantage of OLSR compared to AODV (o DSR) is that the route discovery process does not introduce additional latency to the packet transmission, because routes are always available. The exception to this general case occurs when the need to send a data packet coincide with one route calculation process. But it is arguable that in OLSR there is a continuous network overhead due to the Topology Control packet transmission and the HELLO packet payload. Also, the cost, in terms of power comsumption, could be high [HBT+03]. Depending on the network size and the nodes' mobility parameters, there is a chance that topological changes wont be stabilized in every node when other changes will be produced elsewhere. Additionally, it is also arguable that OLSR introduces computational and store overhead in nodes to obtain the MPR set and the shortest route for every destination node. It appears that OLSR has more disadvantages than advantages for a generic MANET, but there are MANET scenarios where obtaining the minimum end-to-end packet latency is the key design parameter, and in those cases OLSR must be selected instead of AODV/DSR.

2.3 Security concepts

Now, when some routing protocols for MANETs have been introduced, it will be useful to introduce some basic concepts about computer security before we can clearly define the problem we aim this thesis to solve: the black hole attack.

According to the National Institute of Standards and Technology of the U.S. Department of Commerce, **Computer Security** [NIS95] (or Information Security) is defined as *"the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources* [...]". This definition introduces the three basic properties of Computer Security, which form the CIA triad [Cha03]:

- **Confidentiality**: it is a property that assures that private or confidential information is not disclosed to unauthorized individuals.

- **Integrity**: it is a property that has two facets: system integrity and data integrity. System integrity is attained when a computer system performs its intended function without deliberate or inadvertent unauthorized manipulation. Data integrity is a requirement that information and programs are changed only in a specified and authorized manner.

- **Availability**: the system must be in an adequate operating state that allows it to serve promptly legitimate requests from authorized users.

A computer security professional has to enforce this three triad legs, because a weakness in one of them weakens the entire system. Using an easy analogy, computer security is like a Middle Age castled sieged by an enemy army. The defenders must detect and repair every breach they find to avoid the enemy to occupy the castle, under pressure and in a stressful daily basis. But attackers only need one unadverted breach to defeat the defenses and occupy the castle.

Each one of these security properties have its counterpart in the malicious hackers' world, forming which is known as the DAD triad [Cha03]: Disclosure, Alteration and Destruction. These three concepts are the consequences of breaking the information security properties through **attacks**. There are four generic types of attacks [VH02], aimed at affecting one or more security properties of the CIA triad:

- **Interruption**: if the attacker achieves the loose of a system object, or if it leaves it unusable or unavailable. Obviously, this kind of attacks affect the system's Availability and, in certain cases, also its Integrity.

- **Interception**: if the attacker gets unauthorized access to a system object, affecting its Confidentiality.

- **Modification**: if the attacker not only gets unauthorized access to a system object but also he (or she) replaces its content. This kind of attacks damage the system's Confidentiality and Integrity.

- **Fabrication**: when the attacker builds a fake system object to convince other system participants that it is a legitimate one. The effects of this attack depend on the purpose of the system and the intended function of the fake object, so it potentially could affect the three computer security properties.

These four generic attack schemes are depicted in Figure 2.3, where yellow (L) squares represent legitimate system participants and red (A) squares

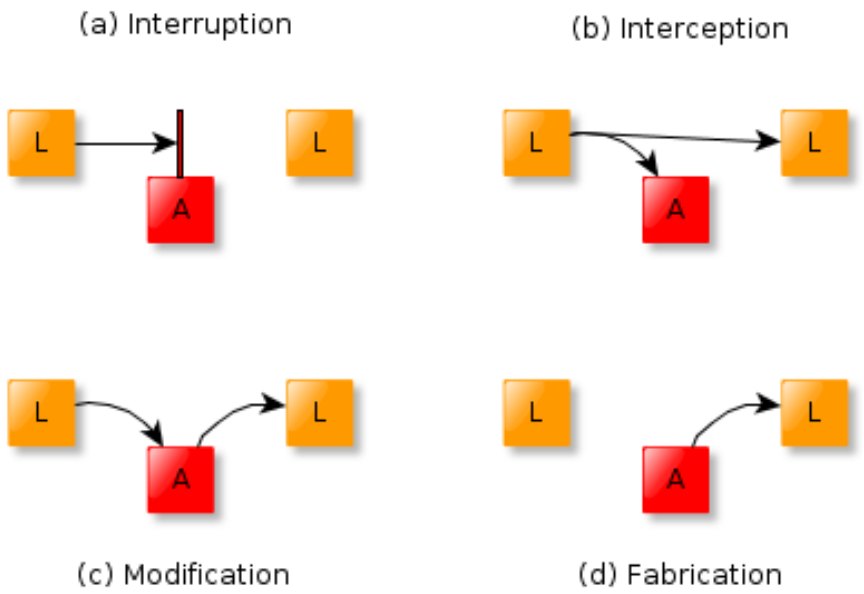


Figure 2.3: Generic attack classes.

represent malicious attackers. In real world, the generic attacks taxonomy is materialized in a large amount of specific attacks and techniques aimed at specific computer system components, but at this point we only want to outline the most widespread types of attacks:

- **Man-in-the-Middle**: It is the easiest attack on MANETs. An attacker intercepts the communication between two legitimate system components, reroutes the communication traffic between them, convincing the attacked components that it is the opposite part of the communication process. It is also a very common attack aimed to steal valuable information between web clients and web servers, even with SSL sessions [Bur02]. In a MANET running the DSR, or AODV or any other protocol, a malicious node could cheat neighbouring nodes to convince them that it is the best next hop for forwarding packets to distant nodes. Once the packets are captured by the attacker, it could perform a variety of malicious activities with these packets, and in MANETs every communication between two distant nodes could be intercepted by any malicious intermediate node.

- **Denial-of-Service (DoS)**: this is an attack performed to interrupt the normal service of the attacked system[RH07], an objective achieved using a variety of techniques. It is very common nowadays in the form of a Distributed Denial-of-Service attacks against web servers on the Internet, when a lot of different malicious clients request service from the attacked server, preventing it from serving legitimate requests or even crashing the service. In a MANET, DoS attacks could be subtile, i.e., when a node does not forward packets in behalf of other nodes (we will discuss this problem later in section 2.4 as the root of the problem we want to solve).

All these attacks and malicious techniques represent *threats* to the system. This threats could also be associated to known, or unknown, *system vulnerabilities*, due to poor programming techniques, system design errors, weak passwords, etc. The probability associated to the materialization of a threat on a information system resource is called a *risk* [VH02]. Computer security can not achieve an aggregated information system risk value of 0, even in very simple ones, due to their intrinsic complexity and heterogeneous components. If we can not guarantee a risk-free computing, we can not say that a computer system is secure. That is the reason why, in this scope, the concept *security* is often replaced by the concept *reliability*, which better defines what we can attain: i.e. a computer system state where "the implemented techniques against threats achieve an affordable level of protection, assuming a certain risk level, associated to those threats which present a very little probability of occurrence, negligible from the system owner's point of view, or their prevention or mitigation techniques are too expensive compared with the values of the assets or resources requiring protection" [VH02].

So, we can say that security is a relative state. In fact, people tend to think they are in a secure state only in two cases. First, if they are not aware of some security risks. For example, this case is typically found in babies, who are not aware of some risks present in their surrounding space and they get hurt in domestic accidents without appropriate adult surveillance. Second, which is more usual, if they assume that their risk level is acceptable to feel secure. For example, when one drives a car, there is an intrinsic risk in speed and the physics behind an unexpected sudden car detention. If we crash at a high speed, the probability of being seriously damaged is also high. As we are aware of this risk, we buy cars with as much security mechanisms as we can afford to reduce the risk of personal damages, and we try to drive carefully. In this case, we mitigate a risk, but we do not remove it. The only way to remove the crash risk is to keep the car stopped and outside the way

of other cars, which is obviously a nonsense. We buy a car to move from one place to another, and we assume and accept a certain level of risk in the task of driving that car as a drawback for the functionality obtained. With computer security we must then balance risks, costs and benefits.

2.4 Misbehaved MANET nodes

MANETs are formed by different types of wireless mobile devices, globally referred as MANET nodes. According to [TKOY10], we will classify those nodes as:

- **Well-behaved nodes**, if they cooperate with the MANET forwarding activities to achieve the community goals. This means that they participate in route discovery processes, providing accurate information to their neighbours, and forwarding data or protocol control packets whenever it is needed.
- **Misbehaved nodes**, if they act against the global MANET goals. In this case, nodes are further classified into three classes:
  - **Faulty nodes**, if they do not cooperate due to a hardware or software malfunction. Their misbehaviour is not counscious, because they are not aware about their disturbing behaviour.
  - **Selfish nodes**, if they drop all the packets whose destination node are not themselves, but they use other nodes to send their own packets, motivated by saving their own resources. Thus, they do not collaborate with the MANET forwarding activities.
  - **Malicious nodes**, if they try to compromise the network security, disturbing the normal behaviour for their own profit, and maybe using multiple potentially damaging techniques.

As we stated earlier, MANETs rely on cooperation between nodes to achieve the maximum network performance. When a MANET is deployed, we have to assume that there could be a percentage of misbehaved nodes. Their number, type, position and movement pattern are key issues which deeply impact the network performance [SS10], but they are *a priori* unknown. So, this network performance could be dramatically reduced if nothing is done to cope with these threats, due to the decreasing packet delivery ratios triggered by the misbehaved nodes dropping packets. To this end, an effective protection against these types of MANET nodes will be mandatory to preserve the correct functionality of the network [KKSW04].

As said previously, in a MANET there are basically two kinds of packet flows: data packets and control packets that implement route discovery and maintenance processes. However, not all misbehaved nodes have the same impact on network performance, due to the type of packet flows they affect. A really malicious node could damage the network by spoofing routes, flooding the wireless channel, or carrying out a man-in-the-middle attack. These are classical attacks that every network could suffer, and solutions have been already devised in literature. It has been clearly stated in section 2.3 that some of these classical attacks can be easily carried out in MANETs because of the nature of the wireless communications channel. However, we are interested in those potential attacks which are specific to MANETs [YHM], and whose effects are significantly worse in this kind of networks. Even if we can achieve a good protection level against certain types of MANET attacks, these kind of networks is prone to suffer other attacks, e.g. Eavesdropping [KN12], Eclipse [SCDR04], or Sybil [Dou02] attacks, whose prevention or remediation techniques would be very interesting but they are outside the scope of this text.

All types of misbehaved nodes –faulty, selfish and malicious– have a common behaviour: they do not participate in forwarding activities, a behaviour which could be classified as a kind of Denial of Service attack. We comprise all these misbehaviour types using the term **black hole**. We define a black hole [HCC+10] as a node that disrupts, intentionally or not, the communication within its neighborhood, dropping the packets received without forwarding them to their final destination. We also include in this definition the concept of **grey hole**, which is a node that selectively forwards only some of the packets, but not all of them.

At this point, it is appropriate to analyze which kind of packets a black hole node could drop. We have to remember that there are two types of packets that must be forwarded by the intermediate nodes along a packet's route from its source node to its destination: control (route request, route reply and route error) packets, and data packets. To participate in MANET communications, nodes must be in routes to other nodes, so the black hole will act again itself if it drops the route request and route reply packets for routes where it probably will be involved. But if the objective of the malicious node is to damage the network performance, dropping route error packets could be a good idea, and dropping data packets is maybe the best one. Thus, we may expect that the misbehaviours we will find could vary from dropping everything, for faulty and selfish nodes, to dropping only route error and data packets, for those intelligent misbehaved nodes.

Our work is aimed at increasing the security level of MANET deployments by reducing the negative effect that black holes could introduce in the MANET performance. So, we do not aim our approach at solving a big set of MANET security threats, because we only will pay attention to black holes. Additionally, we should accomplish this objective in a cost-effective way in terms of time, computational requirements, and communication overhead.

2.5 Proposed approaches

Removing misbehaviour in community-built MANETs is not an easy task. A community-built MANET consists in a set of nodes belonging to a different owner, but all of them willing to cooperate to achieve their individual objectives. The first step to remove misbehaved nodes in that scenario is to detect that misbehaviour. Once detected, certain actions must be taken to mitigate its effects. So, every technique proposed to cope with this threat must have at least two tasks: detect misbehaved nodes and react in consequence. The most important of these tasks is the first one. A good detection is a good start for a good response to a threat, and an efficient detection could be used with different reaction/response schemes. Thus, in our work we have set our focus on the detection mechanism to deal with misbehaved nodes that act as black holes. For the response part of the security scheme, basically there are two approaches in the literature, i.e., isolation and incentivation. Isolation methods are intended to keep the misbehaved nodes outside the network, excluding them from any ongoing communication. Incentivation methods try to convince the misbehaved nodes to change their behaviour, being collaborative instead of malicious. Isolation protects the working network, although it could lead to network partitioning. Incentivation tries to improve the MANET communication capabilities by increasing the number of collaborative nodes and the general collaboration level. Isolation is the only suitable method for all classes of black holes. Incentivation is useful only for selfish nodes (section 2.5.2).

2.5.1 Intrusion Detection Systems

According to [O'L92], the Intrusion Detection System concept "*refers to those systems which are designed to monitor an agent's activity to determine if the agent is acting as expected or if the agent is exhibiting unexpected behaviour[...]*". This definition exactly matches what we are trying to design: a technique which can assess if a MANET node is well-behaved or if it is acting as a black hole. Intrusion Detection Systems, or IDS, are generally

based on statistical data collection to perform their task, and one of the basic IDS forms is known as *Watchdog* [HCCM10]. Watchdog systems analyze network traffic and detect misbehaviours, so they are widely used in most of the proposed approaches from other authors that we will introduce in the next sections.

The main problem that arises at this point when using watchdogs is how to detect misbehaved nodes avoiding as much as possible wrong diagnostics, like false positives or false negatives. A *false positive* appears when the selected technique identifies a well-behaved node as a misbehaved one. A *false negative* appears when the technique can not detect a misbehaved node, so the network believes that it is a legitimate node, with its potentially disruptive effects. Thus, the accuracy level of a black hole detection technique can be evaluated using three cuantitative variables: percentage of real black holes detected (we call it D), percentage of false positives related to the total amount of detections (we call it FP), and percentage of false negatives related to the real amount of black holes (we call it FN). Some appreciations could be done over these metrics:

- D = 100% is the optimal result, but we have to pay attention to the percentage of false positives, because we can achieve a D=100% declaring as a black hole every node in the network, which, depending on the specific response part of the security scheme, could take down the whole MANET. Obviuosly, it is not an adequate strategy, but we have to note that a very strict detection technique can increase D, but it usually also increases FP. So, we have to tune the detection technique to maximize D while minimizing FP to obtain good results.

- The values of D and FN behave oppositely, that is, increasing/decreasing the value of D will decrease/increase the value of FN in the same amount, and viceversa. This is because its addition results in the complete set of existing black holes in the MANET.

Obtaining good results from the detection process in terms of accuracy is one of the desired objectives. But we must select a technique providing a decision over a particular node as soon as possible. First of all, we must not to forget that a MANET consists in a set of mobile nodes, so mobility is a characteristic that makes mandatory to design a quick detection technique capable of obtaining an assesment over a new neighbouring node in a small lapse. If the technique obtains a detection result when the node has left the neighborhood, this technique is useless. So, accuracy and detection speed

are critical issues when designing an approach for black holes detection in MANETs, and the ideal metrics to compare different approaches.

A good watchdog implementation for MANET nodes is required to be as protocol agnostic as possible, that is, no matter which routing protocol the MANET runs, the watchdog obtains very similar metrics. Thus, this aspect must be a design requirement when specifying the watchdog technique we want to build.

2.5.2 Approaches to exclusively deal with selfishness

A selfish node is a MANET node that does not participate in the ongoing communications, saving its own resources. The origins of this behaviour could be a variety of circumstances:

- low battery: in this case, the selfish behaviour should dissapear when the nodes gets their batteries filled up.
- battery saving policy: forwarding packets for other nodes leads to power consumption, and for reduced battery capacity devices could be mandatory to save as much battery as they can in normal operations, avoiding to send packets not belonging to the node.
- misconfiguration: maybe a node should be capable of participate in forwarding activities, but it has not been correctly configured to do so.
- user requirements: the user has decided that his/her device is not available to share its resources with other nodes. It is a human selfishness, not technical.

Anyway, no matter why a node behaves selfish, this behaviour could damage the MANET performance, as the other types of non-collaborative nodes do. In section 2.5.3 we will introduce some generic techniques to deal with the black hole attack, but in this section we will detail two techniques specially designed to cope with the selfish behaviour. Generally, these are incentivation techniques, aimed at re-integrating selfish nodes to the collaborative nodes' set, not to exclude them from the network. In these approaches, the detection and response parts of the technique are not clearly recognizable.

Buttyan and Hubaux [BH00, BH03] presented a method using a virtual currency called *nuglet*. Every node has a credit counter which will be increased when the node forwards packets, and decreased when a node sends his own packets. When a node has no nuglets, it can't send its packets, so it is a strong motivation for nodes to forward other nodes' packets for its

own benefit, and then, for the whole network benefit. The basic component of this protocol is a security module executed by every participating node, which consists in a tamper-proof hardware to store and increase/decrease the nuglet counter. There are some interesting appreciations for this scheme:

- The initial value inserted in the nuglet counter is not expected to be so high to drain the node's battery only sending its own packets.
- At every node, buffers are required to store its own packets, because when the node wont have nuglets it could not send them.
- Associations between neighbouring nodes must be secured through a Public Key Infrastructure, to ensure that the flow of messages and ACK is correctly managed by the security module of each node.

Zhong et al. [ZCY03] go beyond the virtual currency approach present in [BH00, BH03], and they proposed SPRITE, a credit-based system to incentivate participation of selfish nodes in MANET communication. It is based on a Central Clearance System (CCS), which charges or gives credit to nodes when they send or forward a message. So, if a node wants to send a message, it must have sufficient credit to do it. That credit is earned by forwarding messages for other nodes. In essence, their proposal is very similar to the one proposed by Buttyan and Hubaux, but it is a llitle more flexible due to the introduction of a sort of 'credit feeding rounds'. In this rounds, nodes will be rewarded with additional credit by the CCS to ensure that possible lost receipts sent by the nodes to the CCS may take down the network due to a credit crisis. Obviously, this technique introduces more network overhead than the previous one.

As shown, the response module of these two methods is integrated into the incentivation method, so that if a node does not forward other nodes' messages, it will not have credit to send its own messages, and it will be practically excluded from the network until it will have enough credit. However, in general, incentivation methods proposed for MANETs present some basic weaknesses:

- They need some kind of infrastructure to maintain the accounting, so the MANET will lose its 'infrastructureless' characteristic and its functionality will depend on additional elements, which could be affected by other types of vulnerabilities.
- They usually rely on some kind of tamper-proof hardware to store digital certificates or virtual currency amounts, which could be an unaffordable requirement.

- It has been said previously that these techniques do not correctly mitigate the effects of other types of misbehaved nodes other than selfish ones, and it is a risky assumption to believe that this will be the only type of attacker in a MANET environment.

So, once incentivation methods have been discarded as a general solution for the black hole attacks, we will concentrate our efforts in the isolation techniques as a more general response to this problem. We will introduce some of these techniques in the next section.

2.5.3 Approaches to deal with the black holes in MANETs

Several solutions have been proposed for detecting and isolating misbehaved nodes in MANETs. Marti et al. [MGLB00] proposed a Watchdog and a Pathrater over DSR protocol to detect non-forwarding nodes. Their approach maintains a rating for every node (Fresh, Member, Unstable, Suspect or Malicious), based on the surrounding traffic analyzed by the watchdog, and selects routes with the highest average node rating, avoiding those misbehaved nodes. The response module of this technique only relieves misbehaved nodes from forwarding packets, because they are excluded from routes starting and finishing at well-bahaved nodes, but misbehaved node continue getting their traffic forwarded across the network. This technique is very coupled to DSR protocol, and does not properly isolate the misbehaved nodes; its merit stands in the fact that it was the first proposal to deal with this problem.

Another interesting proposal comes from Buchegger and Le Boudec [BLB05]. They proposed the CONFIDANT protocol over DSR, which combines a watchdog, a reputation system, Bayesian filters, and information obtained from a node and its neighbours to accurately detect misbehaved nodes. The system's response is to isolate those nodes from the network, punishing them indefinitely. This approach is very interesting, and it must be acknowledged as the inspiration for our proposal. The basic idea is simple: the system will build an opinion about a node mixing information collected by the watchdog and by its neighbours, just as we usually do in our social relationships.

According to these authors, this kind of technique must have the following functionalities:

- Information representation and classification: to determine how the monitored events are stored, and how they are translated into reputation levels to activate the response mechanism.

- Second-hand information use: which comes from the neighbouring nodes, and that will be used as additional data to obtain the reputation value for each node, taking also into account the effects that malicious nodes could have on the quality of this information.

- Trust: related to reputations, because this trust will be the basis to compute or not the information received from those nodes.

- Redemption and secondary responses: to avoid that an isolated node could not re-integrate into the MANET if its behaviour changes to a well-behaved one.

Other authors, like Kargl, Klenk and others [KKSW04] propose MobIDS as a detection mechanism for malicious nodes. This sytem is basically composed of several software sensors, which are running in parallel to obtain a local evaluation of the surrounding nodes. These local evaluations are shared with other nodes to obtain a global characterization of the node under evaluation. MobIDS only works integrated into a security architecture called SAM, and uses only information from Secure DSR. These are strong prerequisites that turn this approach into a less useful one, excessively tied to a specific environment. On the other hand, the novelty of this approach resides in the utilization of several specialized nodes acting as watchdogs which watch over the neighbouring nodes.

There are other detection mechanisms, more or less similar to the cited above, like CORE [MM02], and SORI [HWKch], also based in some reputation information sharing degree. Finally, there are some approaches designed for certain types of MANETs, like VANETs, whose aim is similar to one reported here [GVKG09].

2.5.4 Standard Watchdog

Once the response module of our IDS is clearly defined as an isolation method, we will deepen in our study on the detection element of these methods, the watchdog, whose basic implementation will be discussed in this section.

The first step to detect misbehaviours is to capture the traffic which a node can *hear* over the wireless channel. To monitor this traffic around a node, the nodes' wireless interface must be able to work in promiscuous mode, capturing all the packets that are sent within the reception range of the node's antenna. A simple watchdog implementation, running in a node, overhears the packets transmitted and received by the node's neighbours,

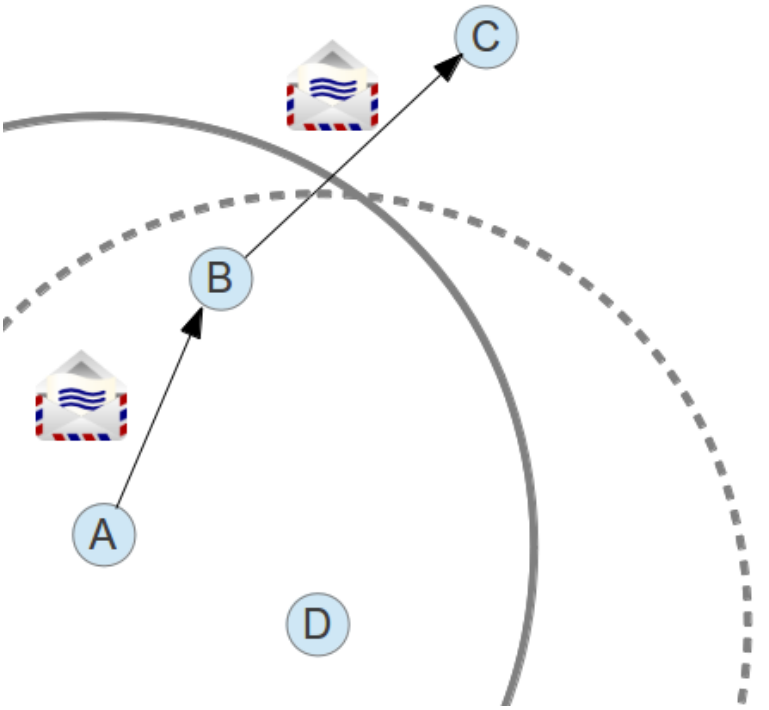


Figure 2.4: Data flow from node A to node C in a MANET.

counting the packets that should be retransmitted by each one, and computing a **trust level** for every neighbour as the ratio of "packets retransmitted" to "packets that should have been retransmitted". If a node retransmits all the packets that it should have retransmitted, it will have a trust level of 1. If a node has a trust level lower than the configured tolerance threshold, that node will be marked as malicious. This tolerance threshold must be tuned to optimize the system performance.

These concepts are illustrated in Figure 2.4, where node D is running a watchdog to detect misbehaved nodes. Node A needs sending a packet to node C, but since node C is outside the neighborhood of node A, the message has to be sent through a multi-hop route which includes node B. In this situation, node D will overhear node A sending the packet to node B, and node B sending the acknoledgement of that packet. So, node D knows that B has to forward this packet because its destination is node C. If node B forwards the packet and **node D overhears this forwarding activity**, it will maintain B's trust level. Otherwise, node B's trust level will be reduced. If B's trust level falls below the tolerance threshold, node D will identify it

as a black hole.

At this point, we must remember that there are two aspects that affect the overhearing capacity of the node running the watchdog, thus influencing the detection process [HCCM10]. They are mobility and transmission errors. Nodes in a MANET move at variable speeds along unknown paths, so maybe node B from Figure 2.4 will be outside the reception range of node D when it forwards the packet from node A to node C. But node D has no way to know whether B has actually sent the packet, and it reduces the trust level of B. The second issue that may affect the detection process is the wireless physical channel, which is prone to transmission errors and interferences. Again, if an interference or packet collision blocks the reception of these signals at D when B fowards the packet, the trust level of node B will be reduced, although its behaviour has been correct. So, if the trust level of node B falls below a tolerance threshold, node D will wrongly identify B as a black hole, generating a false positive detection. Now, let us assume that B is really a black hole node. If B and D are neighbours, there is not enough traffic to let D characterize B as black hole, causing a false negative detection to arise.

In other words, to use a basic watchdog there will be some assumptions we have to keep in mind:

- Conclusions about nodes need a certain amount of traffic overheard by the watchdog for a neighbour to be statistically significative.
- Observations must fade with time, to allow nodes to reintegrate to the network a certain time after been declared misbehaved. This functionality, of course, will only be applicable to those nodes which actually are forwarding packets and some time ago they were not doing so.
- The accuracy level depends mostly in the established threshold. A threshold near to one will probably produce a high level of false positives, leading to a network partitioning. On the other hand, low threshold values will produce many false negatives.

These considerations raise some doubts about if the reliability of this basic implementation is suitable for MANETs, specially when nodes move at high speeds. Studies available in the literature [HCCM10] have shown that this kind of watchdogs are characterized by a significative amount of false positives, basically due to mobility and signal noise over the wireless channel, and that they must be improved to become suitable for a wide range of MANET scenarios. Therefore, we can conclude that the basic watchdog technique is feasible, but unsuitable, for this kind of networks in its current form.

2.5.5 Bayesian Watchdog

As we stated earlier, to detect misbehaved nodes, network monitoring is needed. Every node must be aware of its neighbours' behaviour, and watchdogs are a popular component for Intrusion Detection System dedicated to this task. The main problem is that watchdogs are characterized by a significative amount of false positives [HCCM10], basically due to mobility and signal noise. Previous works from our group have evaluated a bayesian watchdog over Ad-hoc On-demand Distance Vector (AODV) routing in MANETs. This bayesian watchdog results from the aggregation of a bayesian filter with a standard watchdog implementation like the one presented in section 2.5.4. Bayesian filters have been widely used in image treatment software and in anti-SPAM filters for mail systems. The CONFIDANT approach [BLB05] also uses this technique.

The role of the bayesian filter in the watchdog is to probabilistically estimate a system's state from noisy observations [HCC+10]. The mathematical foundation of the bayesian filter is the following: at time t, the state is estimated by a random variable ϑ, which is unknown, and this uncertainty is modeled by assuming that ϑ itself is drawn according to a distribution that is updated as new observations become available. It is commonly called *belief* or $Bel_t(\vartheta)$. To illustrate the concept, let's assume that there is a sequence of time-indexed observations $z_1, z_2, ..., z_n, ..., z_t$. The $Bel_i(\vartheta)$ is then defined by the posterior density over the random variable ϑ conditioned on all sensor data available at time t:

$$Bel_t(\vartheta) = p(\vartheta|z_1, z_2, ..., z_n, ..., z_t) = Beta(\alpha_t, \beta_t, \vartheta) \qquad (2.1)$$

In this approach, the random variable ϑ belongs to the interval [0,1]. Bayesian filtering relies on the Beta distribution[Wal96], a family of probability distributions that stretch from 0 to 1, which is suitable to estimate the belief in this interval, as shown in expression 2.1; α and β represent the state of the system, and they are updated according to the following equations:

$$\alpha_{t+1} = \alpha_t + z_t \qquad (2.2)$$

$$\beta_{t+1} = \beta_t + z_t \qquad (2.3)$$

The Beta function only requires two parameters that are continuously updated as observations are made or reported. In this approach, the observation

Algorithm 2.2 Bayesian Watchdog detection algorithm.

```
Every observation_time Do
  For all Node_j which is a neighbour
    Node_j is well-behaved
    If (BayesianDetection())
    Then Node_j is malicious
    EndIf
  EndFor
EndEvery

Function BayesianDetection()
  Obtain observations
  Compute α and β
  Devaluate observations according to γ
  If relationship between α and β exceeds tolerance Φ
  Then return true
  Else return false
  EndIf
EndFunction
```

z_t represents the information from the local watchdog obtained in time interval $[t, t + \Delta t]$ about the percentage of non-forwarded packets. The bayesian watchdog uses three parameters: the first two parameters are α and β, which are handled over to the Beta function to obtain an estimation of the node's behaviour. Thus, we can say that α and β are the numeric representation of a node's reputation. The third parameter is γ, which represents the devaluation that old observations must suffer to adapt the watchdog's behaviour to a continuously changing scenario without penalizing certain nodes forever. It is a mechanism to re-integrate nodes into the MANET if they change their behaviour into a more cooperative one.

The general functionality of the Detection Module for the Bayesian Watchdog is outlined in Algorithm 2.2. As seen, every configured time lapse, the watchdog evaluates through a bayesian filter all the observations obtained from the traffic overheared. If the relationship between computed α and β exceeds the tolerance threshold, the node is marked as malicious, and the Response Module will act in consequence, generally isolating it from any network communication.

As a result of their work, Hortelano et al. [HCC+10] found that, compared to the standard one, the bayesian watchdog reached a 20% accuracy

gain, and it presents a faster detection on 95% of times. This enhancements allow this technique to be feasible and suitable for MANETs, unlike the standard watchdog. However, although these were good results, a question arises about if they can be enhanced in any way, because this watchdog implementation produces a moderate amount of false negatives and false positives yet.

2.6 Summary

In this chapter we have introduced the MANET concept and its most important routing protocols. We also have presented some basic security-related concepts and we have characterized the different types of MANET nodes from a security-concerned point of view. Finally, it has been presented the state-of-the-art on approaches to deal with different kinds of node misbehaviours. One of this techniques, the Bayesian Watchdog, has been presented as a suitable solution for detecting black holes using locally collected information. But this solution still presents room for improvements in the area of false positives and false negatives production. In the next chapter, we will introduce our proposal to deal with the threat that black hole attacks represent for the MANET existence and performance, which is based in the Bayesian Watchdog.

Chapter 3

A Collaborative Bayesian Watchdog

In this chapter we present our proposal to deal with the generic concept of misbehaving nodes in MANETs. Then we will continue with the implementation details of our Collaborative Bayesian Watchdog. And finally, we will evaluate our proposal through simulation in different scenarios.

3.1 Our approach

In the previous chapter we have claimed that the Bayesian Watchdog technique is good enough to detect black hole attacks in MANETs. However, this technique still presents a lack of accuracy, because it produces a moderate amount of false positives and false negatives. Additionally, detection speed is a performance metric that it could be enhanced. So, in this Chapter we will introduce a Collaborative Watchdog to improve the detection speed and accuracy of the watchdog. Cooperation, or collaboration, is a trademark in MANET environments, so why not combine individual watchdog results with information coming from other nodes to obtain a collaborative detection system?. This is the basic concept behind the proposal of our collaborative bayesian watchdog. Every single node running an instance of a bayesian watchdog combines its direct observations with reputation information received from its neighbouring nodes. The basic assumption in this approach is the *honest majority principle* [PP05], which assumes that the majority of nodes are likely to be well-behaved. It is arguable that this approach could be affected by other types of attacks (as we analyze in section 3.5) and, as a message passing technique, it will generate a little amount of traffic overhead

in the MANET. Nevertheless, this technique has shown to be an excellent solution for the black hole attack problem in these networks. This asseveration does not only come from our simulation results [SOHOC+12, SO11], but also from an analytical model [HOSOC+12b] that we developed to evaluate the system performance, as we will show in following chapters.

3.2 Our Collaborative Bayesian Watchdog

Using the Bayesian Watchdog as the building block, we want to implement a collaborative bayesian watchdog; a technique based on a message-passing mechanism running in every individual node that allows publishing both self and neighbour reputations [SO11]. Similarly to the bayesian watchdog, the collaborative bayesian watchdog overhears the network to collect information about the packets that its neighbours send and receive. Finally, it obtains the α and β values for its whole neighbourhood, exactly in the same way as those obtained by the bayesian watchdog. We call α and β "first hand information" or "direct reputations". In addition, periodically, the watchdog shares its first hand information with its neighbours, for example, stuffing HELLO messages with this information. We call this information "second hand information" or "indirect reputations". Of course, indirect reputations must be modulated using a parameter δ, which represents the confidence degree that a node will put on other node's information about its common neighborhood. Whenever required, every node running the collaborative bayesian watchdog calculates, using expressions 3.1 and 3.2, the values of α' and β', which in this case are passed to the Beta function to obtain an estimation of the maliciousness of a node.

$$\underset{j \in N_i}{\forall} \underset{k \in N_j}{\forall} \alpha(i)'_j = \frac{\alpha(i)_j + \delta \cdot mean(\alpha(i)^k{}_j)}{2} \tag{3.1}$$

$$\underset{j \in N_i}{\forall} \underset{k \in N_j}{\forall} \beta(i)'_j = \frac{\beta(i)_j + \delta \cdot mean(\beta(i)^k{}_j)}{2} \tag{3.2}$$

where

- i is the node which is performing detection
- N_i is the neighbourhood of node i
- $\alpha(i)_j$ is the value of α calculated for every neighbour j of i, obtained from direct observations at i

- $\beta(i)_j$ is the value of β calculated for every neighbour j of i, obtained from direct observations at i
- $\alpha(i)^k{}_j$ is the value of α calculated for every neighbour j of i, obtained from observations of every neighbour k *of* j
- $\beta(i)^k{}_j$ is the value of β calculated for every neighbour j of i, obtained from observations of every neighbours k *of* j
- δ represents the level of trust or the relative importance that a neighbour's observed reputations have for node i

Basically, according to expressions 3.1 and 3.2, each node computes the weighted average reputations for every one of its neighbours, based on the reputation information received from the rest of them (i.e., the term $\delta{\cdot}mean(\alpha(i)^k{}_j$ in expression 3.1). This calculated values are then operated to average them with the direct reputation data obtained by the node itself.

When indirect reputations arrive at a node from one neighbour node, it only processes those reputations for its own neighbours, since reputations about nodes that are not in its neighbourhood are not very useful at that moment. Once the reputations for every neighbour have been obtained, the watchdog obtains the ratio between α' and β'. Then, the detection only needs a predefined tolerance threshold to compare with, thus identifying whether a node is a misbehaved one.

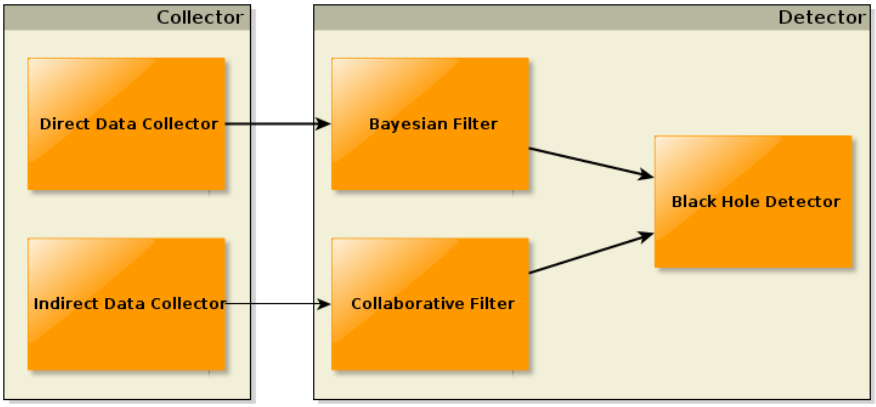


Figure 3.1: Main components of the Detection Module for the collaborative bayesian watchdog.

Figure 3.1 shows the main components of the Detection Module for our

Algorithm 3.1 Black Hole Detector processing algorithm for the Collaborative Bayesian Watchdog.

```
Every observation_time Do
  For all Node_j which is a neighbour
    Node_j is well-behaved
    If ( BayesianDetection() or CollaborativeDetection() )
    Then Node_j is malicious
    EndIf
  EndFor
EndEvery

Function BayesianDetection()
  Obtain observations
  Compute α and β
  Devaluate observations according to γ
  If relationship between α and β exceeds tolerance Φ
  Then return true
  Else return false
  EndIf
EndFunction

Function CollaborativeDetection()
  Obtain neighbourhood reputations
  Compute α' and β'
  If relationship between α' and β' exceeds tolerance Φ
  Then return true
  Else return false
  EndIf
EndFunction
```

collaborative bayesian watchdog. First, each individual watchdog overhears the network to make direct observations of its neighbours, thereby detecting black holes as the bayesian watchdog does. Periodically, it receives reputation information coming from its neighbours and evaluates their behaviour taking into account this second hand information as well as its direct observations.

The functionality of the detector module is outlined in Algorithm 3.1. Basically, the BayesianDetection function performs an analysis over direct observations, obtaining α and β, as seen previously. If the relationship be-

tween α and β exceeds a predefined tolerance level Φ, the watchdog identifies that node as malicious. These values of α and β for every neighbour are then passed to the CollaborativeDetection function, which operates them with second-hand information weighted with parameter δ, according to expressions 3.1 and 3.2, to obtain α' and β' for the same neighbours set.

For the sake of clarity, we will describe how our watchdog works through the following example based on the MANET from Figure 3.2. Table 3.1 shows the second hand information received by node A from its neighbours[1].

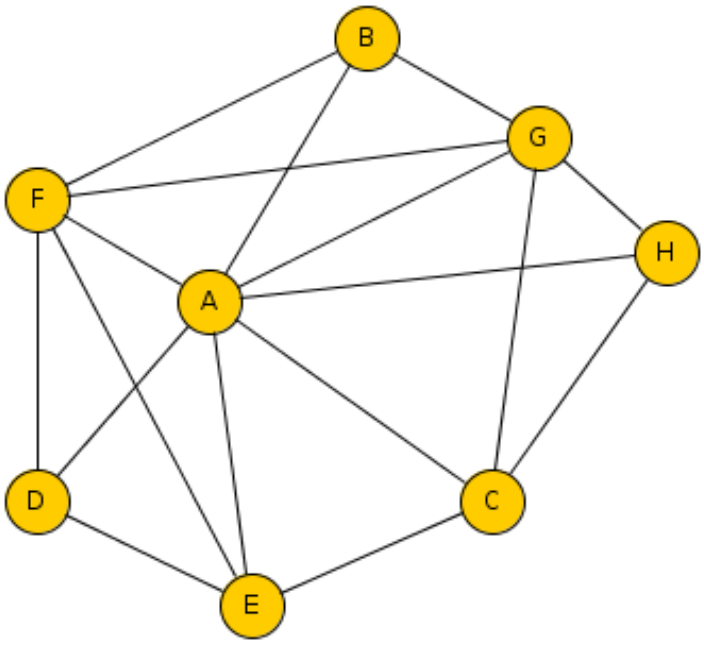


Figure 3.2: MANET for describing the Collaborative Bayesian Watchdog functionality.

Node A combines data from Table 3.1 with the direct reputations obtained by itself, and, for the sake of simplicity, it uses a δ value of 1 in this example; the tolerance threshold Φ is set to 50[2]. These operations are executed in every node running our collaborative bayesian watchdog with its own received and produced data, but in this example we show in Table 3.2

[1]Information received about itself by node A is discarded, and it is not shown here

[2]The tolerance threshold configured here raises a black hole detection alarm if α' is 50 times bigger than β'

Neighbour	Reputations received ($\{\alpha(A)^k{}_j, \beta(A)^k{}_j\}$)
B	F: {5,1}, G:{11,1}
C	E:{1,4}, G:{18,1}, H:{1,1}
D	E:{1,2}, F:{7,1}
E	C:{34,1}, D:{1,6}, F:{15,1}
F	B:{1,1}, D:{1,4}, E:{1,3}, G:{13,1}
G	B:{1,2}, C:{52,1}, F:{27,1}, H:{1,6}
H	C:{21,2}, G:{2,13}

Table 3.1: Second hand information received in node A

only the values obtained at node A.

Neighbour	Reputations		$\{\alpha(A)'_j, \beta(A)'_j\}$	Detected as Black Hole?	
	Direct	Indirect		Bayesian	Collaborative
B	{1, 2}	{1, 1.5}	{1, 1.75}	No	No
C	{43, 1}	{57, 1}	{50, 1}	No	Yes
D	{1, 4}	{1, 5}	{1, 4.5}	No	No
E	{1, 1}	{1, 3}	{1, 2}	No	No
F	{1, 4}	{14, 1}	{7.5, 2.5}	No	No
G	{3, 1}	{14, 1}	{8.5, 1}	No	No
H	{68, 1}	{44,1}	{56, 1}	Yes	Yes

Table 3.2: Values of collaborative reputations calculated at node A of the example

As Table 3.2 shows, node A will identify node H as a black hole using the bayesian and collaborative versions of the watchdog, because both α and α' are 50 times bigger than β and β', respectively. However, only node C will be detected as malicious by the collaborative version, reducing the false negative ratio, and thus improving the watchdog accuracy. In the next section we will evaluate the improvements of this approach through simulations.

3.3 Simulation Performance Evaluation

The goal of this section is to evaluate the local improvements of our Collaborative Bayesian Watchdog compared to previous versions of watchdog implementations. The main objectives of the collaborative bayesian watchdog are shown graphically in Figure 3.3. In that Figure, we show a set of nodes,

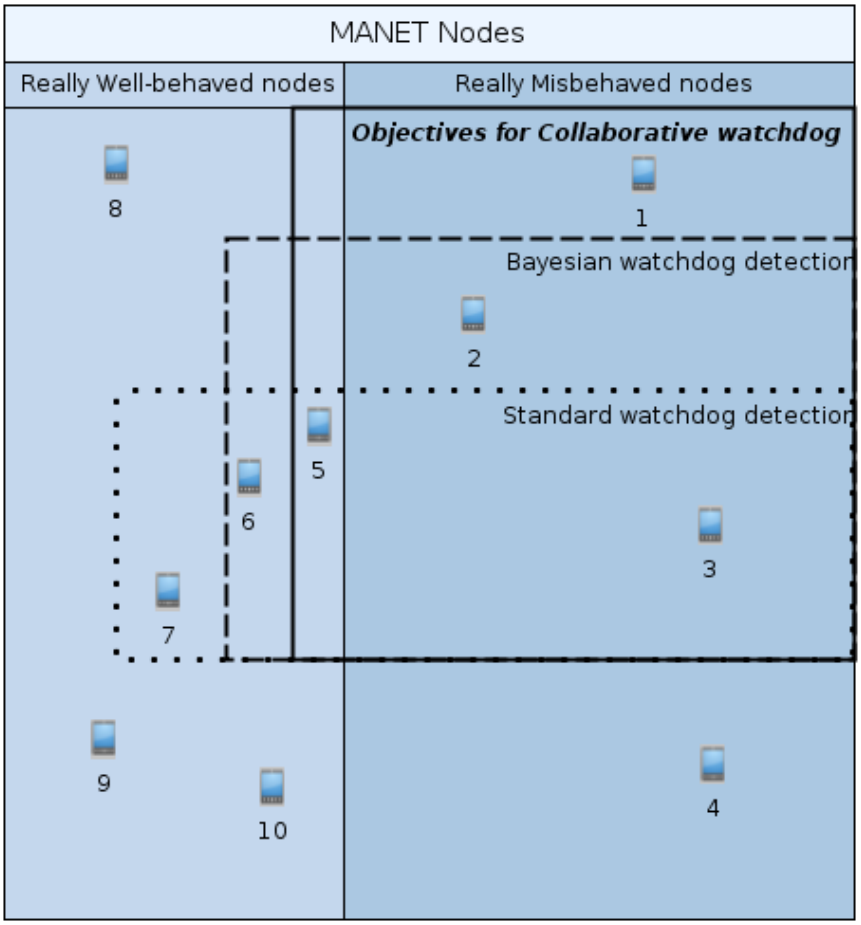


Figure 3.3: Graphical representation of the evaluation objectives for the Collaborative Bayesian Watchdog.

classified as well-behaved and misbehaved. Over them, we draw three subsets representing which nodes are detected as misbehaved by the two types of non-collaborative watchdogs shown previously in sections 2.5.4 and 2.5.5, and the expected results for our collaborative watchdog. Additionally, in Table 3.3 we summarize these expectations.

We have implemented our collaborative bayesian watchdog as a Network Simulator 2 (ns-2) extension to the AODV routing protocol, although this implementation is **protocol-agnostic**. Once implemented, we have evalu-

Node #	Positives	Negative	False Positives	False Negatives
Standard	3	8,9,10	5,6,7	1,2,4
Bayesian	2,3	7,8,9,10	5,6	1,2
Collaborative	1,2,3	6,7,8,9,10	5	4

Table 3.3: Summary of detection results for the three types of watchdogs in Figure 3.3.

ated the impact that this approach has over the accuracy and the detection speed, comparing the results from our collaborative bayesian watchdog with those obtained with the non-collaborative version, the bayesian watchdog[3]. Table 3.4 shows the characteristics of the scenarios we have selected for our performance evaluation.

Parameter	Value
Nodes	50
Area	1000 x 1000 m.
Wireless interface and bandwith	802.11 at 54 Mbps
Antenna	Onmidirectional
Antenna range	250m.
Node speed	5, 10, 15 and 20 m/s.
% of black holes	10%
δ	0.8
γ	0.85
Φ	50
Fading	1
Neighbour time	1s.
Observation time	0.2s.
UDP Unicast traffic	Three flows
UDP Broadcast traffic	every 5s.
Simulation time	352 s.
Scenarios	20

Table 3.4: Simulation parameters

[3] Comparing the Colaborative Bayesian Watchdog to the Standard Watchdog lacks of interest, because as it has been demonstrated in [HCC+10], even the non-collaborative Bayesian Watchdog performs much better than the Standard Watchdog. Thus, it makes sense to compare our collaborative version only to the non-collaborative one.

Some of these parameters, like the area, the number of nodes or the speed, are needed by ns-2 to execute the simulation. Others, like δ, γ, Φ, or *Observation time*, are needed by our code as input parameters. For each test, we averaged the results of 20 independent simulations. To obtain normalized results, we simultaneously executed a simulation of the bayesian watchdog, and the collaborative bayesian watchdog with the same scenarios and parameters.

3.3.1 Evaluating the detection speed

Accuracy is a key issue when detecting black holes, but speed is also important. A watchdog that detects 100% of black holes but requires 10 minutes is a useless watchdog. So, it is crucial that accuracy and speed will be well balanced. In that sense, watchdog enhancements will target both speed and accuracy issues.

Table 3.5 shows that, on average, 7% of the times our approach detected black holes before the bayesian watchdog, with the same traffic pattern. For the rest of the cases, it detects the malicious nodes at the same time. When a node B enters[4] node A's neighbourhood, our approach allows node A to identify node B as a black hole with only a reputations sharing phase with its common neighbours. This means that even if node B does not send or receive any data or routing packet when it enters node A's neighbourhood, if it has been previously detected as black hole, node A will quickly mark it as a black hole too.

Node Speed (m/s.)	Percentage of earlier detections
5	1.04%
10	11.88%
15	9.66%
20	5.72%

Table 3.5: Percentage of detections where the Collaborative Bayesian Watchdog detects the black holes before the Bayesian Watchdog

In dense networks with traffic load equally balanced between malicious and well-behaved nodes, both watchdog versions will perform nearly equally, despite of the smaller number of packets that the collaborative bayesian

[4]In this context, entering a node's neighbourhood means that this node is within communication range and it announces its presence, for example, through a standard HELLO message.

watchdog needs to perform detections. This is because the interval between packets is very short. Nevertheless, in networks with low traffic load and with black holes that transmit a very small amount of packets, the performance difference between the two approaches could be more significant in terms of time. A single packet would make the difference between detecting a black hole or not, and the collaborative bayesian watchdog obtains better results in those cases.

Additionally, we can say that the collaborative bayesian watchdog obtains the best results at a node speed of 10 m/s. In fact, when nodes move at 10 m/s. and 20 m/s. our approach introduces improvements of nearly 12% and 6%, respectively. These results lead to the conclusion our collaborative bayesian watchdog could be suitable for Vehicular Ah Hoc Networks.

3.3.2 Evaluating the detection accuracy

We now present and evaluate the results about the accuracy of our approach. Tables 3.6-3.9 summarize the results of our simulations. The meanings of the different rows are the following:

- "*(A) % of Accuracy*": it shows the ratio of right detections with respect to the total number of detections. Note that $(100 - Accuracy)$ is the % of false negatives.
- "*(B) % of Coverage*": it denotes the percentage of real black holes present in the MANET that have correctly been detected.
- "*(C) % of False Positives*": it indicates the percentage of detected black holes that are not real black holes.
- "*(D) % Only detector*": it shows the percentage of total detections (right or wrong) where the collaborative bayesian watchdog has been the only one doing that detection

The results show that the detection accuracy (A in Tables 3.6-3.9) is also slightly better than that for the non-collaborative bayesian watchdog, since it is able to reduce the number of false negatives. For example, in Table 3.6, our collaborative bayesian watchdog reduces the false negatives by 1.17%. The fact is that a small amount of black holes, which are not detected by the bayesian watchdog, are now detected by the collaborative bayesian watchdog (row D). In fact, our approach is able to detect cases where a black hole enters and exits from the range of a watchdog quickly. Although there is not a big difference between them, the collaborative bayesian watchdog performs

better in terms of accuracy compared to the bayesian watchdog, despite of the node speed. With respect to the standard watchdog, our approach clearly surpasses it in terms of detection accuracy.

	Standard	Bayesian	Collaborative	Difference
(A) % of Accuracy	61.19	91.15	92.23	1.17
(B) % of Coverage	24.00	30.00	30.00	0.00
(C) % of False Positives	64.00	17.00	17.00	0.00
(D) % Only detector				0.78

Table 3.6: Simulation results for nodes moving at 5 m/s.

	Standard	Bayesian	Collaborative	Difference
(A) % of Accuracy	57.27	96.88	97.39	0.53
(B) % of Coverage	13.00	26.00	26.00	0.00
(C) % of False Positives	37.00	20.00	20.00	0.00
(D) % Only detector				3.77

Table 3.7: Simulation results for nodes moving at 10 m/s.

	Standard	Bayesian	Collaborative	Difference
(A) % of Accuracy	55.45	95.41	96.06	0.67
(B) % of Coverage	22.00	33.00	33.00	0.00
(C) % of False Positives	42.00	18.00	18.00	0.00
(D) % Only detector				0.78

Table 3.8: Simulation results for nodes moving at 15 m/s.

	Standard	Bayesian	Collaborative	Difference
(A) % of Accuracy	40.45	91.57	92.25	0.74
(B) % of Coverage	17.00	37.00	37.00	0.00
(C) % of False Positives	42.00	29.00	29.00	0.00
(D) % Only detector				0.55

Table 3.9: Simulation results for nodes moving at 20 m/s.

3.4 Cost estimations

It is obvious that a message-passing technique introduces overhead in the network, because the information shared by the nodes executing the collaborative bayesian watchdog competes for the channel with data packets and control packets. We propose that this reputation information will be included in standard HELLO packets, used by DSR, AODV, and OLSR protocols. In this case, if we compare the number of messages sent by a node using a non-collaborative watchdog and those sent by a node running our collaborative watchdog, there will be no difference between them. However, the key issue here is how to reduce the total amount of **bytes transferred** between nodes when they exchange reputation information, affecting as low as possible the data packets' transmission. The amount of information that a node will send to its neighbours depends only on two dimensions: the size of the neighbourhood and the interval established for the sharing process. The bigger its neighbourhood is, and the shorter the interval is set, the greater the total amount of data transferred. Also, the way individual watchdogs send this information could increase the total amount of data exchanged. In an *in-band* protocol, this information is attached to other protocol messages. In an *out-of-band* protocol, the reputation information will be sent using special-purpose packets. Since each packet introduces overhead due to headers and trailers introduced by the different network layers, we propose compressing the reputations data and insert them in standard HELLO packets (*in-band* protocol). Let be

- S_H: size of standard HELLO messages (bytes).
- S_R: size of a single node's reputation record (bytes).
- H: overhead introduced in the network by protocols at lower layers for every packet transmitted (bytes).
- N: total number of nodes in the MANET.
- n: average number of neighbouring nodes for every node during the life of the network or simulation time.
- t_h: interval between two consecutive HELLO messages sent by a single node (seconds).
- t_m: interval between two consecutive sharing reputation messages sent by a single node in a *out-of-band* protocol (seconds).

- r: compression ratio (r:1). On average, a compresión ratio of 2:1 reduces the size of data by 50%.
- TC_i: total cost of an *in-band* protocol (bytes).
- TC_o: total cost of an *out-of-band* protocol (bytes).

In order to compare the incurred overhead we need to introduce the cost of transmitting the HELLO messages and the special packets. Equations 3.3 and 3.4 analyze the generalized overhead generated by an *in-band* protocol and *out-of-band* protocol for a given time T, respectively.

$$TC_i = (H + S_H + (\frac{S_R \cdot n}{r})) \cdot N \cdot \frac{T}{t_h} \tag{3.3}$$

$$TC_o = (H + S_H) \cdot N \cdot \frac{T}{t_h} + (H + (\frac{S_R \cdot n}{r})) \cdot N \cdot \frac{T}{t_m} \tag{3.4}$$

Note, that the out-of-band equation 3.4 includes the cost of transmitting the HELLO message. Let's denote $(\frac{S_R \cdot n}{r})$ *as* S_C, the average size of compressed complete neighbourhood reputations, and substitute it in expressions 3.3 and 3.4. To compare the costs for each alternative, let's substract TC_i from TC_o, as:

$$TC_o - TC_i = ((H+S_H) \cdot N \cdot \frac{T}{t_h} + (H+S_C) \cdot N \cdot \frac{T}{t_m}) - (H+S_H+S_C) \cdot N \cdot \frac{T}{t_h} \tag{3.5}$$

Next, we operate Expression 3.5 to simplify it:

$$TC_o - TC_i = N \cdot T \cdot (\frac{H + S_C}{t_m} - \frac{S_C}{t_h}) \tag{3.6}$$

For a similar accuracy and speed results between the *in-band* protocol and the *out-of-band* protocol, we have to agree that the average time between two HELLO messages and the average time between two special reputation packets must be similar, so we can say[5] that $t_h = t_m = t$. Making this substitution in Expression 3.6, the difference between both approaches is $TC_o - TC_i = N \cdot H \cdot \frac{T}{t}$, that is, the difference in bytes between these proposals is, at least, equal to the number of nodes, multiplied by the header size for each packet, multiplied by the average number of times a reputation packet is

[5] *Note that if $t_h < t_m$, the performance of the watchdog using the out-of-band protocol will decrease, and if $t_h > t_m$, its overhead will clearly increase with no guarantee of better performance.*

sent. This value is obviously greater than zero, because none of the operands can be zero. Thus, we can say that using an *out-of-band* protocol to exchange reputation information is more expensive than using an *in-band* protocol, not only if we compare the number of messages transmitted, but also in terms of bytes transmitted.

As a result, we can conclude that our approach could **allow saving a significant amount of bandwith** while achieving the results previously shown in terms of black hole detection. In some routing protocols, like in OLSR, HELLO packets are actually stuffed with neighbourhood information, so it will be easy to develop a minor protocol revision to include reputation information in those packets. On the other hand, by using HELLO packets to send reputation information we increase the dependence of our watchdog proposal to the underlying routing protocol.

3.5 Weaknesses and known limitations

Due to its current definition, our watchdog is not capable to deal with certain type of attacks closely related to the black hole attack. In this section we analize those issues that had arisen while trying to use our collaboration technique in different related environments, because we consider that it is also important to know which weaknesses and limitations our proposal pesents.

3.5.1 Fabrication attacks and Liars

Our approach relies on collaboration between well-behaved nodes to detect those misbehaved ones. In our proposal, if a node sends false reputations about its neighbours or other nodes, acting as an individual liar [WMHil], it will affect the perception that its neighbours have about the whole neighbourhood, no matter if the fake information is positive or negative. Anyway, the error level introduced in the detection process by this attitude will depend only on the number of well-behaved neighbours around the liar node which send correct information[MLB08]. Thanks to the honest majority principle, and using the δ parameter to modulate the weight of neighbours' opinions in the characterization of surrounding nodes, our approach is almost immune to this problem.

If liars are present in the MANET, it will be very easy to deploy a defense against them using its own reputation. Let node L be a liar node, and let nodes A, B, C, and D their neighbours. If node L sends false reputations about node D to nodes A, B, and C, these nodes can compare the reputation

about node D they have, and if a big difference exists between these reputations and those received from node L, in a continuous pattern, L could be marked as malicious and isolated. So reputation information could be useful not only to detect black holes, but also to detect liars. This technique is not yet implemented in our approach.

3.5.2 Cooperative attacks

A cooperative attack is carried out by a set of malicious nodes which act coordinated to damage the network. In this scope, there are two kinds of cooperative attacks which must raise our concerns[6]:

- Cooperative black hole attacks: a group of nodes act as individual black holes cooperating to isolate other nodes and partition the network [TS08]. A more ellaborated attack could consist in two lines of malicious nodes, where the first line accept packets from the attacked well-behaved nodes, and forward them to the black holes in the second line. This attitude would make the misbehaviour detection nearly undetectable from inside the attacked area.

- Cooperative liar attacks: in this case, a group of liars cooperate to diseminate false reputation information.

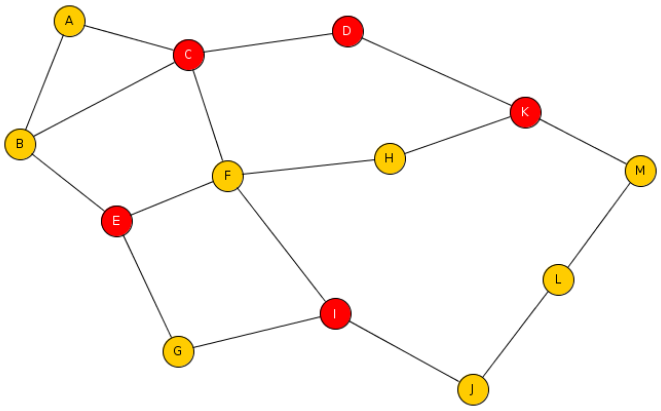


Figure 3.4: Cooperative attack.

[6]Of course, there are other cooperative attacks, like those against the route discovery process, but they are not detailed here because they are out of the scope of this text.

Those nodes must cooperate and move together to stay in touch with each other to perform their attack, surrounding well-behaved nodes, and they can also run a mix of cooperative attacks, lying and dropping packets at a time. In Figure 3.4, red nodes (C, D, E, I, and K) cooperate to disrupt the network. In this example, nodes F and H will be disconected from the rest of the network if malicious nodes act as black holes or if they distribute false reputation information about F and H.

Regrettably, our proposal is not capable, in its current implementation, to detect, prevent or mitigate these types of attacks, except if the attackers are in the limits of the attackers' group, where true reputations can be collected from well-behaved nodes and individual black holes should be detected. However, those detections will not stop the attack, because malicious nodes will continue affecting surrounded nodes, and watchdogs running inside the affected area will not be able to detect them, and surely they will be isolated from the rest of the network. Then, when the group of attackers leave the area, those affected nodes will be isolated until their reputations will not be recalculated by its new neighbours.

3.6 Summary

In this chapter we have introduced our approach to deal with the black hole attack in MANETs. We also have evaluated it through simulation scenarios, and analyzed its costs and weaknesses. In general, our Collaborative Bayesian Watchdog has performed better than the non-collaborative bayesian watchdog, both in terms of detection speed and accuracy, in this latter case, by reducing the amount of false negatives.

But performing these studies in a simulator is a very time-consuming task. First, the simulation parametrization, the scenario generation and the simulation execution. Next, the collected data analysis. Thus, it is a considerable effort behind a detailed study of a moderate amount of scenarios. That the motivation for us to go further in our research to obtain performance results in a efortless way. In the next chapters we will present a model to evaluate these watchdog systems without the need of extensive simulation and post-simulation data analysis.

Chapter 4

An Analytical Model for Collaborative Watchdogs

In the previous chapter we focused on evaluating the local performance of the collaborative bayesian watchdog to detect black hole attacks. In order to evaluate the global behaviour we found that simulation was not feasible, due to its cost in terms of time. Simulating realistic scenarios is a complex and time consuming task, which in addition to the also expensive analysis activities on collected data, lead us to develop an analytical model. Thus, the goal of this task is to model and evaluate the performance of our collaborative bayesian watchdog taking into account the effect of collaboration, false positives and false negatives, using Markov chains.

In this chapter, we introduce the first version of our model for evaluating the detection of black hole nodes, which only takes into account the effect of collaboration and detection probabilities. In the next chapter, a more accurate but complex model is introduced to evaluate the impact of false positives and false negatives.

4.1 A brief introduction to Markov chains

Classical probability studies deal with independent events, where the knowledge of the outcome of previous experiments does not influence the predictions about the outcomes of the next experiment. If we consider the 'Heads or Tails' (coin tossing or throwing a coin in the air) experiment, the probability of every possible result is not influenced in any way by the result of the previous throwing. If we know that the last result was 'Heads', the probability of the 'Heads' result in the next throwing is the same in this

case that if it has been 'Tails' previously. This kind of probability scenario is called an *independent trial process*.

But not all chance processes are independent. Modern probability theory studies chance processes for which the knowledge of previous outcomes influences predictions for future experiments[GS97], and Markov Chains are a fundamental part of these studies. A Markov Chain could be described as follows: we have a set of *states* $S = \{s_1, s_2, ..., s_\tau\}$, and the process starts from one state s_i and moves to other state s_j with a *transition probability* p_{ij}. Every move is called *step*, but it is possible that a step does not imply a state transition, so it exists p_{ii} as the probability of no transition. Often an initial state is specified as the starting state of the chain, but it is possible to define a probability vector for every possible starting state. Usually, the complete set of transition probabilities is denoted by a square $\tau \times \tau$ **transition matrix P**. If the chain is in state s_i, we denote the probability that it will be in state s_j n steps after as $p_{ij}^{(n)}$. In general, if a Markov chain has r states, the transition probabilities for two steps after the current state will be [GS97]

$$p_{ij}^{(2)} = \sum_{k=1}^{r} p_{ik} p_{kj} \tag{4.1}$$

In fact, the probabilities for n steps after the current state are obtained as the n-th power of matrix **P** (P^n).

When the probability of every transition from a state s_i to any other state is zero, so it is impossible to leave it, this state is called *absorbing state*, and that chain is called *absorbing Markov chain*. In this type of chains, every non-absorbing state is called *transient*. There are some interesting questions related to absorbing Markov chains:

- What is the probability that the system will reach an absorbing state?
- How long will it take for the process to be absorbed, on the average?
- How many times will the process be in each transient state, also on the average?

In general, the answers to these questions depend on the initial state and the probability matrix, but we need to address them because they will be useful for the evaluation of our model.

For an absorbing Markov chain, if the transitional states and absorbing states are reordered and grouped, the transition matrix P in **canonical form** is [GS97]

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix} \tag{4.2}$$

If ν is the number of absorbing states and τ is the number of transient states, $\mathbf{I}$ is a $\nu \times \nu$ identity matrix, $\mathbf{0}$ is a $\nu \times \tau$ zero matrix, $\mathbf{Q}$ is a $\tau \times \tau$ matrix whose elements p_{ij} denote the transition probability between transient states i and j, and $\mathbf{R}$ is a $\tau \times \nu$ matrix whose elements p_{ij} denote the transition probability between a transient state i to an absorbing state j. It is not difficult to derive that the transition matrix after n steps, P^n, is in the following form:

$$P^n = \begin{pmatrix} Q^n & * \\ 0 & I \end{pmatrix} \tag{4.3}$$

where submatrix * could be expressed in terms of $\mathbf{Q}$ and $\mathbf{R}$ but it is too complex to show it here and does not contribute to our goals at this time. Q^n, which represent the transition probabilities between transient states, tends to 0, because transition probabilities p_{ij} in P are in the range [0, 1] and the probability of not being absorbed in n steps is monotone decreasing. So, as n increases, the probability of being in a transient state after n steps approaches zero, so the probability of absorption approaches to 1.

For an absorbing Markov chain $\mathbf{P}$, the matrix $N = (I - Q)^{-1}$ is called the *fundamental matrix* for P. The entry n_{ij} of $\mathbf{N}$ gives the expected number of times that the process is in the transient state s_j if it is started in the transient state s_i.

Additionally, given that the chain starts in state s_i, it will be necessary for us to know what is the expected number of steps before the chain is absorbed. Using the fundamental matrix, we can derive that t_i is the expected number of steps before the chain is absorbed, given that the chain starts in state s_i. This t_i is the ith element of vector $\mathbf{t}$ in expression 4.4

$$t = Nc \tag{4.4}$$

where c is a column vector all of whose entries are 1. This is because if we add all the entries in the ith row of $\mathbf{N}$, we will have the expected number of times the process arrives at any of the transient states, given starting state s_i. Thus, t_i is the sum of the entries in the ith row of $\mathbf{N}$.

Since this point, we have studied the principal aspects of Discrete Time Markov chains, characterized by the transition between states at every step. But we must introduce now the concept of **Continuous Time Markov chain**. In this type of chains, after the previous transition, the process

remains in the current state for some random amount of time and then transitions to a different state. At this point, it is important to cite two additional types of Markov chains, which are not directly implied in our work but are somewhat interesting:

- Regular Markov chains: those processes that long-range predictions for them are independent of the starting state, that is, no matter which was the starting state, a long term prediction on its outcome will always be the same.
- Ergodic (or irreducible) Markov chains: those chains where it is possible to go from every state to every other state, but not necessarily in one move.

Once introduced the mathematical foundations of our model evaluation, in the next section we present the basic model.

4.2 Modelling collaborative detection

It is commonly accepted that building a model of a system requires some simplifications. In our case, the main difference between the watchdog implemented in the simulator and its model relies on the fact that the model does not allow that nodes previously detected as misbehaved could re-enter the well-behaved nodes set after a certain amount of time. This means that if a node B is detected as misbehaved by other node A, this node A will retain this information during its pertenence to the network and it will share it with every node it will contact afterwards, spreading the bad reputation of node B. In this case, we say that node A has a *positive* about node B's maliciousness. So collaboration, in this scope, does not rely on sharing numerical representations of reputations, but relies on sharing a positive for every misbehaved node known, no matter if it has been directly contacted by the reporting node or not.

A collaborative node can have a positive about another node by one of the following ways:

- Misbehaved (or malicious) contact: when a collaborative node detects a misbehaved node through its local watchdog. Our model also allows that even when a contact between a misbehaved[1] node and a well-behaved node occurs, there is a probability that the well-behaved node

[1] Following what we introduced in section 2.4, in this scope we will interchangeably continue using the expressions 'misbehaved node', 'malicious node' and 'black hole' to denote the node whose behaviour must be detected by the watchdog.

does not detect the other node as a malicious one. We model this behaviour as the *probability of detection* (p_d), which depends on the efectiveness of the watchdog and other parameters like the relative node speed.

- Collaborative contact: when two well-behaved nodes contact each other, they share their own list of misbehaved nodes. As in the previous case, we model the *probability of collaboration* (p_c), because a contact does not always imply collaboration. This parameter allows us to adjust the general level of collaboration in the network, from 0 (no collaboration at all) to 1 (full collaboration).

First at all, we model the network as a set of N wireless mobile nodes, with C collaborative nodes and S black hole nodes ($N = C + S$) [HOSOC+12b]. Our goals are: (i) to obtain the time required by all collaborative nodes to realize who are the S black hole nodes in the network, and (ii) to calculate how many reputation messages are generated. To do so, when a contact occurs between two nodes, each one shares the information about which black holes it knows that exist in the MANET. For our model, we assume that the occurrence of contacts between two nodes follows a Poisson distribution with rate λ. This has shown to be valid for both human and vehicle mobility patterns [GNK05, ZFX+10, LSW+11]. There is some controversy about whether this exponential distribution can reflect some real mobility patterns. Empirical results have shown that the aggregated inter-contact times distribution follows a power-law and has a long tail [CHC+07]. In [CE09] it is shown that, in a bounded domain (such as the one selected along this thesis), the inter-contact distribution is exponential but, in an unbounded domain, it follows a power-law distribution. The dichotomy of this distribution is described in [KLBV07]: where a truncated power law with exponential decay appears in its tail after some cut-off point. The work in [GLZC09] analyzed some popular mobility traces, and found that over 85% of the *individual pair distribution* fits an exponential distribution. Therefore, we consider that using an exponential fit is a good choice to model inter-contact times. Moreover, using exponential distributions we can formulate analytical models using Markov chains.

4.2.1 Our basic model

Our basic model asumes that there is only one black hole in the network (S=1), so every node in the network will only be in one of two possible states: NOINFO, if it has no information about the misbehaved node; and

POSITIVE, if it knows which is the black hole node. In our model, at the beginning the collaborative nodes have no information about the rest of the network nodes, so all of them are in a NOINFO state which may change when a contact occurs (see Figure 4.2). In Figure 4.1 (and its associated Table 4.1), at the beginning, nodes 1, 2 and 3 are in the NOINFO state (see Figure 4.1.(a)). After some time, node 2 detects the black hole node using its watchdog (see Figure 4.1.(b)), which is a malicious contact. Some time after, node 3 receives a reputation message from node 2 by a collaborative contact and learns which is the black hole node (see Figure 4.1.(c))[2].

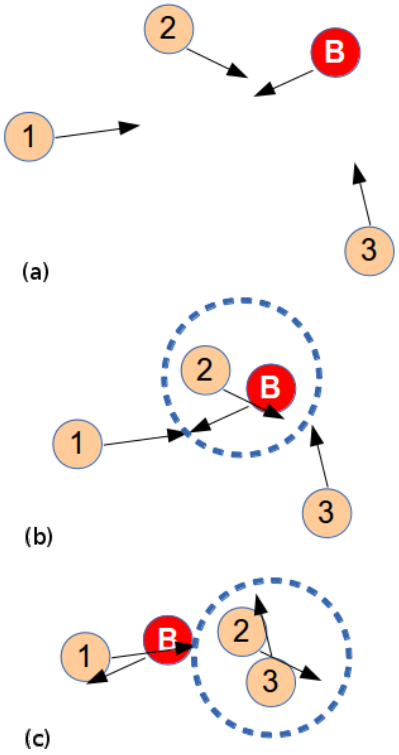


Figure 4.1: Obtaining positives.

[2]Looking at this Figure, node 1 could be in POSITIVE state because it is very close to the black hole, but there is a probability of $(1 - p_d)$ that the black hole will not be detected by the running watchdog at node 1.

Node	Status in (a)	Status in (b)	Status in (c)
1	NOINFO	NOINFO	NOINFO
2	NOINFO	**POSITIVE**	**POSITIVE**
3	NOINFO	NOINFO	**POSITIVE**

Table 4.1: Status table for Figure 4.1

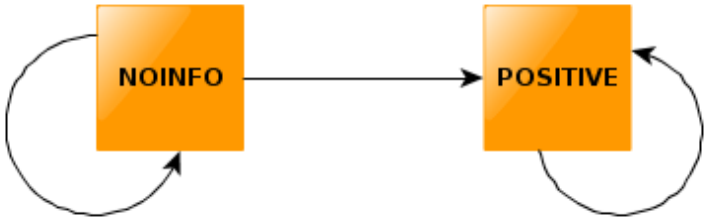


Figure 4.2: State transition diagram when updating information about contacted nodes.

Using a contact rate λ, we can model the network using a Continuous Time Markov Chain (CTMC) with states $s_i = (c)$, where c represents the number of collaboratives nodes in the POSITIVE state. When a contact occurs, c may increase by one if one of the intervening nodes has a POSITIVE. Thus, the final, or absorbing, state is $c{=}C$, and the system could be modelled using a CTMC from initial state $s_1 = (0)$, $\tau = (C-1)$ intermediate states (which include s_1) , and an absorbing state $s_{\tau+1} = (C)$.

Next, we need to obtain the probabilities associated to every possible transition (p_{ij}). Given a state $s_i = (c)$, the possible transitions which can occur are:

- *(c)* to *(c+1)*: this transition takes place when a collaborative node changes from NOINFO state to POSITIVE state. So the probability of this transition could be derived as $t_c = (\lambda{\cdot}p_d + \lambda{\cdot}c{\cdot}p_c)(C - c)$. In this expression, the term $\lambda{\cdot}p_d$ represents the probability of detection due to the accuracy of the local watchdog. On the other hand, $\lambda{\cdot}c{\cdot}p_c$ represents the probability of transmission of the information about the black hole node. This latter term depends on c because the probability of transmission increases as the number of nodes in POSITIVE state does. Of course, factor *(C-c)* represents the number of nodes in NOINFO state.

- *(c)* to *(c)*: this occurrence represents 'no changes'. The probability of this event obviously is $t_0 = (1 - t_c)$.

Once we know the probablities associated to every transition, and using the transition matrix $\mathbf{P}$, it is possible to derive for the detection time T_d and the overall overhead or cost M_d thanks to the mathematical foundations introduced in the section 4.1. For the detection time T_d, we can use a modified version of expression 4.4, as we only need the absorption time for state $s_1 = (0)$:

$$T_d = v_1 N v \tag{4.5}$$

where $v_1 = [1, 0, 0, ..., 0]$.

To obtain the overall overhead or transmission cost, we have to obtain the amount of messages transmitted to publish the information about black holes in the network in each state s_i. It is obvious that in state s_1 all the nodes are in state NOINFO, so the amount of transmitted messages is zero ($m_1 = 0$). State s_2 starts when a node enters the POSITIVE state, so this POSITIVE could only be sent by one sender to potentially all the other nodes except himself for the duration of the state (this duration is denoted as f_2) with a rate λ and probability p_c. We can obtain the duration of every state using the fundamental matrix $\mathbf{N}$, whose first row elements are the expected times in each state starting from state 0. Thus, the expected duration of state s_i will be $f_i = N(1,\text{i})$ Then, the number of messages sent in this state could be obtained as $m_2 = f_2 \cdot \lambda \cdot (\text{C-1}) \cdot p_c$. Analogously, for state s_3 , the number of messages will be $m_3 = 2 \cdot f_2 \cdot \lambda \cdot (\text{C-1}) \cdot p_c$, because there will be two possible senders. Then, for state s_i, the number of messages will be $m_i = (i-1) \cdot f_i \cdot \lambda \cdot (\text{C-1}) \cdot p_c$. Summing up all the messages sent in each state:

$$M_d = \lambda \cdot (\text{C-1}) \cdot p_c \cdot \sum_{i=1}^{\tau} \Phi(s_i) \cdot \text{N(1,i)} \tag{4.6}$$

where $\Phi(s_i)$ is the number of senders in state s_i, that is (i-1).

4.2.2 Enhancing the model to deal with more than one black hole

We can easily extend this model to a more complex one where the number of black holes is larger than one (S>1) [HOSOC+12b]. To do it, we must use a S-dimensional Continuous Time Markov chain, starting with S=2, which conforms a two-dimensions CTMC (for short, a 2D-CTMC). In this case,

every state has two values, $s_i = (c_2, c_1)$, instead of one as introduced in the previous section. The first value represents the number of nodes which have a POSITIVE for black hole 1 and, analogously, the second value represents the number of nodes which have a POSITIVE for black hole 2. When a contact occurs, c_1 and c_2 can increase by one, and the absorbing state $s_{\tau+1} = (C, C)$. So this 2D-CTMC has an initial state $s_1 = (0, 0)$, $(C+1)^2 - 2$ transient states, and a final state $s_{\tau+1} = (C, C)$. The transition rates p_{ij}, given the state $s_i = (c_2, c_1)$ are the following:

- (c_2, c_1) to $(c_2, c_1 + 1)$: it is the same as when S=1, replacing c by c_1, that is, $t_{c1} = (\lambda{\cdot}p_d + \lambda{\cdot}c_1{\cdot}p_c)(C - c_1)$.
- (c_2, c_1) to $(c_2 + 1, c_1)$: it is like the previous case, replacing c_1 by c_2, that is, $t_{c2} = (\lambda{\cdot}p_d + \lambda{\cdot}c_2{\cdot}p_c)(C - c_2)$.
- (c_2, c_1) to (c_2, c_1): analogously to the S=1 case, $t_0 = (1 - t_{c1} - t_{c2})$.

With these values we can build the transition matrix $\mathbf{P}$ and the fundamental matrix $\mathbf{N}$. Then, the detection time T_d can be obtained using expression 4.5 for the initial state $s_1 = (0, 0)$. Once obtained the expressions for S=2, we can derive them for S>2. In general, we have $\tau = (C+1)^S - 1$ transient states and, for every state $s_i = (c_S, c_{S-1}, ..., c_2, c_1)$, the transition rate from c_j to $c_j + 1$ will be $t_{cj} = (\lambda{\cdot}p_d + \lambda{\cdot}c_j{\cdot}p_c)(C - c_j)$.

The only thing which remains to be complete for this model is the generic expression for the overhead. It is assumed that every node transmits only one message with all its POSITIVES when there is a contact. So, to use expression 4.6, we need to establish the number of senders in every state, which could be very complex for high values of S, because the number of messages depends on the distribution of POSITIVES. So another simplification is needed here: we must approximate the value of senders ($\Phi(s_i)$) by bounding it. It is easy to observe that the number of senders in each state is between the maximum of c_j and the minimum between the sum of c_j and C. That is, $max(s_i) \leq \Phi(s_i) \leq min(sum(s_i), C)$, where $max(s_i) = max_{j=1}^{S}(c_j)$ and $sum(s_i) = \sum_{j=1}^{S} c_j$. Estimating that

$$\Phi(s_i) \approx average(max_{j=1}^{S}(c_j), \sum_{j=1}^{S} c_j) \tag{4.7}$$

that is, estimating $\Phi(s_i)$ as the average between the lower and upper bounds, the number of messages could finally be calculated by using expression 4.6.

4.3 Model validation

In this section we describe the validation process of the models presented in the previous sections. In order to validate these models, the results obtained with the models were compared with the simulation results. We implemented all the models and the simulator in Matlab. The simulator is a simple event driven simulator. The network model of this simulator has C collaborative nodes, D destination nodes and S selfish nodes. This simulator generates contact events with a given λ rate. All the nodes have a vector of size S that stores the information about each black hole node. This vector is initialized with no state info and it can change to a positive state. When a contact event occurs, it implements the behavior of the different models, using the probabilities of detection (p_d) and collaboration (p_c) to change the state of a node. The simulation finishes when all the destination nodes have a positive for all the black hole nodes.

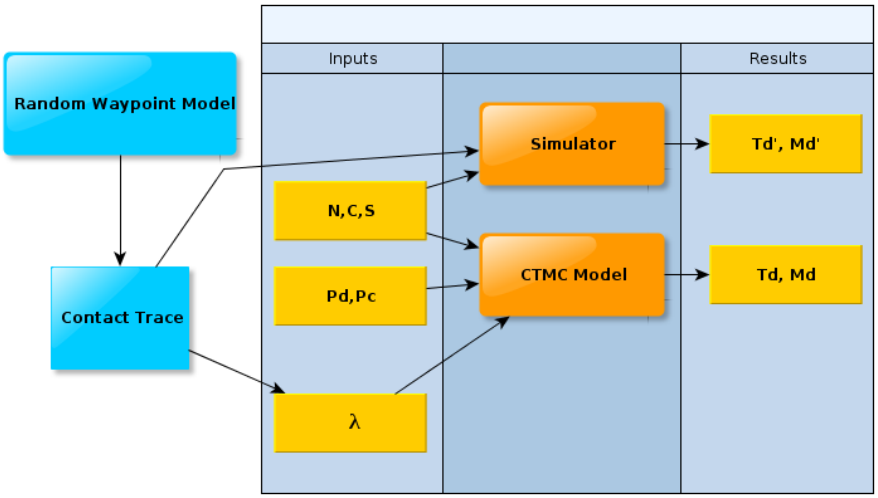


Figure 4.3: Model validation process.

	T_d Error (%)	M_d Error (%)
S=1	0.60 [0.14, 2.5]	1.40 [2.32, 5.2]
S>1	5.09 [2.84, 12.4]	9.31 [3.42, 153]

Table 4.2: Validation results for 100 random tests, presenting mean error and 95% confidence intervals (in brackets)

The model obtains the time and overhead (T_d,M_d) from a set of inputs: the rate of contacts (λ), the network (N ,C,S) and the watchdog parameters (p_d,p_c). The correctness of the model was validated by comparing the results obtained from the model with simulation results. A graphical representation of this process is shown in Figure 4.3. We used a random waypoint model (RWP) generator to create a contact trace, which is used, on the one hand to fit the λ value that is used in our performance model and, on the other hand, to simulate the contacts to obtain the simulation results. The tests have different parameter values that are randomly generated within a pre-defined range. Each simulation was repeated 1000 times in order to obtain a reliable mean value for the detection time and cost ($\bar{T}_d^S, \bar{M}_d^S$,...). For example, for the detection time, the relative error is $\epsilon = \frac{T_d - \bar{T}_d^S}{T_d} \cdot 100$. The validation of the models was based on a set of 100 repeated random tests. For each test, a relative error ϵ_i of the detection time and cost were obtained. The final result of the validation is the mean and the 95% confidence intervals. For example, in the first validation, the values p_d and p_c were randomly distributed between 0.1 and 1, the number of nodes N between 5 and 100, and finally the λ value has a random distribution of $0,1^n$ with n from 1 to 5. In order to evaluate the accuracy of the mean max approximations for S > 1, we performed different test for S = 1 and for S > 1. The results are shown in Table 4.2. We can see that the differences between the models and the simulation results are low. For S = 1 the results are very accurate for all the models. For S > 1 the results show that the model is accurate. The greatest error values take place for higher values of S and N, since the number of mathematical operations is huge, and so the precision is reduced.

4.4 Basic Model evaluation

To evaluate our model it is mandatory to use a known and suitable contact rate. The value of $\lambda = 0.0135$ contacts per hour (that is, $3.71 \times 10^{-6} s^{-1}$), obtained in [ZFX+10], is a very good one for our purposes, because it is based on real motion traces from about 2100 operational taxis in Shanghai city. Using this rate, we studied the influence of the degree of collaboration, the number of nodes and the number of misbehaved nodes in the results obtained using our model.

Figure 4.4 shows how the probability of collaboration p_c affects the detection time and the message overhead for three different values of the probability of detection p_d (0.1, 0.5, and 1.0). It is clear that a small increase in p_c from 0 to 0.2 exponentially decreases the detection time and increases the

message overhead as well, and this behaviour is more visible for lower values of p_d. We must remark that with no collaboration at all ($p_c = 0$), detection time could reach to 3300 hours with $p_d = 0.1$. So, watchdog accuracy (which affect p_d) and collaboration between nodes (with better p_c) both reduce the detection time in this scenario with only one black hole. The best collaboration scenario, correspond to $p_c = 1$, where every node runs a collaborative watchdog, and the detection time is very low. On the other hand, message overhead reaches its maximum, but we must say that it is lower than 7 messages per hour. Finally, we want to remark that increasing the probability of collaboration from 0.4 to 1 has a low impact on the detection time and message overhead.

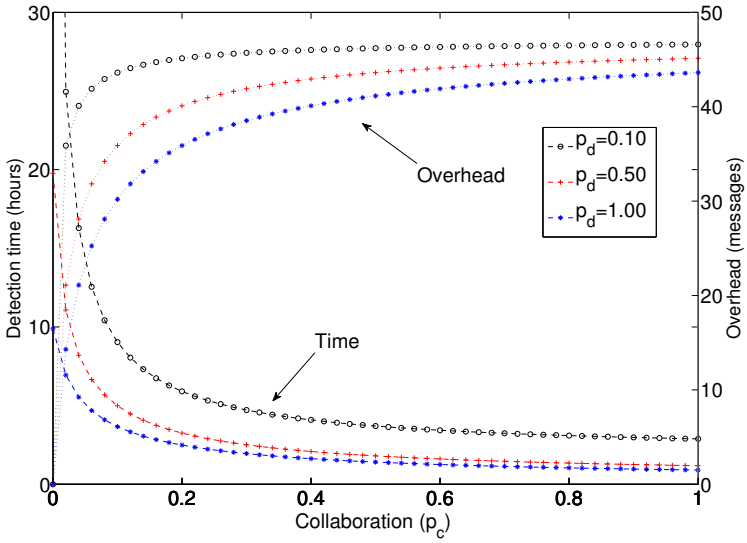


Figure 4.4: Influence of the degree of collaboration, for S=1 and N=50

To analyze the impact of the number of nodes in the network (see Figure 4.5), we have set three fixed pairs of p_c and p_d values. This analysis shows that message overhead increases proportionally to the number of nodes, while detection time reduces exponentially using any of the three pairs of values. As expected, reducing collaboration and/or accuracy increases detection time and message overhead.

To end with this basic model evaluation, we also studied the influence that the number of black hole nodes will have on the results (see Figure 4.6). As expected, the higher their number is, the higher the detection time is.

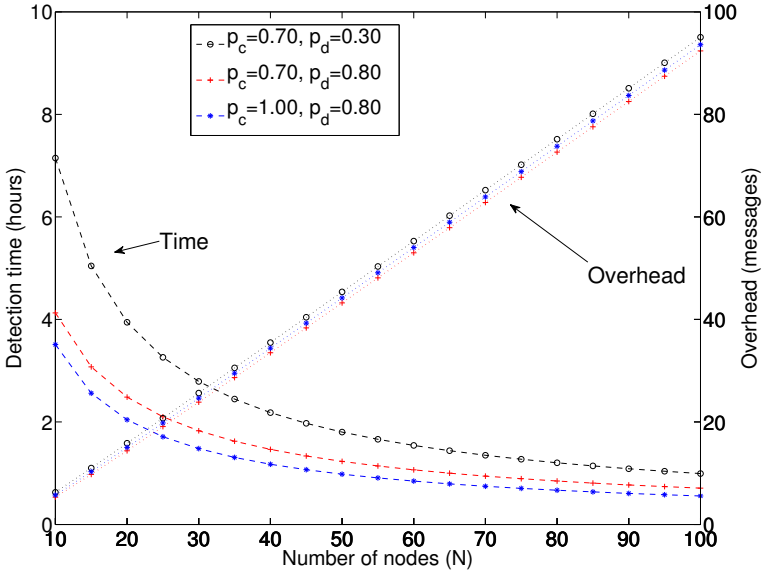


Figure 4.5: Influence of the number of nodes, for S=1

Regarding to the message overhead, there is a exponential increase even for small number of misbehaved nodes, but it begins to decay for S>10, due to the reduced set of collaborative nodes, which can not perform collaborative contacts as often as when the number of misbehaved nodes was lower. So if there are less collaborative contacts, there will be less messages transmissions.

4.5 Summary

In this chapter we have introduced and evaluated a basic model for the perfomance evaluation of collaborative watchdogs for the detection of black holes in MANETs. The aim of this work is reduce the effort needed to evaluate the performance of these tools without simulating or building a real testbed. The core of our model is a Continuous Time Markov chain, parametrized with the degree of collaboration and the accuracy of the watchdog. The evaluated metrics have been detection time and message overhead, and numerical results showed that our collaborative watchdog can reduce the overall detection time with reduced costs in terms of message overhead. This reduction can be obtained even with a moderate degree if collaboration.

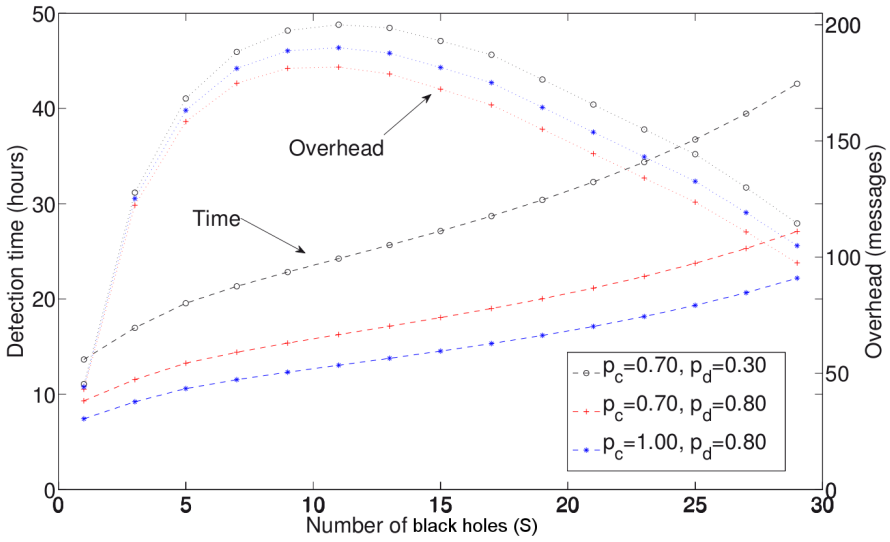


Figure 4.6: Effect of the number of black holes (S).

Our basic model does not deal with false positives or false negatives events. Also it does not work well if we need to evaluate the performance when only certain nodes must have the information about who the black hole nodes are, which could be useful for applying our model to DTNs. In the next chapter, we will propose and evaluate a more sophisticated model including these functionalities.

Chapter 5

Enhancing the Model for the Collaborative Watchdog

As introduced in the previous chapter, there are some immediate improvements which will allow us to better model the Collaborative Bayesian Watchdog. In sections 2.5.4 and 3.3, we paid attention to two undesired results from the watchdog techniques: the appearance of false positives and false negatives, so we must deal with them in our model. Thus, we will introduce in our models the pernicious effect of false positives and false negatives. As in the basic model previously proposed, the network is modeled as a set of N wireless mobile nodes, where C of them are collaborative, D are the destination nodes (that is, the nodes that are going to receive the packet), and S=1 is the black hole node [HOSOC$^+$12a]. Our goal will be to obtain the time and overhead that a set of destination nodes need to detect who the black hole is. The case when S>1 is not modeled, because the number of states increases exponentially with S, so it can be computationally intractable.

5.1 System Model

In this model, every node has a list with the nodes it knows, and the state of each one. Initially, each node has no information about the network. When a contact occur, the black hole detection module of the watchdog can generate the following events:

- ***PosEvt*** (positive event): if the watchdog detects the contacted node as a black hole.
- ***NegEvt*** (negative event): if the watchdog believes that the contacted

node is not a black hole.

- ***NoInfoEvt*** (no info event): if the watchdog has not been able to decide wheter the contacted node is a black hole. This could be caused by a very short contact time or a very small amount of overheared messages.

Due to the generation of this events, a node can update its states table for the other nodes in the MANET according to Figure 5.1. Every entry has an expiration time, so the information about a particular node is deleted after some time without contacting the node.

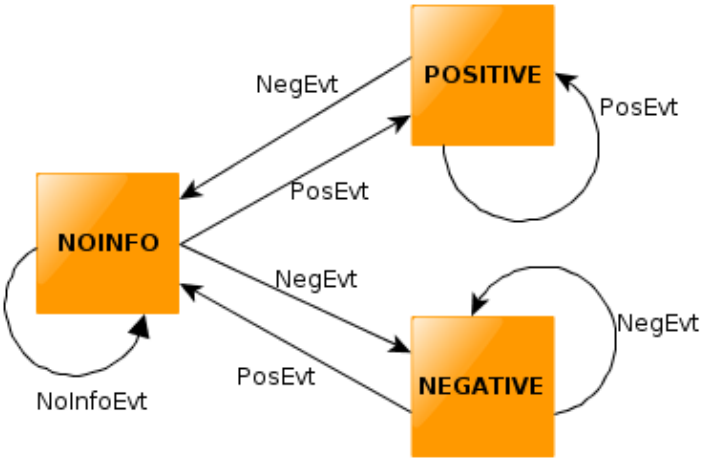


Figure 5.1: State transition diagram when updating information about contacted nodes.

Other major difference between the model we are proposing and the basic model proposed in Chapter 4 is the reputation information diffusion. When a contact between two nodes occurs, there is a transmission and reception of information about known nodes between these contacted nodes. The information about the positives is always transmitted, but information about the negatives is troublesome, because it will produce excessive messaging or fast false negatives diffusion. Thus, we have introduced a **negative diffusion factor** γ, that is the ratio of negatives that a node sends when it contacts another node. The value of γ ranges from 0 (no negatives transmission) to 1 (all the negatives are transmitted). The importance and influence of γ will be detailed in section 5.3.

Now, our watchdog is modelled using four parameters: the probability of detection (p_d), the ratio of false positives (p_{fp}), the ratio of false negatives (p_{fn}) and the probability of collaboration (p_c). The first parameter, p_d, reflects the probability that, when a node contacts another node, its watchdog has enough information to decide whether a node is a black hole or not, that is, to generete a *PosEvt* or a *PosNeg* events. This value depends mainly on the observation time, and the transmission and mobility pattern of the nodes, as we have demonstrated when evaluating the detection speed of the Collaborative Bayesian Watchdog in section 3.3.1. Because the watchdog can generate false positives and false negatives, we have introduced two new parameters which can be expressed as a ratio or probability: p_{fp} is the ratio of false positives generated when a collaborative node contacts other collaborative node, and p_{fn} is the ratio of false negatives generated when a node contacts a black hole node. These values depend on the accuracy of the watchdog (see section 3.3.2). The fourth parameter, p_c, as in the basic model, models the probability of collaboration between two contacted nodes.

Finally, we must remark that this extended model, like the previous one and the simulator, does not support liars which spread false reputation information, thus all the nodes are considered collaborative nodes or black holes.

Using the previous four parameters, we can derive the associated probabilities of the *PosEvt* and *NegEvt* events when a contact occurs [HOSOC+12a]:

- *PosEvt* event:
  - The node contacts a misbehaved node and its watchdog detects it with probability $p_d \cdot (1 - p_{fn})$.
  - The node contacts a collaborative node that has a POSITIVE state about a black hole with probability p_c.
  - A false positive can also be generated in a contact with a collaborative node with probability $p_d \cdot p_{fp}$.
- *NegEvt* event:
  - The node contacts a collaborative node with probability $p_d \cdot (1 - p_{fp})$.
  - The node contacts a collaborative node that has a NEGATIVE state about a collaborative node with probability $\gamma \cdot p_c$.
  - A false negative can also be generated if a contact with a misbehaved node occurs with probability $p_d \cdot p_{fn}$.

Now, we are going to introduce several models that take into account the effect of false positives and false negatives.

5.2 New analytical models

First, we are going to study the impact of false negatives. To ease the exposition, we will initially assume that $D = C$, and later we will extend the model to the generic case when $D \leq C$.

5.2.1 The model for D=C

Using λ we can model the network using a 2D Continuous Time Markov chain (2D-CTMC) with states $(c_p(t), c_n(t))_{t\geq 0}$, where $c_p(t)$ represents the number of collaborative nodes that have a POSITIVE about the black hole at time t, and $c_n(t)$ represents the number of *collaborative* nodes that have a NEGATIVE about the black hole (note that, in this case, is a false negative). At the beginning all nodes have no information. Then, when a contact occurs, $c_p(t)$ and $c_n(t)$ can be increased by one. Note, that c_p and c_n are not independent: $c_p + c_n \leq C$, so some states are not reachable. The absorbing state is achieved when $c_p(t) = C$. This 2D-CTMC model has an initial state $s_1 = (0, 0)$, a final state (C,0) and τ transient states, which are all the possible permutations that sum C or less. In general, $\tau = P^S(C) = 0.5(C+1)(C+2)$. Again, υ is the number of absorbing states ($\upsilon = 1$). This model can be expressed using the transition matrix P in the canonical form as the basic model (see 4.2).

Now, we derive the transition rates p_{ij}. Given the state $s_i = (c_p, c_n)$ the following transitions can occur:

- (c_p, c_n) to $(c_p + 1, c_n)$: A new collaborative node has a POSITIVE. The transition probability is $t_P = \lambda(p_d(1 - p_{fn}) + p_c c_p)(C - c_p - c_n)$. The term $p_d(1 - p_{fn})$ represents the probability of a *PosEvt* from the watchdog and $p_c c_p$ the probability of a *PosEvt* from collaboration. Finally, the factor $(C - c_p - c_n)$ represents the number of pending collaborative nodes. If there are no pending nodes, this value is 0.

- (c_p, c_n) to $(c_p, c_n + 1)$: A new collaborative node has a NEGATIVE (note that it is a *false negative*). The transition probability is $t_N = \lambda(p_d p_{fn} + \gamma p_c c_p)(C - c_p - c_n)$.

- (c_p+1, c_n) to (c_p, c_n): A collaborative node that has a POSITIVE state changes to NOINFO due to a *NegEvt*. So, the transition probability is

$s_i \rightarrow s_j$	0,0	0,1	0,2	1,0	1,1	2,0
0, 0	t_0	t_N	0	t_P	0	0
0, 1	$t_{N'}$	t_0	t_N	0	t_P	0
0, 2	0	$t_{N'}$	t_0	0	0	0
1, 0	$t_{P'}$	0	0	t_0	t_N	t_P
1, 1	0	$t_{P'}$	0	$t_{N'}$	t_0	0
2, 0	0	0	0	0	0	1

Table 5.1: Transition matrix for N=3.

similar to the new negative case: $t_{P'} = \lambda(p_d p_{fn} + \gamma p_c c_n)c_p$.

- $(c_p, c_n + 1)$ to (c_p, c_n): A collaborative node that has a NEGATIVE changes to NOINFO due to a *PosEvt*. The transition probability is similar to the new positive case $t_{N'}\lambda(p_d(1 - p_{fn}) + \gamma p_c c_p)c_n$.

- (c_p, c_n) to (c_p, c_n): This is the probability of no changes, and it is $t_0 = 1 - t_P - t_N - t_{P'} - t_{N'}$.

For example, for N=3, we have C=2, so $\tau = 5$ and $\upsilon = 1$, and the transition matrix is shown in Table 5.1:

At this point, we are able to obtain how long will it take for the 2D-CTMC to be absorbed using the transition matrix $\mathbf{P}$ with these rates p_{ij} and the same expression we used in the basic model (see expression 4.5):

$$T_d = v_1 N v \tag{5.1}$$

where $v_1 = [1, 0, 0, ..., 0]$ and v is a column vector of 1s.

Regarding the overhead, again we need to obtain the number of messages sent in each state s_i, so we need first to know the duration of each state using the fundamental matrix $\mathbf{N}$. By definition, the elements of the first row of $\mathbf{N}$ are the expected durations in each state starting from state s_1, so the duration of state s_i is $f_i = N(1, i)$.

More difficult is to obtain the expected number of messages m_i, because it depends on the diffusion model. Again, to ease the exposition we start with $\gamma = 0$ (only positives are transmitted):

- From state $s_1 = (0, 0)$ to state $s_{C+1} = (0, C)$, no node has a POSITIVE state, so no messages are transmitted and $m_1 = 0$.

- From state $s_{C+2} = (1, 0)$ to state $s_{2C+2} = (1, C - 1)$, only one node has a POSITIVE state that can be transmitted to all the rest of the

collaborative nodes. for the duration of each state i $(N(1,i))$ with a rate λ and probability p_c. So, $m_i = N(1,i)\lambda(C-1)p_c$.

- From state $s_{2C+3} = (2,0)$ to state $s_{3C+2} = (21, C-2)$, there are two nodes which have a POSITIVE state that can be transmitted to all the rest of the collaborative nodes. for the duration of each state i $(N(1,i))$ with a rate λ and probability p_c. So, in this case the value of m_i is $2N(1,i)\lambda(C-1)p_c$.

Summing up, we can conclude that the overhead due to message transmission is:

$$O_d = \lambda\cdot(\text{C-1})\cdot p_c\cdot \sum_{i=1}^{\tau} \Phi(s_i)\cdot\text{N(1,i)} \tag{5.2}$$

where $\Phi(s_i) = c_p$ is the number of nodes with a POSITIVE in state s_i. Now, if $\gamma > 0$, the ratio of nodes that will transmit information about their negatives is exactly γ, so finally $\Phi(s_i) = c_p + \gamma c_n$.

5.2.2 The model for $D \leq C$

Before we can evaluate the new model, we have a pending task, which corresponds to extend the model for the generic case when $D \leq C$. In this generic case, the collaborative nodes set is divided into two separate subsets: a set with D detecting nodes, and a set of M=C-D middle (or non-detecting) nodes. This division is intended to analytically obtain the time and the overhead required for the subset of detecting nodes to be aware of which the black nodes are. For this task we will make use of a Four Dimensional Continuous Time Markov Chain (4D-CTMC) with states $(d_p(t), d_n(t), m_p(t), m_n(t))$, where

- $d_p(t)$ represents the number of detecting nodes with a POSITIVE state at time t.
- $d_n(t)$ represents the number of detecting nodes with a NEGATIVE state at time t.
- $m_p(t)$ represents the number of middle nodes with a POSITIVE state at time t.
- $m_n(t)$ represents the number of middle nodes with a NEGATIVE state at time t.

There are two condition that these states must verify: $d_p(t)+d_n(t) \leq D$ and $m_p(t)+m_n(t) \leq M$. There will be $v = P^S(M)$ absorbing states, which occur when $d_p(t) = D$, and $\tau = (P^S(D)-1) - P^S(M)$ transient states. Again, we can derive the transition rates p_{ij}, given the state $s_i = (d_p, d_n, m_p, m_n)$ as follows:

- (d_p, d_n, m_p, m_n) to (d_p, d_n, m_p+1, m_n): A new middle node has a POSITIVE. The transition probability is $T_{mP} = \lambda(p_d(1-p_{fn}) + p_c(m_p + d_p))(M - m_p - m_n)$.
- (d_p, d_n, m_p, m_n) to (d_p, d_n, m_p, m_n+1): A new middle node has a NEGATIVE (note that it is a *false negative*). The transition probability is $T_{mN} = \lambda(p_d p_{fn} + \gamma p_c(m_n + d_n)(M - m_p - m_n)$.
- $(d_p, d_n, m_p + 1, m_n)$ to (d_p, d_n, m_p, m_n): A middle node that has a POSITIVE state changes to NOINFO. The transition probability is $T_{mP'} = \lambda(p_d p_{fn} + \gamma p_c(m_n + d_n)m_p$.
- $(d_p, d_n, m_p, m_n + 1)$ to (d_p, d_n, m_p, m_n): A middle node that has a NEGATIVE state changes to NOINFO. The transition probability is $T_{mN'} = \lambda(p_d(1 - p_{fn}) + p_c(m_p + d_p)m_n$.
- (d_p, d_n, m_p, m_n) to $(d_p + 1, d_n, m_p, m_n)$: A new detecting node has a POSITIVE. The transition probability is $T_{dP} = \lambda(p_d(1-p_{fn})+p_c(m_p+d_p))(D - d_p - d_n)$.
- (d_p, d_n, m_p, m_n) to $(d_p, d_n + 1, m_p, m_n + 1)$: A new detecting node has a NEGATIVE (note that again it is a *false negative*). The transition probability is $T_{dN} = \lambda(p_d p_{fn} + \gamma p_c(m_n + d_n)(D - d_p - d_n)$.
- $(d_p - 1, d_n, m_p, m_n)$ to (d_p, d_n, m_p, m_n): A detecting node that has a POSITIVE state changes to NOINFO. The transition probability is $T_{dP'} = \lambda(p_d p_{fn} + \gamma p_c(m_n + d_n)d_p$.
- $(d_p, d_n - 1, m_p, m_n)$ to (d_p, d_n, m_p, m_n): A detecting node that has a NEGATIVE state changes to NOINFO. The transition probability is $T_{dN'} = \lambda(p_d(1 - p_{fn}) + p_c(m_p + d_p)d_n$.
- (d_p, d_n, m_p, m_n) to (d_p, d_n, m_p, m_n): This is the probability of no changes (p_{ii}), and it is $T_0 = 1 - \sum_{j \neq i} p_{ij}$.

Using the transition matrix $\mathbf{P}$ with these rates p_{ij} and expression 5.1, we can obtain the expected detection time. Analogously, using expression 5.2, we can obtain the message overhead.

5.2.3 The effect of false positives

When a node has a false positive, the problem is that, due to the diffusion of positives, this false positive can be quickly distributed in the network. A way to evaluate this diffusion is to obtain the time when all nodes have a false positive about a given node. Following the same process that in the false negatives' model, we have a 2D-CMTC with the same states (c_p, c_n), but in this case c_p represents the number of nodes with false positives, and c_n the number of nodes with a negative. The transition rates (p_{ij}) of the transition matrix $\mathbf{P}$ are:

$$p_{ij} = \begin{cases} \lambda(p_d p_{fp}) + f_{cp}(c_p, c_n)(C - c_p - c_n) & (c_p \to c_p + 1) \\ \lambda(p_d(1 - p_{fp}) + f_{cn}(c_p, c_n)(C - c_p - c_n) & (c_n \to c_n + 1) \\ \lambda(p_d(1 - p_{fp}) + f_{cn}(c_p, c_n) \cdot c_p & (c_p \to c_p - 1) \\ \lambda(p_d p_{fn} + f_{cp}(c_p, c_n) \cdot c_n & (c_n \to c_n - 1) \end{cases} \tag{5.3}$$

where $(x \to x{+}1)$ denotes a transition from state (...,x,...) to state (...,x+1,...), and, analogously $(x \to x - 1)$ denotes a transition from state (...,x,...) to state (...,x-1,...). From these expressions, we can observe that the transition rates are the same than in the false negative model in section 5.2 by replacing $p_{fp} = 1 - p_{fn}$. Therefore, we can use the previous models for obtaining the detection time T_d and the overhead O_d, since false positives do not affect the model itself.

5.3 Model evaluation

To finish our work with the analytical model, in this section we will evaluate it. Specifically, we evaluate the effect that false positives and false negatives will have on the performance of the collaborative bayesian watchdog using our extended model. First, we will focus on evaluating the impact of false negatives. Next, the influence of false positives. And at the end, we will compare our contact-based message diffusion approach to a classical periodic diffusion approach. All the experiments have been performed using a λ value of 0.01 contacts/s., which has been shown to be a valid value for vehicular scenarios [ZFX+10].

In this evaluation we will not repeat the study on the influence of the degree of collaboration and the number of nodes in the watchdog performance, because it has been done for the basic model (see section 4.4) for $\gamma = 0$, that is, no transmission of false negatives at all, and it is still valid for the extended model.

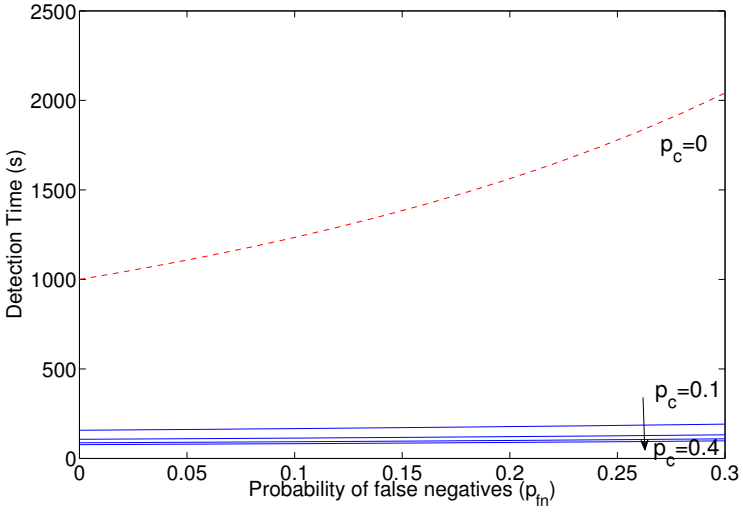


Figure 5.2: Impact of false negatives for $p_d = 0.1$ with $\gamma = 0$ for several values of p_c.

5.3.1 Influence of false negatives

To analyze the influence of the transmission of false negatives, we started with the experiments where there was no false negatives' transmissions ($\gamma = 0$), as in Figure 5.2. In this case, where S=1, D=1 and N=25, it is very clear that the detection time is greatly reduced when p_c is greater than zero. Additionally, Figure 5.2 shows that the probability of false negatives does not affect the detection time. This experiment showed that, as expectect, false negatives had no influence on overhead where $\gamma = 0$, with a value around 20 messages.

With the same parameters, but with $\gamma = 1$, Figure 5.3 shows the effect of full transmission of false negatives. For $p_{fn} = 0$, the results are very similar for the $\gamma = 0$ case, because if there is no possibility of appearance of a false negative it does not matter if the nodes transmit them, because there will not be anything to transmit. However, we can observe that for low degrees of collaboration the detection time decreases and overhead increases in a similar way that in the $\gamma = 0$ case. But when p_c increases, the detection time increases again and the overhead increases exponentially. The conclusion is that it seems that collaboration amplifies the impact of false negatives. To confirm this assumption, Figure 5.4 shows that the curves for greater values of p_c have a greater exponential slope. A similar behaviour has been observed

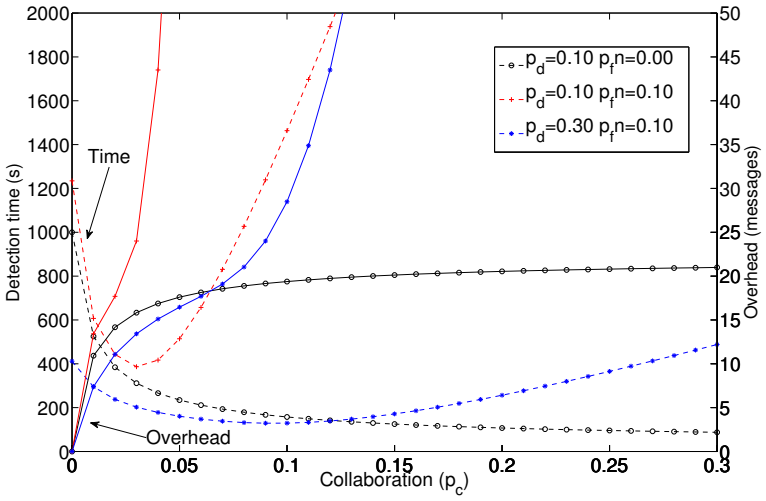


Figure 5.3: Full transmission of negatives $\gamma = 1$: Detection time and overhead depending on collaboration.

for the overhead, because, in general, a greater detection time implies a greater overhead.

Summing up, if only positives are transmitted, the detection time is greatly reduced and the impact of false negatives is also reduced. However, when all known negatives are transmitted, collaboration amplifies the effect of false negatives on the watchdog performance. The results of these experiments in the preliminary stage of the enhancement of our basic model lead us to propose that not all the negatives must be transmitted ($0 \leq \gamma < 1$).

5.3.2 Influence of false positives

To evaluate the influence of false positives in the watchdog, we will now use the model developed in section 5.2.3. We expect that the diffusion of negatives will reduce the influence of false positives. In order to understand the following experiments, we must note that a greater detection time means that the effect of false positives is reduced, because the difussion speed of this false positives is also reduced. If we are not wrong, when $\gamma = 0$ the influence of false positives on the watchdog performance will increase when compared to scenarios where $\gamma > 0$.

In Figure 5.5a, we can observe that the detection time experiments a drastic reduction due to the fast spreading of false positives, when $\gamma = 0$.

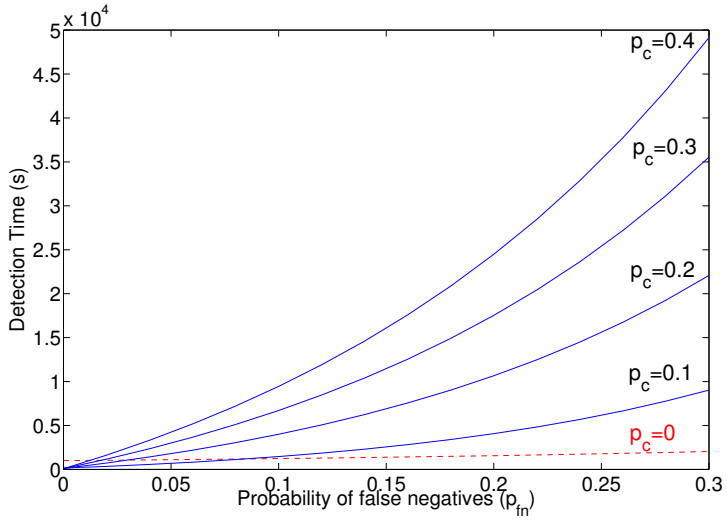


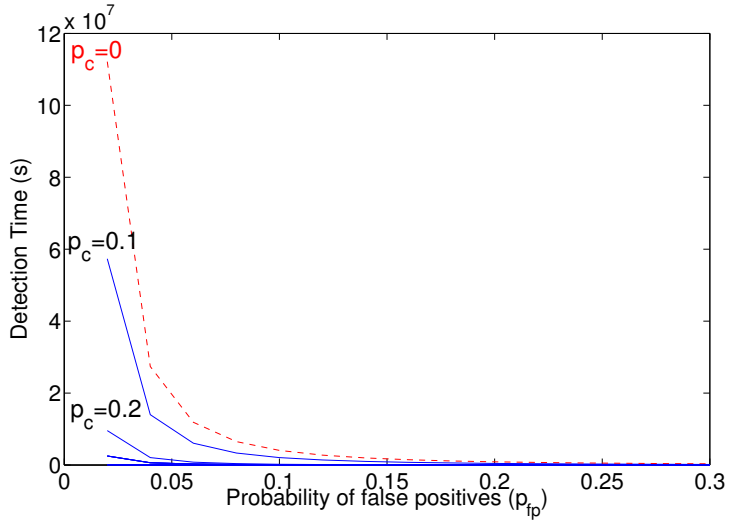
Figure 5.4: Impact of false negatives for $\gamma = 1$ for several values of p_c.

This behaviour is clear for those curves when $p_c > 0$. This means that exists an undesired effect which increases the false positives rate (p_{fp}), so we must also transmit the negatives to reduce this effect.

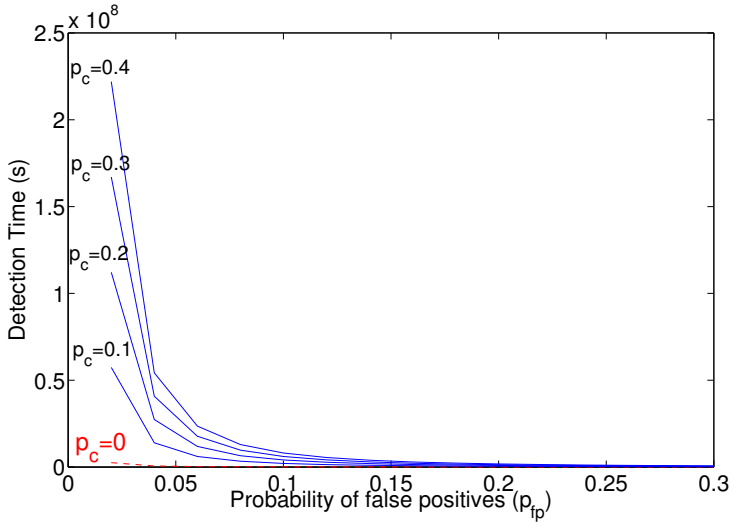
On the other hand, Figure 5.5b shows the results for $\gamma = 1$. In this case, the detection time is highly increased when the collaboration increases, thus reducing the effect of false positives. The main conclusion of this analysis is that we have the inverse case that in the false negatives case. If only POSITIVES are transmitted, the effect of false positives is magnified, so we need to transmit the NEGATIVES to reduce their impact, modulating it with the γ factor.

Finally, we evaluated the same scenario with $\gamma = 0.25$, and presented the results in Figures 5.6 and 5.7. Figure 5.6 shows that the detection time is reduced even if the ratio of false negatives is high. Finally, Figure 5.7 depicts how the detection time increases when the collaboration increases, which effectively reduces the effect of false positives.

The main conclusion is that the γ parameter must be tuned up to achieve the desired behaviour. A γ values near to zero reduces the detection time, but increases the diffusion of false positives. On the other hand, a values of γ near to one increases the detection time (due to the effect of false negatives) while reducing the difussion of false positives.



(a) Impact of false positives when $\gamma = 0$ for several values of p_c.



(b) Impact of false positives for $\gamma = 1$ for several values of p_c.

Figure 5.5: Impact of false positives for several values of p_c.

5.3.3 Contact-based difussion vs. other approaches

Finally, we compare our contact-based difussion scheme with a classical periodic message diffusion, like the one used in [BLB05, PW02]. We only

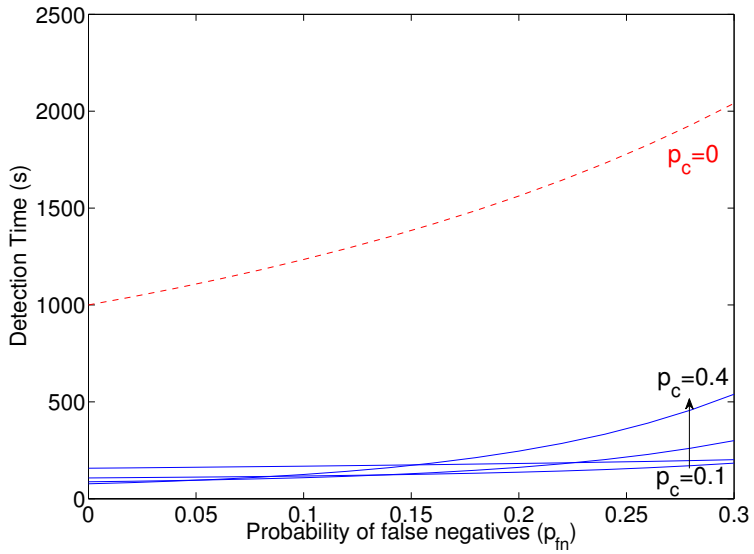


Figure 5.6: Results for a controlled diffusion of false negatives ($\gamma = 0.25$): impact of false negatives.

compare the diffusion protocol, where when a node has information to share it spreads it with a given period P. This reputation message will be received by all the nodes present in the originator's neighborhood. The performance of this protocol clearly depends on the particular period P, because a short period reduces the detection time but increases the overhead. A bigger period P will increase the detection time while reducing the overhead.

The comparison between the two approaches, i.e. our contact-based diffusion model and the periodic diffusion model, has been done through simulation, using *ns-2*, with mobility scenarios generated by *setdest*. We must note that in the periodic approach only positives are sent. The parameters we used in simulation are detailed in Table 5.2.

Figure 5.8 shows the results for the simulations, with period P ranging from 1 to 30 seconds, in scenarios with 30 to 50 nodes. These results confirm that increasing P implies a higher detection time and a smaller overhead. Comparing these results with the ones obtained by our Collaborative Bayesian Watchdog, included in Figure 5.8, the periodic diffusion scheme has a lower detection time for periods below 4s, but with a higher overhead. If P=2s, the detection time for the periodic approach is 963s (a 9% less than the contact-based one), while the overhead is 5212 messages (an increase of

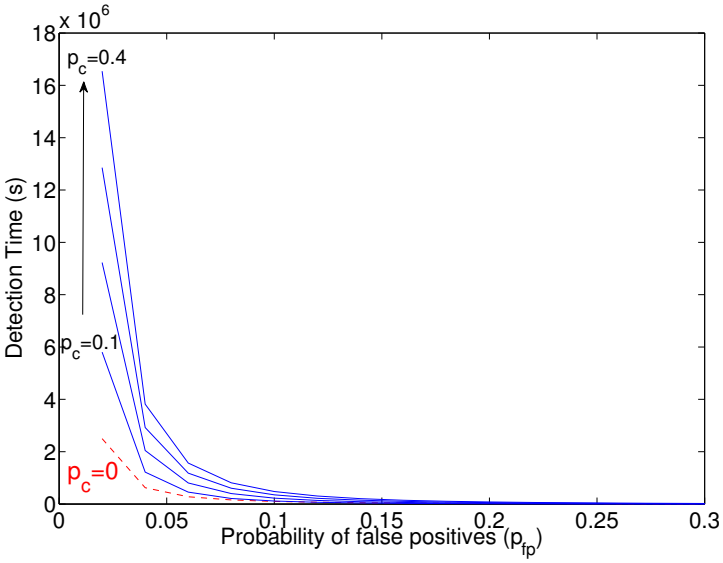


Figure 5.7: Results for a controlled diffusion of false negatives ($\gamma = 0.25$): impact of false positives.

Parameter	Value
Nodes	30, 40 or 50
p_{fp}	0.17
p_{fn}	0.08
p_d	0.11
p_c	0.2
γ	0.25

Table 5.2: Simulation parameters to compare contact-based and periodic diffusions.

4378% over the contact-based protocol). For similar detection times between the two approaches (when P=4s), the periodic difussion approach shows an overhead increment of 2972% compared to our contact-based approach.

Finally, regarding to the difussion time of false positives, although it is reduced, the result show that it presents a false positive rate of 0.72, which is unacceptable. This leads to the conclusion that using periodic diffusion slightly reduces the detection time at the cost of highly increasing the overhead, while it also exhibits a high impact of false positives on the

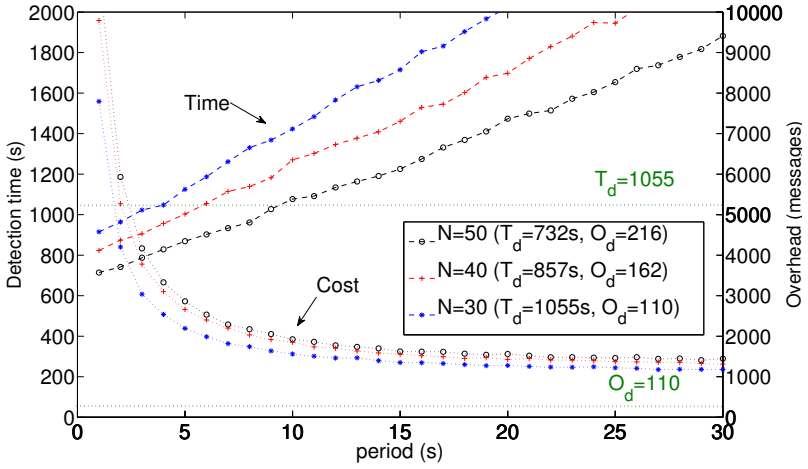


Figure 5.8: Detection time and overhead for the periodic approach when varying period P.

performance of this strategy, which leads us to consider it as a non viable strategy.

5.4 Summary

In this chapter we have presented an enhanced version of the model presented in Chapter 4. These enhancements consist on including into the model three interesting aspects: modelling false positives, modelling false negatives, and allowing that the detecting nodes set to be smaller than the collaborative nodes set. This model spreads information about all the POSITIVES, but only a fraction of the information about the NEGATIVES, indicated by the γ factor, when a contact between collaborative nodes occurs. We have evaluated our extended model to conclude that the utilization of γ reduces the harmful impact of false negatives and also the impact of false positives. This means that the effect of controlled collaboration proposed in our approach can reduce the detection time, while increasing the detection accuracy at a moderated message cost. This statement has also been validated by comparing our contact-based approach to a periodic diffusion approach, showing that our proposal offers better results.

Chapter 6

Conclusions, Publications and Future Work

To finish this work, in this chapter we introduce our conclusions and outcomes, the detailed list of publications generated during the reseach period, and an outline of the tasks that could follow this thesis.

6.1 Conclusions

Throughout this thesis, two main contributions have been made: a security technique, classified as an Intrusion Detection System, called *Collaborative Bayesian Watchdog*, and an analytical model to evalute its performance in different scenarios.

The first contibution has been the implementation of our Collaborative Bayesian Watchdog, a security mechanism which allows to cooperatively detect black hole attacks in Mobile Ad hoc Networks. In addition, we have evaluated it, comparing the results with those obtained from other non-collaborative watchdog versions. To easily compare the outcomes of the different approaches, we have set two main metrics: detection speed, and accuracy. The results showed that our approach improves previous non-cooperative versions at an affordable cost in terms of computational complexity, and message overhead. In terms of detection speed, in average, our proposal detects earlier the black hole in 7% of times compared to previous proposals. In terms of accuracy, in average, our approach reduces the amount of false negatives by 1.17%. That is, our watchdog detects black holes which are not detected by other approaches. This technique has produced good results, but it takes a big amount of time and effort to obtain statistically

significative results in large scenarios.

The second main contribution we made has been the proposal of an analytical model to evaluate the performance of collaborative watchdogs. With these models, we dramatically reduced the time needed to evaluate this technique in different network scenarios. Initially, we presented a basic model, where we only take into account the probabilities of detection and collaboration. We obtained the detection time needed for all the collaborative nodes to know who are the black holes present in the MANET, and how many messages they have sent to achieve this state.

Once this model has been validated, our next step has been to improve it to take into account the presence of false positives and false negatives in the detections.

We have demonstrated that these models are suitable not only for MANETs, but also for DTN systems. At this point, our models deal with the probability of detection, the probability of collaboration, the effect of false negatives, and the effect of false positives, thus allowing us to extensively study the impact of every modelled parameter.

The evaluations done with this model show that if only information about the positive detections are transmitted, the detection time is greatly reduced and the impact of false negatives is also reduced. However, when all known negative detections are transmitted, collaboration amplifies the effect of false negatives on the watchdog performance. Thus, a controlled diffusion of information, as the one proposed in this thesis, can reduce the detection time, while increasing the detection accuracy at a moderated message cost, reducing the harmful impact of false negatives and also the impact of false positives.

Overall, we explored and evaluated cooperation techniques that lead to enhance the results of non-cooperating security techniques, like the watchdog, at an affordable cost. Throughout this work, with the Collaborative Watchdog, we have demonstrated that using the adequate cooperative technique could be suitable, in terms of efficiency, to solve certain problems related to misbehaving nodes in MANETs and DTNs. Additionally, our model has allowed us to quickly study and demonstrate that contact-based diffusion strategy, like the one implemented in our Collaboration Bayesian Watchdog, obtains better results than the classical periodic diffusion strategy proposed in some previous approaches. Related to this subject, we can say that our studies show that for similar detection times, periodic diffusion scheme has a 2972% more of message overhead than our proposed contact-based diffusion scheme.

6.2 Publications Related with this Thesis

The research work related to this thesis has resulted in nine publications; among them we have three journal articles listed in the Journal Citation Report, one book chapter, four international conference papers, some of them indexed by the Computer Science Conference Ranking or the Computing Research and Education (CORE), and one paper in national conferences. We now proceed by presenting the publications list.

6.2.1 Journals

- Hernandez-Orallo, E.; Serrat-Olmos, M.D.; Cano, J.; Calafate, C.T.; Manzoni, P.; "Improving Selfish Node Detection in MANETs Using a Collaborative Watchdog" IEEE Communications Letters , vol.16, no.5, pp.642-645, May 2012. JCR Impact Factor: 1.060.

- Serrat-Olmos, M.D.; Hernández-Orallo, E.; Cano, J.; Calafate, C.T.; Manzoni,P.; "A Novel Approach for the Fast Detection of Black Holes in MANETs" SAGE Concurrent Engineering Research and Applications journal (accepted, 2013). JCR Impact Factor: 0.478.

- Hernandez-Orallo, E.; Serrat-Olmos, M.D.; Cano, J.; Calafate, C.T.; Manzoni, P.; "A Fast Model for Evaluating the Detection of Selfish Nodes Using a Collaborative Approach in MANETs" Wireless Personal Communications journal (accepted, 2013). JCR Impact Factor 2011: 0.503.

6.2.2 Book Chapter

- Serrat-Olmos, M.D.; Hernández-Orallo, E.; Cano, J.; Calafate, C.T.; Manzoni,P.; "Fighting against black hole attacks in Mobile Ad Hoc Networks" Book chapter (accepted, 2013), in the book "Security for Multihop Wireless Networks", to be published by Auerbach Publications, Taylor & Francis Group, USA .

6.2.3 International Conferences

- Hernandez-Orallo, E.; Serrat-Olmos, M.D.; Cano, J.; Calafate, C.T.; Manzoni, P.;"Collaborative watchdogs: A fast and efficient approach to deal with selfish nodes in MANETs," Fourth International Conference on Ubiquitous and Future Networks (ICUFN), 2012, pages 68-73, 4-6 July 2012, Phuket (Thailand).

- Serrat-Olmos, M.D.; Hernández-Orallo, E.; Cano, J.; Calafate, C.T.; Manzoni,P.; "A Collaborative Bayesian Watchdog for Detecting Black Holes in MANETs" Proceedings of the 6th International Symposium on Intelligent Distributed Computing (IDC), 2012. ISBN 978-3-642-32523-6, pages 221-230 September 2012 Calabria (Italy).

- Hernandez-Orallo, E.; Serrat-Olmos, M.D.; Cano, J.; Calafate, C.T.; Manzoni, P.; "Evaluation of collaborative selfish node detection in MANETS and DTNs". Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '12), 2012, pages 159-166. October 2012, Paphos (Cyprus).

- Serrat-Olmos, M.D.; Hernández-Orallo, E.; Cano, J.; Calafate, C.T.; Manzoni,P.; "Accurate Detection of Black Holes in MANETs using Collaborative Bayesian Watchdogs " Wireless Days (WD), 2012 IFIP , vol., no., pp.1-6, 21-23, November 2012, Dublin (Ireland).

6.2.4 National Conferences

- Serrat-Olmos, M.D.; Hernández-Orallo, E.; Cano, J.; Calafate, C.T.; Manzoni,P.; "Collaborative Watchdog to Improve the Detection Speed of Black Holes in MANETs", Actas de las XXIII Jornadas de Paralelismo (JS 2012), Elche (Spain), September 2012

6.3 Future Work

In the development of this thesis several issues emerged which deserve further scrutiny in the future. The ones we consider more relevant are the following:

- Is it feasible to enhance the collaborative bayesian watchdog we have implemented on the simulator to detect black hole attacks even better and quicker? That is, we have to analyze if we can obtain better results with different watchdog parametrizations or with different statistical functions over the collected data, thus enhancing the bayesian detector. Also, there are open issues over the message overhead optimization.

- Which weaknesses and related attacks could be addressed with enhanced versions of our collaborative bayesian watchdog? In section 3.5 we cited some related attacks which our model does not properly address, so this area is conducive for watchdog enhancements.

- Small aspects of the implemented watchdog are not already included in the models proposed, like the δ parameter (see section 3.2), to allow the model to weight the trust on the cooperating nodes. Also, in the early stages of the model development we had to focus on simple network models, and some simplifications had been done to start our work. For example, our model currently does not support the obvervations devaluation, which allow a node to be reinserted into the well-behaved node set if it starts to behave well after been previously detected as a black hole. Thus, these model enhancement are pending.

- Could our analytical model evolve to include all the new functionalities? Maybe some improvements in the watchdog performance, like those related to better parametrizations, could not have a counterpart in the analytical model, but we have to work on adapting the model to the future watchdog implementations.

- Is it feasible to implement this techniques over a real testbed? Another pending work consists on implementing the collaborative bayesian watchdog in a real hardware environment, like our MANET testbed "Castadiva" [HNC+07]. It would be very interesting to compare the results obtained by using real hardware with those obtained in simulation, and so, with those obtained with the analytical model.

Bibliography

[BH00] L. Buttyan and J.-P. Hubaux. Enforcing service availability in mobile ad-hoc wans. In *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC'2000)*, 2000.

[BH03] L. Buttyan and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8(5), October 2003.

[BK09] Osamah S. Badarneh and Michel Kadoch. Multicast routing protocols in mobile ad hoc networks: A comparative survey and taxonomy. *EURASIP Journal on Wireless Communications and Networking*, 2009.

[BLB05] S. Buchegger and J.-Y. Le Boudec. Self-policing mobile ad hoc networks by reputation systems. *IEEE Communications Magazine*, 43, Issue: 7:101 – 107, July 2005.

[Bou08] Azzedine Boukerche, editor. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks.* Wiley-IEEE Press, November 2008.

[Bur02] Peter Burkholder. Ssl man-in-the-middle attacks, February 2002.

[CE09] Han Cai and Do Young Eun. Crossing over the bounded domain: From exponential to power-law intermeeting time in mobile ad hoc networks. *Networking, IEEE/ACM Transactions on*, 17(5):1578 –1591, oct. 2009.

[Cha03] Mike Chapple. *The GSEC Prep Guide: Mastering SANS GIAC Security Essentials.* Wiley ISBN 978-0764539329, 2003.

[CHC+07] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6:606–620, June 2007.

[CJ03] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr), 2003.

[CP10] I. Chakeres and C. Perkins. Dynamic manet on-demand (dymo) routing, 2010.

[CPD03] E. Belding-Royer C. Perkins and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[DJH07] D. Maltz D. Johnson and Y-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr), 2007.

[Dou02] J. R. Douceur. The sybil attack. *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002.

[Fee99] Laura Marie Feeney. A taxonomy for routing protocols in mobile ad hoc networks, 1999.

[Gio02] S. Giordano. *Handbook of Wireless Networks and Mobile Computing.* , John Wiley & Sons, Inc., New York, USA, 2002.

[GLZC09] Wei Gao, Qinghua Li, Bo Zhao, and Guohong Cao. Multicasting in delay tolerant networks: a social network perspective. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 299–308, New York, NY, USA, 2009. ACM.

[GNK05] Robin Groenevelt, Philippe Nain, and Ger Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62:210–228, October 2005.

[GS97] Charles M. Grinstead and J. Laurie Snell. *Introduction to probability.* 1997.

[GVKG09] M. Ghosh, A. Varghese, A.-A. Kherani, and A. Gupta. Distributed misbehavior detection in vanets. In *Proceeding of the IEEE Wireless Communications and Networking Conference*, 2009.

[HBT+03] Julian Hsu, Sameer Bhatia, Mineo Takai, Rajive L. Bagrodia, and Michael J. Acriche. Performance of mobile ad hoc networking routing protocols in realistic scenarios. In *IEEE Military Communications Conference, 2003. MILCOM '03*, volume 2, pages 1268 – 1273, 2003.

[HCC+10] J. Hortelano, C.-T. Calafate, J.-C. Cano, M. de Leoni, P. Manzoni, and M. Mecella. Black-hole attacks in p2p mobile networks discovered through bayesian filters. In *Proceedings of OTM Workshops'2010*, pages 543–552, 2010.

[HCCM10] J. Hortelano, J.-C. Cano, C.-T. Calafate, and P. Manzoni. Watchdog intrusion detection systems: Are they feasible in manets? In *XXI Jornadas de Paralelismo (CEDI'2010)*, 2010.

[HNC+07] J. Hortelano, M. Nacher, J.-C. Cano, C. Calafate, and P. Manzoni. Castadiva: A test-bed architecture for mobile ad hoc networks. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, 2007.

[HOSOC+12a] E. Hernandez-Orallo, M.D. Serrat-Olmos, J.-C. Cano, C. Calafate, and P. Manzoni. Evaluation of collaborative selfish node detection in manets and dtns. In *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2012.

[HOSOC+12b] E. Hernandez-Orallo, M.D. Serrat-Olmos, J.-C. Cano, C. Calafate, and P. Manzoni. Improving selfish node detection in manets using a collaborative watchdog. *IEEE Communications Letters*, 16 , Issue: 5:642 – 645, 2012.

[HWKch] Qi He, Dapeng Wu, and Pradeep Khosla. Sori: a secure and objective reputation-based incentive scheme for ad-hoc networks. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 2, pages 825–830 Vol.2, March.

[IET] Mobile ad-hoc networks ietf working group.

[KKSW04] F. Kargl, A. Klenk, S. Schlot, and M. Webber. Advanced detection of selfish or malicious nodes in ad hoc networks. In

Proceedings of the First European Conference on Security in Ad-Hoc and Sensor Networks (ESAS 2004), 2004.

[KLBV07] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 183–194, New York, NY, USA, 2007. ACM.

[KN12] H. Kandavalli and M.V.S.S NagendraNath. Minimizing malicious eavesdropping ability in wireless mesh networks using skems. *International Journal of Computer Science and Information Technologies*, 3 Issue 2:3476–3478, 2012.

[LSW+11] Yong Li, Guolong Su, D.O. Wu, Depeng Jin, Li Su, and Lieguang Zeng. The impact of node selfishness on multicasting in delay tolerant networks. *Vehicular Technology, IEEE Transactions on*, 60(5):2224 –2238, jun 2011.

[MGLB00] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom'00)*, 2000.

[Mis09] Isaac; Misra Subhas Chandra Misra, Sudip; Woungang, editor. *Guide to Wireless Ad Hoc Networks*. Computer Communications and Networks. Springer-Verlag, 2009.

[MLB08] Jochen Mundinger and Jean-Yves Le Boudec. Analysis of a reputation system for mobile ad-hoc networks with liars. *Perform. Eval.*, 65(3-4):212–226, March 2008.

[MM02] Pietro Michiardi and Refik Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*, pages 107–121, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V.

[NIS95] An introduction to computer security: The nist handbook (special publication 800-12) national institute of standards and technology - u.s. department of commerce, October 1995.

[O'L92] Daniel O'Leary. Intrusion-detection systems. *Journal of Information Systems*, pages 63–74, Spring 1992.

[PP05] B. Parno and A. Perrig. Challenges in securing vehicular networks. In *Proceedings of the Workshop on Hot Topics in Networks*, 2005.

[PW02] K. Paul and D. Westhoff. Context aware detection of selfish nodes in dsr based ad-hoc networks. In *In Proceedings of IEEE Globecom*, 2002.

[RH07] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security - Special Issue on Security of Ad-hoc and Sensor Networks*, 15 Issue 1:39–68, 2007.

[SCDR04] Atul Singh, Miguel Castro, Peter Druschel, and Antony Rowstron. Defending against eclipse attacks on overlay networks. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, EW 11, New York, NY, USA, 2004. ACM.

[SO11] Manuel David Serrat-Olmos. Watchdogs colaborativos para la deteccion de nodos maliciosos en redes manet. Master's thesis, Universitat Politecnica de Valencia, 2011.

[SOHOC+12] M.D. Serrat-Olmos, E. Hernandez-Orallo, J.-C. Cano, C. Calafate, and P. Manzoni. A collaborative bayesian watchdog for detecting black holes in manets. In *Proceedings of the 6th International Symposium on Intelligent Distributed Computing*, 2012.

[SS10] T. Sundarajan and A. Shammugam. Modeling the behavior of selfish forwarding nodes to stimulate cooperation in manet. *International Journal of Network Security and its Applications (IJNSA)*, 2(2), April 2010.

[TKOY10] C.-K. Toh, D. Kim, S. Oh, and H. Yoo. The controversy of selfish nodes in ad hoc networks. In *Proceedings of the Twelveth international conference on Advanced communication technology (ICACT'10)*, 2010.

[TS08] Latha Tamilselvan and V. Sankaranarayanan. Prevention of co-operative black hole attack in manet. *Journal of Networks*, 3(5), 2008.

[VH02] Antonio Villalon Huerta. Seguridad en unix y redes, July 2002.

[Wal96] Christian Walck. Hand-book on statistical distributions for experimentalists, December 1996.

[WMHil] Fei Wang, Yijun Mo, and Benxiong Huang. Defending reputation system against false recommendation in mobile ad hoc network. In *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pages 488–493, April.

[YHM] Po-Wah Yau, Shenglan Hu, and Chris J. Mitchell. Abstract malicious attacks on ad hoc network routing protocols.

[YMF06] S. Yousefi, M. S. Mousavi, and M. Fathy. Vehicular ad hoc networks (vanets): Challenges and perspectives. In *Proceedings of the 6th International Conference on ITS Telecommunications*, 2006.

[ZCY03] S. Zhong, J. Chen, and Y.R. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of the Twenty-second Annual Joint Conference of the IEEE Computer And Communications Societies (INFOCOM'03)*, 2003.

[ZFX+10] Hongzi Zhu, Luoyi Fu, Guangtao Xue, Yanmin Zhu, Minglu Li, and Lionel M. Ni. Recognizing exponential inter-contact time in vanets. In *Proceedings of the 29th conference on Information communications (INFOCOM'10), San Diego, CA,USA, March 15-19*, INFOCOM'10, pages 101–105, Piscataway, NJ, USA, 2010. IEEE Press.