# SESSIONS-BASED MISBEHAVIOUR DETECTION FRAMEWORK FOR WIRELESS MOBILE AD HOC NETWORKS

*By*

TARAG FAHAD

A THESIS

**Submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy**

School of Computing and Mathematical Sciences

Liverpool John Moores University, U.K.

November 2007


بسم الله الرحمن الرحيم

" وَعَلَّمَكَ مَا لَمْ تَكُنْ تَعْلَمُ وَكَانَ فَضْلُ اللّهِ عَلَيْكَ عَظِيمًا"

*اية 113 من سورة النساء*

"Allah taught you what you did not know before, and Allah's grace on you is very great"

*Verse 113, Chapter 3 of Quran.*

# DETECATION

- To my beloved parents
- my brothers and sisters
- my wife
- my new born daughter Shaima

# ACKNOWLEDGEMENTS

I would like to express my deep appreciation and greatest thanks to the following persons:

- My principal supervisor Dr. Robert Askwith for his continued support, guidance and encouragement throughout the period of this research.

- My second supervisor Prof. Madjid Merabti for the advice, help and support he provided.

- Prof. George Pavlou and Dr. Michael Howarth, from Surrey University for their support during my first year of this research.

- My colleagues both at Liverpool John Moores University and around the globe, especially Dr. Djamel Djanouri of CERIST Algaria for his help and co-operation.

- My family who offered me unconditional love and support throughout the course of my studies.

I also acknowledge that the funding for this PhD project was provided by the Libyan General People's Committee for Higher Education to whom I am most grateful.

# ABSTRACT

There has been a tremendous growth over the past decade in the use of wireless communication. As the cost of wireless access drops, wireless communications could replace wired in many settings. Today, widely travelling laptop users access the Internet at a variety of places including their homes, and even at public places such as airports. Mobile Wireless Ad hoc Networks (MANET) is one such type of wireless network that have many useful applications including Wireless Sensors Networks which is now used in many civilian and environmental application areas.

In mobile ad-hoc networks, nodes act as both routers and terminals. For the lack of routing infrastructure, they have to cooperate to communicate. Misbehaviour means deviation from regular routing and forwarding. It occurs by either selfish or malicious nodes. In both types misbehaviour's impact on MANET's proves to be detrimental, decreasing the performance and the fairness of the network, and in the extreme case resulting in a non-functional network. In this thesis we have addressed the requirements that nodes misbehaviour detection solution in MANET's should achieve. Existing solutions related to nodes misbehaviour detection in MANET were shown to fail to meet all of our requirements.

The main direction of our work has been to look for an effective approach that can satisfy our requirements. The result is a new novel low cost framework entitled Sessions-based Misbehaviour Detection Framework (SMDF). It consists of three components, the detection component, the decision component and the isolation component. We analysed and evaluated the proposed schemes by simulation techniques. By comparing our results to those of other mechanisms available in the literature, we showed that our solution has low cost in terms of communication overhead and has the lowest False Positive as well as the highest value of True Positive Detection Rates. It also showed that our solution has lower energy consumption rate and is scalable. Finally we present a series of proposals for future research work that have been raised by this work, such as tackling detection complications in hybrid ad hoc network environments.

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ACK | Acknowledgement |
| AODV | Ad hoc On-demand Distance Vector |
| AP | Access Point |
| BS | Base Station |
| CBR | Constant Bit Rate |
| CH | Cluster Head |
| COL | Collision |
| DS | Data Sending |
| DSR | Dynamic Source Routing |
| FAP | Forward Approval Packet |
| FTP | File Transfer Protocol |
| GloMoSim | Global Mobile Information Systems Simulation |
| GPS | Global Positioning System |
| GPSR | Greedy Perimeter Stateless Routing |
| GUI | Graphical User Interface |
| IETF | Internet Engineering Task Force |
| IP | Internet Protocol |
| ISP | Internet Service Provider |
| MAC | Medium Access Control |
| MANET | Mobile Ad hoc NETworks |
| MH | Mobile Host |
| MN | Mobile Node |
| MTS | Mobile Telephony System |
| nc | Neighbour Count |
| NPS | Number of Packets Sent |
| NPR | Number of Packet Received |
| OLSR | Optimised Link State Routing |
| PAN | Personal Area Network |
| QoS | Quality of Service |
| RF | Radio Frequency |

| | |
|---|---|
| RN | Regular Node |
| RREP | Route REPly |
| RREQ | Route REQuest |
| RTS | Request-To-Send |
| RTS | Ready To Send |
| R2HACK | Random Two Hop ACKnowledgment |
| SC | Sample Count |
| SMDF | Sessions-based Misbehaviour Detection Framework |
| SMDP | Sessions-based Misbehaviour Detection Protocol |
| TCP | Transmission Control Protocol |
| TDD | Time Division Duplex |
| TDMA | Time-Division Multiple Access |
| TE | Traffic Engineering |
| TO | Traffic Optimisation |
| UDP | User Datagram Protocol |
| WLAN | Wireless Local Area Network |
| WSP | Widest Shortest Path |
| WTP | Waiting Time Priority |
| WD | Watchdog |
| WSN | Wireless Sensor Network |

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

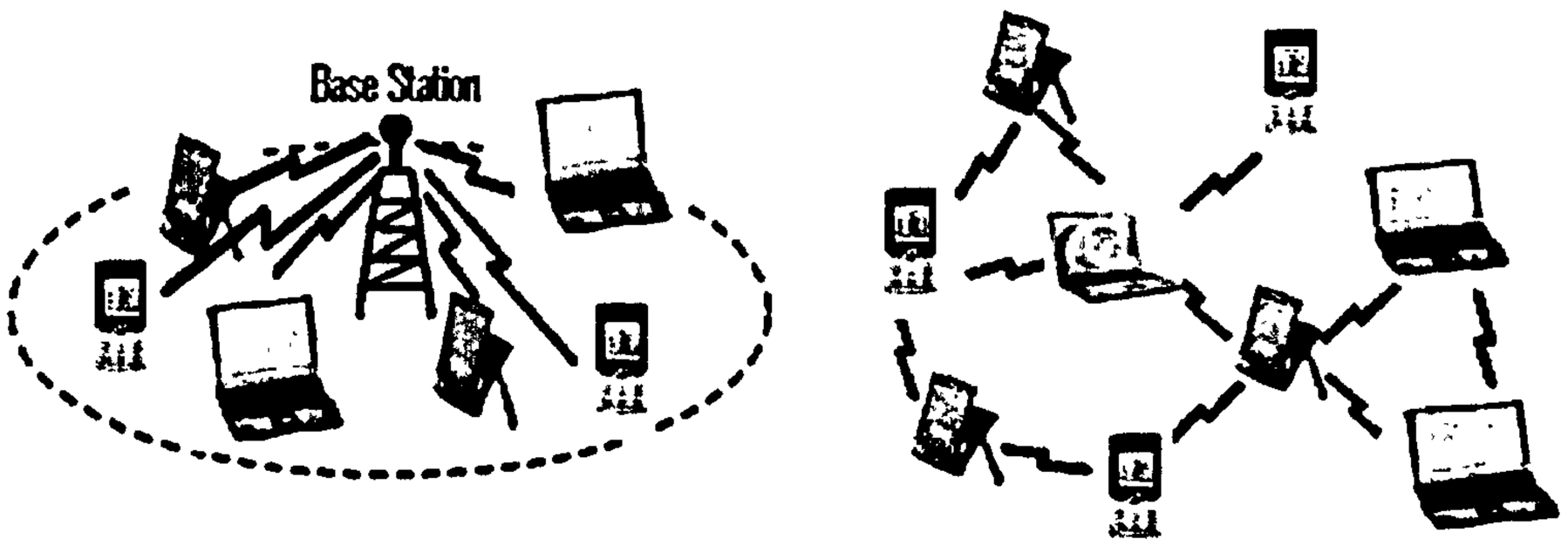
# 1. INTRODUCTION

Computer networks have come a long way since their initiation, and wireless networks are the new trend in the IT market. Wireless networks have become increasingly popular in the past few decades, particularly within the 1990's when they were adapted to enable mobility and wireless devices became popular. Wireless communication brings fundamental changes to data networking and telecommunications. Air is used as the transmission medium, which allows a great flexibility. This way, networks can easily and rapidly be deployed in environments where cabling is difficult. Low prices and good performance incite more and more companies and home users to choose those new kinds of networks. As the cost of wireless access drops, wireless communications could replace wired in many settings. Today, widely travelling laptop users access the Internet at a variety of places and environments including their homes, corporate offices, and even at public places such as conference venues, airports, shopping malls, hotels, libraries, arenas, and so on. One advantage of wireless is the ability to transmit data among users in a common area while remaining mobile. However, the distance between participants is limited by the range of transmitters or their proximity to wireless access points. Mobile Ad hoc wireless networks (MANET) solve this problem by allowing out of range nodes to route data through intermediate nodes.

Interest in commercialization of MANET is recently growing at a much faster rate due to their portability and proliferation of mobile communication devices like laptops, PDAs, mobile phones, and other intelligent radio devices [Akyildiz'05]. Unlike the typical Internet, which has dedicated nodes for basic network operations such as authorization, routing, packet forwarding, and network management, all these functions should be performed by the nodes themselves in MANET. However,

typical nodes cannot be trusted with these important network functions. Thus, security has become an essential consideration in MANET, especially in open MANET, where a variety of mobile nodes supplied by different manufacturers compose a MANET in a self-organizing manner and share their resources for global connectivity with their own goals. Moreover, MANET nodes still have small storage, low bandwidth, high error rates, and limited battery power, in spite of recent appreciable advances in terms of power efficiency, flexibility, and robustness. For these reasons, common security algorithms designed for traditional networks are difficult to use in MANET. Hence, new security approaches need to be developed for MANET.

There are currently two kinds of mobile wireless networks. The first is known as infrastructure networks with fixed and wired gateways. Typical applications of this type of "one-hop" wireless network include wireless local area networks (WLANs). The most commonly used wireless technology is the WiFi (802.11) also known as WLAN which enables network communication via the Internet Protocol (IP) [Basile'03]. There are two modes in which the WLAN technology can be used. The most common is the "Access Point" or *hotspots* mode where the clients, usually laptops or all other kind of mobile devices, connect to a network via an access point. In this scenario, the access point is typically connected to a wired network, which usually offers services e.g. Internet access.

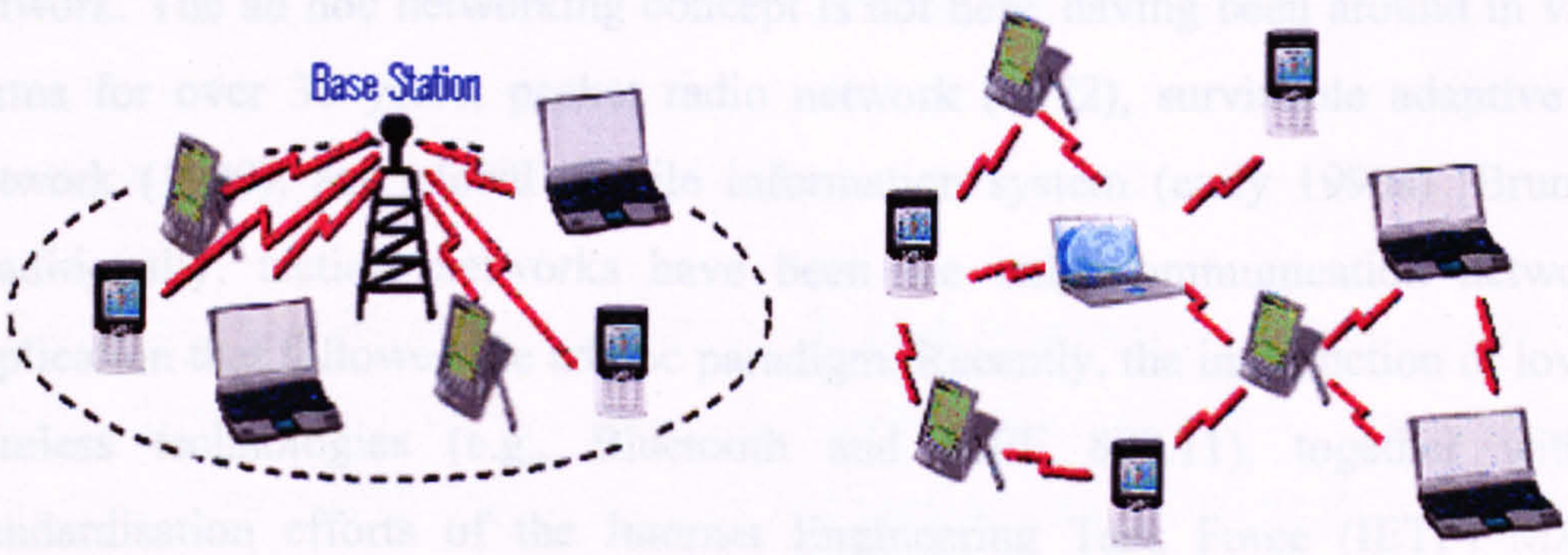



1) Infrastructure-based wireless network 2) Mobile Ad hoc wireless networks

**Figure 1-1: Examples of Infrastructure and infrastructure-less wireless Ad hoc networks**

typical nodes cannot be trusted with these important network functions. Thus, security has become an essential consideration in MANET, especially in open MANET, where a variety of mobile nodes supplied by different manufacturers compose a MANET in a self-organizing manner and share their resources for global connectivity with their own goals. Moreover, MANET nodes still have small storage, low bandwidth, high error rates, and limited battery power, in spite of recent appreciable advances in terms of power efficiency, flexibility, and robustness. For these reasons, common security algorithms designed for traditional networks are difficult to use in MANET. Hence, new security approaches need to be developed for MANET.

There are currently two kinds of mobile wireless networks. The first is known as infrastructure networks with fixed and wired gateways. Typical applications of this type of "one-hop" wireless network include wireless local area networks (WLANs). The most commonly used wireless technology is the WiFi (802.11) also known as WLAN which enables network communication via the Internet Protocol (IP) [Basile'03]. There are two modes in which the WLAN technology can be used. The most common is the "Access Point" or *hotspots* mode where the clients, usually laptops or all other kind of mobile devices, connect to a network via an access point. In this scenario, the access point is typically connected to a wired network, which usually offers services e.g. Internet access.




1) Infrastructure-based wireless network 2) Mobile Ad hoc wireless networks

**Figure 1-1: Examples of Infrastructure and infrastructure-less wireless Ad hoc networks**

The second type of mobile wireless network is the infrastructure-less mobile network, commonly known as the Mobile ad hoc network (MANET). Figure 1-1 shows examples of the two kinds.

The remainder of this chapter is structured as follows. In section 1.1, 1.2 and 1.3 we introduce Mobile Ad hoc Networks, their applications and their main characteristics respectively. The next two sections 1.4 and 1.5 discuss the wider context and outline the problem of node misbehaviour and its impacts in Mobile ad hoc networks. Section 1.6 sets out the aims of the thesis before we detail the novel results of our work in section 1.7. The chapter closes with a description of the thesis structure in section 1.8.

## 1.1 Mobile Ad hoc Network

Mobile ad hoc networks (MANET) are collections of mobile nodes connected together over a wireless medium. Nodes are computing and communication devices that can be laptop computers, PDAs, mobile phones or even sensors. These nodes can freely and dynamically self-organise into arbitrary and temporary ad hoc network topologies, allowing people and devices to seamlessly internetwork in areas with no pre-existing communication infrastructure (e.g., disaster recovery and battlefield environments). Nodes in the ad hoc network are often mobile, but can also consist of stationary nodes, such as an access point to the Internet or as a wireless sensor network. The ad hoc networking concept is not new, having been around in various forms for over 30 years, packet radio network (1972), survivable adaptive radio network (1980), and global mobile information system (early 1990s) [Bruno'05]. Traditionally, tactical networks have been the only communication networking application that followed the ad hoc paradigm. Recently, the introduction of low-cost wireless technologies (e.g., Bluetooth and IEEE 802.11), together with the standardisation efforts of the Internet Engineering Task Force (IETF) MANET Working Group [IETF'07], have been generating renewed and growing interest in research and development of MANET outside the military field. IETF MANET WG is standardizing four routing protocols, and 802.11 wireless cards are ubiquitous.

MANET are built on a mix of fixed and mobile nodes interconnected via wireless links to form a multi-hop ad hoc network. Users' devices are an active part of the network. They dynamically join the network, acting as both user terminals and routers for other devices, consequently further extending network coverage.

## 1.2 MANET Applications

The set of applications for MANET is diverse, ranging from large-scale, mobile, highly dynamic networks, to small, static networks that are constrained by power sources. Besides the legacy applications that move from traditional infrastructure environment into the ad hoc context, a great deal of new services can and will be generated for the new environment [Akyildiz'05]. Typical applications include:

***1) Wireless Sensor network (WSN).*** It is one of the famous and successful applications of MANET. It is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications where military sensor networks deploy to detect enemy movements and to allow the military to take advantage of commonplace network technologies to maintain an information network between the soldiers, vehicles, and military information head quarters. It can be also used for Chemical/Biological weapon detection. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

***2) Commercial applications.*** Ad hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. in fire, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed. Information is relayed from one rescue team member to another over a small handheld. Other commercial scenarios include e.g. ship-to-ship ad hoc mobile communication, law enforcement,

intelligent transportation systems, public Internet access and public safety, e.g. police, fire departments, first responders, and emergency services.

***3) Local level.*** MANETs can autonomously link an instant and temporary multimedia network using notebook computers or palmtop computers to spread and share information among participants at e.g. conferences or classrooms. Another appropriate local level application might be in home networks where devices can communicate directly to exchange information. Similarly in other civilian environments like taxicab, sports stadium, boat and small aircraft, mobile ad hoc communications will have many applications.

***4) Personal Area Network (PAN).*** Short-range MANET can simplify the intercommunication between various mobile devices (such as a PDA, a laptop, and a cellular phone). Tedious wired cables are replaced with wireless connections. Such an ad hoc network can also extend the access to the Internet or other networks by mechanisms, e.g. Wireless LAN (WLAN), GPRS, and UMTS. The PAN is potentially a promising application field of MANET in the future pervasive computing context.

## 1.3 Characteristics of MANET

MANET has the following characteristics:

***1) Multi-hop routing.*** Basic types of ad hoc routing algorithms can be single-hop and multi-hop, based on different link layer attributes and routing protocols. Single-hop MANET is simpler than multi-hop in terms of structure and implementation, with lower cost of functionality and applicability. When delivering data packets from a source to its destination out of the direct wireless transmission range, the packets should be forwarded via one or more intermediate nodes.

***2) Dynamic network topology.*** Since the nodes are mobile, the network topology may change rapidly and unpredictably and the connectivity among the terminals may

vary with time. MANET should adapt to the traffic and propagation conditions as well as the mobility patterns of the mobile network nodes. The mobile nodes in the network dynamically establish routing among themselves as they move about, forming their own network on the fly. Moreover, a user in the MANET may not only operate within the ad hoc network, but may require access to a public fixed network (e.g. Internet).

***3) Autonomous terminal.*** In MANET, each mobile terminal is an autonomous node, which may function as both a host and a router. In other words, besides the basic processing ability as a host, the mobile nodes can also perform switching functions as a router. So usually endpoints and switches are indistinguishable in MANET.

***4) Distributed operation***. Since there is no backbone network for the central control of the network operations, the control and management of the network is distributed among the terminals. The nodes involved in a MANET should collaborate amongst themselves and each node acts as a relay as needed, to implement functions e.g. security and routing.

***5) Fluctuating link capacity***. The nature of high bit-error rates of wireless connection might be more profound in a MANET. One end-to-end path can be shared by several sessions. The channel over which the terminals communicate is subject to noise, fading, and interference, and has less bandwidth than a wired network. In some scenarios, the path between any pair of users can traverse multiple wireless links and the link themselves can be heterogeneous.

***6) Light-weight terminals***. In most cases, the MANET nodes are mobile devices with less CPU processing capability, small memory size, and low power storage. Such devices need optimised algorithms and mechanisms that implement the computing and communicating functions.

## 1.4 Node Misbehaviour in MANET

Misbehaviour in mobile ad-hoc networks occurs for several reasons [Ang'04]. Mainly Selfish nodes misbehave to save power or to improve their access to service relative to others. These nodes aim to get the greatest benefits from the networks while trying to preserve their own resources, e.g. battery life or bandwidth.

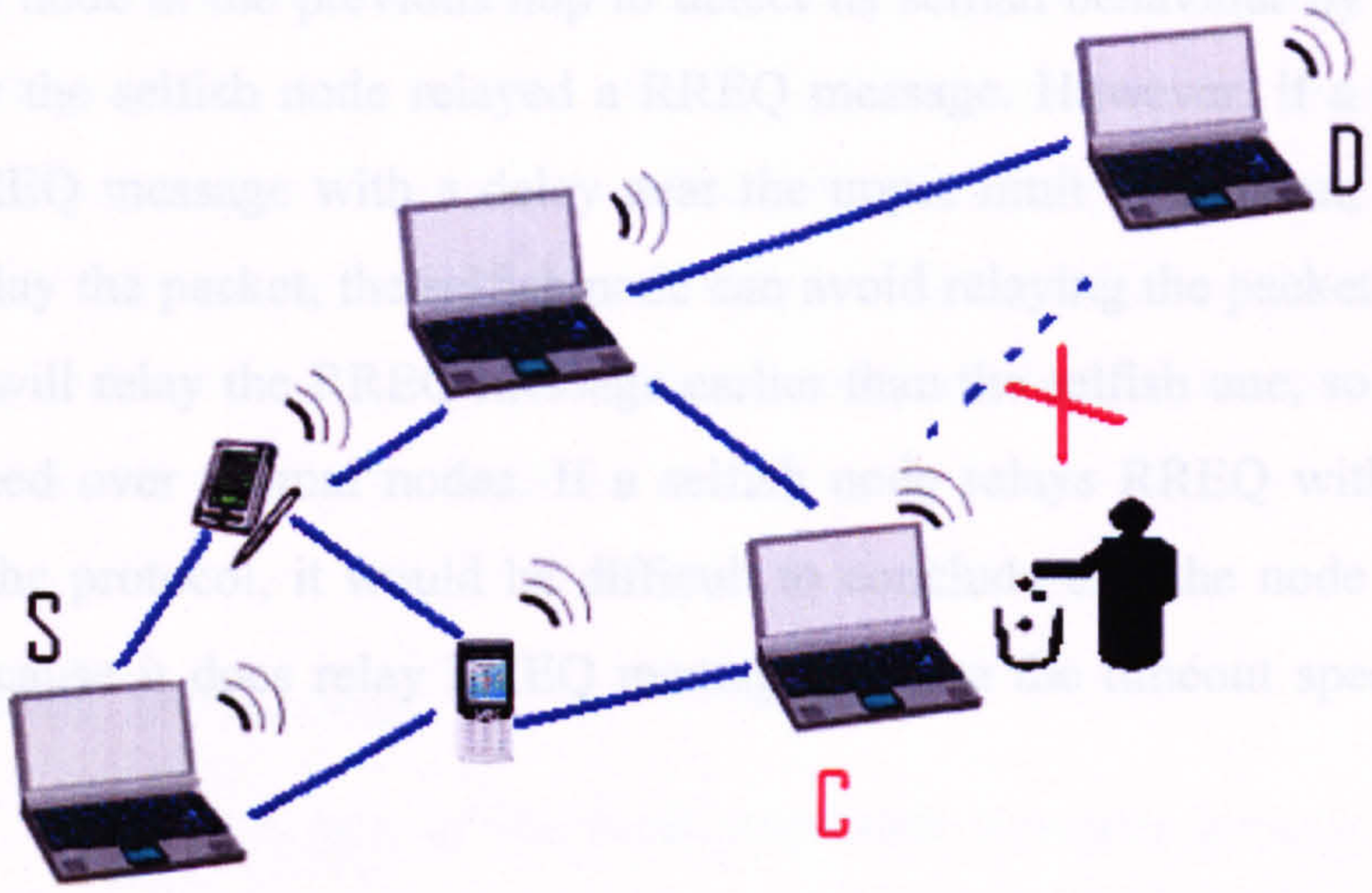



**Figure 1-2: An example of node misbehaviour in MANET**

Selfish nodes attempt to maintain communications with the nodes it wants to send data packets to but may refuse to cooperate when it receives routing or data packets that it has no interest in. Therefore, it may either drop data packets as illustrated in figure 1-2, where node C misbehaves by dropping packets that should be forwarded to node D, or refuse to retransmit routing packets that it has no interest in. These nodes have four patterns of selfish behaviour as follows:

**1)** Will not relay route request (RREQ) messages: while normal nodes relay RREQ messages to each other, selfish nodes will not relay these messages to avoid being included in the others' routes and avoid routing for others.

**2)** Will not send HELLO messages: while normal nodes send HELLO messages, a kind of route reply (RREP) messages sent periodically to notify neighbour nodes of their presence and manage link state, selfish nodes will not handle messages that do

not concern themselves. By behaving so, selfish nodes have no risk of being suspected by other nodes because they are not seen by others until they themselves send the first packet. As in 1), they will not relay RREQ messages.

**3)** Intentionally delays relaying RREQ messages: after a selfish node that behaves as in 1) and 2) sends a packet, it will be found by its neighbour nodes. Then it would be easy for the node at the previous hop to detect its selfish behaviour by watching to see whether the selfish node relayed a RREQ message. However, if a selfish node relays a RREQ message with a delay near the upper limit of timeout, and another node can relay the packet, the selfish node can avoid relaying the packet because the other node will relay the RREQ message earlier than the selfish one, so a route will be established over normal nodes. If a selfish node relays RREQ within the time defined in the protocol, it would be difficult to conclude that the node is behaving selfishly because it does relay RREQ messages before the timeout specified in the protocol.

**4)** Relays routing messages but not data packets: while a selfish node may correctly handle routing messages, it will not relay data packets.

These selfish behaviours are difficult to distinguish from the packet loss of normal nodes or faulty nodes that simply misbehave accidentally [Buchegger'05]. Regardless of the motivation for misbehaviour its impact on the mobile ad-hoc network proves to be detrimental, decreasing the performance and the fairness of the network, and in the extreme case, resulting in a non-functional network.

Recent studies show that most of one node's energy in MANET is likely to be devoted for packets relaying. For instance, the simulation study in [Buttyan'03] shows that when the average number of hops from a source to a destination is around 5, then almost 80% of a node's transmission energy will be devoted to forward packets for others. This motivates nodes to behave selfishly, and makes them unwilling to relay packets not of direct interest to them. Another motivation for dropping data packets is to launch a Denial of Service (DoS) attack targeting either the source or the destination of packets. The full reliance on nodes' cooperation

makes ad hoc networks hugely vulnerable to this attack. The packet dropping misbehaviour may lead to serious problems when performed by many nodes in the network, such as throughput degradation, latency rise, and network partition that threats the service availability which is one of the security requirements, which will be discussed later in chapter 2. All these problems affect both well-behaving and misbehaving nodes. Marti et al. [Marti'00] have shown by simulation that if 10% to 40% among the network's nodes misbehave on data forwarding, then the average throughput degrades by 16% to 32%. Another study performed by [Buttyan'01] has been devoted to investigate the impact of the network size by simulating networks of different sizes with the same density, and comparing the effect of the same rates of misbehaving nodes on the throughput. The results show that large networks are more vulnerable to this kind of misbehaviour.

## 1.5 Problem Definition

In MANET, security is one of the most important concerns because a MANET system is much more vulnerable to attacks than a wired or infrastructure-based wireless network. Designing an effective security protocol for MANET is a very challenging task. This is mainly due to the unique characteristics of MANET, namely shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among users, limited availability of resources, and physical vulnerability.

Due to this infrastructure-less features of MANET mentioned above, all networking functions must be performed by the nodes themselves. In particular, data packets sent between distant nodes are expected to be relayed by intermediate nodes, which act as routers and provide the forwarding service. The forwarding service correctly relays the received packets from node to node until reaching their final destination, following routes selected and maintained by the routing protocol. The routing and data forwarding together are at the core of the network layer. Even more challenging, stationary MANET and its successful type of applications namely the static wireless sensor networks are highly susceptible to denial of service attacks due to their inherent characteristics i.e., low computational power, limited memory and

communication bandwidth coupled with use of insecure wireless channel. Therefore, a sink/black-hole attack can be easily launched by an adversary node in the sensor network. The malicious node starts advertising very attractive routes to data sink (e.g. zero-cost routes to every other node) [Ahmed'05]. The neighbour nodes select the malicious node as the next hop for message forwarding considering it a high quality route and propagate this route to other nodes. Almost all traffic is thus attracted to the malicious node that can either drop it, selectively forward it based on some malicious filtering mechanism or change the content of the messages before relaying it. This malicious node has thus formed a sink-hole with itself at the centre. The sink-hole is characterized by intense resource contention among neighbouring nodes of the malicious node for the limited bandwidth and channel access. These result in congestion and can accelerate the energy consumption of the nodes involved, leading to the formation of routing holes due to nodes failure. With sink holes forming in a sensor network, several other types of denial of service attacks are then possible [Wood'02].

On the other hand, forwarding packets for other nodes is not always in the direct interest of every node, so there is no good reason to trust nodes and assume that they always cooperate. Indeed, nodes try to preserve their resources, and particularly their batteries. To mitigate such problem, many power-aware routing protocols have been proposed [Doshi'02b, Doshi'02a, Krunz'04, Jung'05, Djenouri'06a] but all of these solutions do not completely solve the problem due to the complex nature of the network. As a result, users will be concerned about their limited batteries, which may lead the nodes to behave selfishly and drop packets. A selfish node regarding the packet forwarding process is the one that takes advantage of the distributed forwarding service and asks others to forward its own packets, but would not correctly participate in this service. Without appropriate countermeasures, the effects of misbehaviour dramatically decrease network performance. Depending on the proportion of misbehaved nodes and their specific strategies network throughput can be severely degraded, packet loss increases, nodes can be denied service, and the network can be partitioned. Moreover, this misbehaviour represents a potential danger that threatens the quality of service, as well as one of the most important

network security requirements, namely the availability. The detrimental effects of misbehaviour result in unfairness and degraded performance and they can endanger the functioning of the entire network.

## 1.6 Thesis Aims

It is the aim of this thesis to describe a framework for detecting nodes misbehaviour in stationary MANET environments. The framework combine to provide high accuracy detection rates at lower cost in terms of communication overhead and energy consumption. In order to do this we must first address three issues:

1. What are the requirements for a misbehaviour detection framework within ad hoc networking environments? Any requirements should take into account both the characteristics of MANET, in particular low power storage and multi-hop capabilities as well as the need for low cost. The design of a framework should have both simplicity as well as precision.

2. What existing techniques are appropriate in developing solutions for these requirements? This literature survey shall examine the building blocks of misbehaviour detection enhancement as well as those efforts that have contributed directly to the knowledge about MANET security.

3. Finally the framework should enable us to better understand the complex problem of Misbehaviour, both generally, and specifically in terms of mobile ad hoc networking. The framework should allow us to ask further research questions regarding the field.

## 1.7 Novel Research Contributions

The problem of node misbehaviour in stationary MANET and static wireless sensors network is rarely dealt with in a serious enough manner. All of the existing research solutions in the area of Data forwarding Detection are only focusing on mobile scenarios in MANET (i.e. when nodes are freely mobile). However, little has been

done in the terms of examine and applying such mechanisms to Stationary MANET when nodes are static or with very low level of mobility. An example of stationary MANET is a wireless sensor networks for civil and military applications (e.g. security management, surveillance, automation, wildlife and environmental monitoring) that are typically deployed today, and have small to medium scale (tens to hundreds of sensors) across small to medium geographical distances. Since every node is potentially a router, this adds new vulnerabilities to the network-layer problems experienced on the Internet. Misbehaviour detection protocols must be simple enough to scale up to large networks such as stationary wireless sensors networks, yet robust enough to cope with failures that occur many hops away from a source. This thesis contributes to our understanding of Security and Misbehaviour Detection Systems in wireless ad hoc environments in the following ways:

- Our first contribution is to provide a set of requirements for an efficient misbehaviour detection framework in mobile ad hoc networking environments and examine these against existing research in literature [Fahad'06a]. These requirements enable the network providers to operate a secure system whilst consuming low energy and producing low communication overhead. The requirements are similar to those of existing work but have been reconsidered to reflect the changing nature of ad hoc networks, especially as it applies to sensor networks. A survey of research literature in the field revealed that no results completely meet these requirements. These techniques focus on either high accuracy detection rate at huge cost in terms of energy and communication overhead such as [Kargl'04, Djenouri'05] or on poor accuracy detection rate at a medium cost like [Marti'00, Buchegger'02b, He'04, Michiardi'02a, Miranda'03, Yang'02]. Others such as [Buttyan'03, Zhong'03, Papadimitratos'03] fail because they aim to encourage good behaviour among nodes without fair and firm mechanisms to deal with those who misbehave. Additionally we also bring together relevant ideas of use in search for effective misbehaviour detection in MANET environments.

- Using the set of requirements and inspiration from relevant literature, this thesis proposes a novel solution to accurately and effectively detect and deal with node misbehaviour in mobile ad hoc networking environments including wireless sensor network, and it is called Sessions-based Misbehaviour Detection Formwork (SMDF) [Fahad'07a, Fahad'07b]. The new framework consists of three components, Detection Component, Decision Component and finally Isolation Component. Each component in SMDF provides different functionalities, and all of these components are integrated to provide an efficient and robust solution against node misbehaviour in MANET. The major advantages of our SMDF included its capability of working either independently or be integrated with other routing protocols. The new framework is also extensible and flexible as it has the capability of adding new components to it or removing existing components from it as necessary. Moreover, the new framework is transparent in terms of its capability to integrate with other mechanisms as required. The detection component contains our novel Sessions-based Misbehaviour Detection Protocol SMDP to detect selfish or malicious nodes that drop packets partially or completely to launch either black-hole or data dropping attacks. For the decision component we have enhanced an existing Bayesian approach to decide whether the node deliberately misbehaved or not. For the Isolation component, we have modified an existing approach and used an Observation-Based Protocol to isolate misbehaving nodes. It uses neighbouring observations experience to isolate misbehaved nodes. We analysed and evaluated the proposed framework by simulation techniques. Our evaluation was focused on six important parameters, namely Throughput, Overheard, True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. By comparing our results to those of other mechanisms available on literature, we showed that our solution has low cost in terms of communication overhead than other approaches. We showed also that our framework has the lowest False Positive Detection Rate of misbehaving nodes amongst other approaches, and that it has highest value of True Positive Detection Rate compared with other approaches. Our evaluation also

showed that our solution has lower energy consumption rate compared with other existing approaches. The evaluation showed also that our framework is scalable and can work with higher number of nodes, especially in wireless sensor networks.

- The first new component of our framework we have developed is the Detection Component, which is the most important component in any misbehaviour detection framework. For this reason we have developed a new novel Sessions-based Misbehaviour Detection Protocol (SMDP) [Fahad'06b, Fahad'07a, Fahad'07b]. The SMDP deals with the network in terms of sessions, and uses cross-layer collaboration between the session layer and the network layer in order to know the start and the end of each session. In SMDP each node in the route session monitors all of its direct neighbours within a one hop communication, and checks whether they correctly forward packets or drop them completely or partially in order to launch an attack such as black-hole and packet dropping attacks. SMDP is cost effective as it reduces the communication overhead, by using only one hop communication (no flooding), and sending control packets only at the end of sessions, instead of doing so for each packet, in contrary to the current solutions that exist in literature. The new SMDP also has an advantage of being independent of the routing protocol, as well as its ability to work with any MANET routing protocol, unlike most of the existing mechanism in literature who work as an extension of one particular routing protocol. We evaluated the proposed protocol by simulation and showed that our approach is more efficient and scalable than other approaches found in the literature. It showed also that it has a low cost in terms of both communication overhead and energy consumption compared to other approaches.

- We have developed the second component of our framework (SMDF) which is the Decision Component [Fahad'07a, Fahad'07b]. After detecting those nodes dropping packets using our detection component SMDP, we have used the decision component of the SMDF to decide whether the nodes

misbehaved or not. As the nodes might drop packets for innocent reasons such as collision or faulty packets, the decision component of SMDF take all of this into account, in order to make a fair decision. For the decision component we have enhanced an existing mathematical estimation method, which has been used in the literature and we have modified it effectively to suit our new framework requirements. It based on Bayesian standard approach, which consists of estimating a parameter the observations of which follow a Bernouli distribution by a Beta distribution. In our approach, well behaving nodes improve their reputation, whereas misbehaving nodes in terms of either intentional or unintentional packet dropping will decreases it. Moreover, our approach allows redemption before making decisions, and decreases false accusations due to, for example channel conditions or collision. Furthermore, in Bayesian approach only the latest observations are watched over, and not all the observations, as a result it has the advantage of not requiring a memory. We evaluated our proposed approach by simulations and showed that our approach is more accurate in identifying the real misbehaving nodes than existing approaches. It also has lower communications overhead compared to other approaches.

- Having identified the misbehaving nodes locally, we developed our Isolation Component which will then punish them by not routing packets through them and by not forwarding packets for them. For this component, we have modified an existing approach and we used an Observation-Based Protocol to isolate misbehaving nodes. Once a node is judged locally as misbehaving by some other node, this latter must approve its detection to ensure the isolation by all nodes. The Observation-Based Protocol uses neighbouring observations experience to mitigate false detections and false accusations vulnerabilities that exist in other approaches such as [Marti'00, Buchegger'02b, Djenouri'05]. In this protocol, a node that detects and accuses another node of misbehaviour must approve its accusation before taking any measure against it. It should not isolate the assumed misbehaving unilaterally, because this could result in false detections against it. However, it could

avoid routing its own packets through this node in all cases. The Observation-Based Protocol enforces the accusing node to collect a certain number of observations from neighbouring nodes in terms of signatures before isolating the detected node. Once the accuser node collects this number, it broadcasts an Isolation packet including all observations through the network to isolate the misbehaving node. This broadcast will not be performed until a node is detected and approved as misbehaving. As a result, our solution produces less overhead as long as nodes well-behave, as no opinions are exchanged periodically. Our simulation results suggest that our approach has the lowest percentage in falsely accusing well behaving nodes of misbehaviour compared to other existing approaches. It showed also that it has a low cost in terms of communication overhead.

- Our final contribution is that this research poses some new questions that had not been made explicit before. Among the questions for further work are issues of tackling detection complications in hybrid ad hoc network environments. Two other important issues raised are those of dealing with control packets dropper and mobility handling issues in terms of mobile Wireless Sensor Networks. These questions are examined together with an evaluation of the project in terms of the shortcomings of the framework and comparison with closely related work.

## 1.8 Thesis Structure

The thesis is structured into the following six chapters:

Our introduction to the area in Chapter 1 discusses the wider context and outlines the problem of node misbehaviour in Mobile ad hoc network. It outlines the definition of MANET and its main applications. It also identifies MANET main characteristics that make the design of routing and detection protocols of such kind of networks a challenging task. Chapter 1 also highlights the consequences of node misbehaviour

in MANET and its impact on MANET performance. Finally, we outline the thesis aims and the contribution of our work, and the structure of the thesis.

In Chapter 2 we give an overview of MANET architecture and its security issues. We first discuss the security issues related to MANET Network layer. We also describe two main MANET routing protocols, namely Dynamic Source Routing (DSR) and Ad hoc On-Demand Distance Vector (AODV). We highlight the security threats and attacks on routing protocols and list the most common attacks such as Denial of Service (DoS). Then we move down one step to the MAC layer and discuss its security issues in MANET, which includes misbehaviour issues in the channel access. At the end of this chapter we list the main security attributes and discuss some methods of achieving them including cryptography.

In Chapter 3 we survey the literature and related work. We present and discuss the existing solutions that aim at detecting misbehaviour on packet forwarding when it appears in the network. We classify these solutions into two main techniques, Reactive and Preventive. The reactive solutions are split into two main classes, monitoring and reputation-based solutions. Chapter 3 also points out the main drawbacks of the existing work and issues that need to be addresses.

In Chapter 4 we present our framework, Sessions-based Misbehaviour Detection Framework (SMDF) and briefly describe its three components. First we describe our research objectives that form a comprehensive set of support mechanisms and schemes. We discuses the gaps in the current knowledge that this thesis will address in the requirements review. We identify requirements, issues and challenges are important when designing an effective misbehaviour Detection Framework in stationary MANET and static wireless sensor network. Then we present our SMDF and its cross layer collaboration.

In Chapter 5 we describe each component of our framework SMDF in detail. We start with our new detection component and its novel Sessions-based Misbehaviour Detection Protocol (SMDP), where we explain the concepts of the monitoring

mechanism our protocol is using and the algorithm to do so. We give two case studies in order to illustrate how our SMDF works. Then we explain our modified Bayesian approach for the decision component. This is followed up by the Isolation component explanation where misbehaving nodes will be penalised for their attacks.

In Chapter 6 we present the simulation design, analysis, results and performance evaluation of our SMDF. We start by showing how selfish misbehaviour affects the performance of MANET in terms of reducing the throughput. We then evaluate our system through simulation with two of the well-known existing systems, in terms of six different metrics. We then look at the overall achievements including evaluation against our initial requirements specified in chapter 4 and discuss the problems remaining.

In Chapter 7, we look back at the achievements of the thesis and conclude what we have learnt from the field of misbehaviour detection in MANET. The chapter is then able to pose some further research questions, and finally provide our conclusion.

# 2. MANET: SECURITY OVERVIEW

Security has become a major concern in order to provide safe and protected communication between MANET's nodes, especially in a hostile environment. Unlike the wire-line networks, the unique characteristics of MANET's create a number of huge challenges to security design, such as open peer-to-peer network architecture, shared wireless medium, stringent resource constraints, and nodes' misbehaviour. These challenges clearly give a motivation for building robust security solutions that achieve both broad protection and desirable network performance. This Chapter gives an overview of MANET architecture and its security issues. We first discuss the security issues related to MANET's network layer. We also describe two main MANET routing protocols, namely Dynamic Source Routing (DSR) and Ad hoc On-Demand Distance Vector (AODV). We highlight the security threats and attacks on routing protocols and list the most common attacks such as Denial of Service (DoS) and black-hole attacks. Then we move down one step to the MAC layer and discuss its security issues in MANET, which includes misbehaviour issues in the channel access. At the end of this chapter we list main security attributes and discuss some methods of achieving them including cryptography.

## 2.1 MANET Architecture Overview

Ad hoc networks are composed of autonomous nodes that are independent of any fixed infrastructure as shown in figure 2-1. Mobile ad hoc networks have a fully decentralised topology and they are dynamically changing. Besides this, the wireless transmission medium introduces limitations in communication. For these reasons,

providing security guarantees is particularly difficult. As mentioned in chapter 1, in a mobile ad hoc network every node acts as a router for its neighbours. The routing protocols that have been proposed assume that the nodes will fully participate. Unfortunately, node misbehaviour is a likely phenomenon. This misbehaviour is due to selfish or malicious reasons. Another reason is a faulty link due to the wireless medium. Misbehaviour can take place at all layers. At the Physical layer a misbehaving node can increase its transmitting power, adversely affecting the network performance. At the MAC (Medium Access Control) layer a node may choose to avoid waiting for his turn to access the medium, taking unfair advantage of the shared medium.

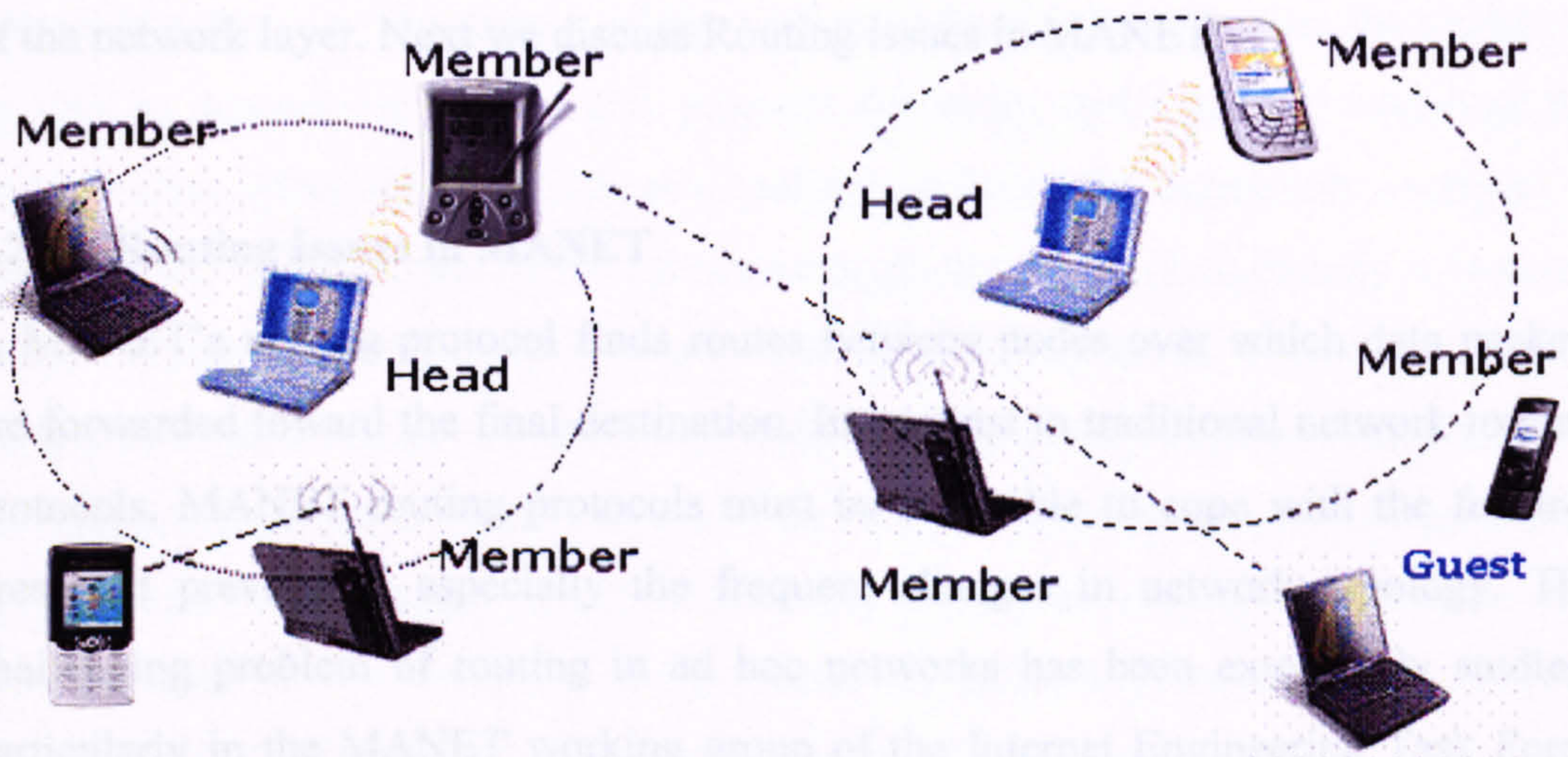



**Figure 2-1: MANET Architecture**

The basic threat at the Network layer is the non-cooperative behaviour as far as packet forwarding is concerned. The proper execution of a routing protocol demands that the intermediate nodes in a path forward correctly the packets to the intended receivers. Misbehaving nodes may deny forwarding these packets. A routing protocol for MANET should give incentives for cooperative action or, at least it should be able to detect misbehaving nodes and punish them. At the Transport layer there is a research action that aims to improve the performance of TCP over wireless networks

[Hsieh'02, Chen'06]. At the Application layer there is a huge effort of developing applications that can perform well over mobile ad hoc networks. The misbehaviour problem is not clearly addressed in this area yet. There exist only a few intrusion detection techniques [Anjum'03, Huang'03, Kachirski'03, Zhang'03, Liu, Y.'06, Karim'06] that operate at this layer which are based on trace analysis of historical data.

## 2.2 Network Layer Security Issues in MANET

Protecting the network layer in MANET is a highly important issue. The core functionalities provided in this layer are routing and packet forwarding, and are closely related. These services (routing and data forwarding) together are at the core of the network layer. Next we discuss Routing issues in MANET.

### 2.2.1 Routing Issues in MANET

A MANET's routing protocol finds routes between nodes over which data packets are forwarded toward the final destination. In contrast to traditional network routing protocols, MANET routing protocols must be adaptable to cope with the features presented previously, especially the frequent changes in network topology. The challenging problem of routing in ad hoc networks has been extensively studied, particularly in the MANET working group of the Internet Engineering Task Force (IETF) [IETF'07]. These studies have resulted in several mature protocols [Perkins'94, Johnson'96, Toh'96, Murthy'96, Park'97, Ko'98, Perkins'99], which can be divided into two classes: proactive (table driven) and reactive (on-demand). It has been shown in [Royer'99, Ashwini'05] that reactive protocols are more adaptable to MANET environments than proactive protocols. However, the problem with all of these solutions is that they trust all nodes and do not account for security, therefore they are vulnerable to attacks. It is highly important to secure the routing protocol. If the routing protocol can be subverted and messages can be altered in transit, then no amount of security on the data packets at the upper layers can mitigate threats. Recently, several secure MANET routing protocols have been proposed [Perrig'01b, Hu'02a, Hu'02b, Castelluccia'02, Papadimitratos'02, Sanzgiri'02, Zapata'02, Hu'03,

Hu'04]. In the following we give a description of DSR and brief one for AODV, two protocols adopted by the IETF MANET working group.

### 2.2.2 Dynamic Source Routing (DSR)

Misbehaviour detection systems for mobile ad-hoc networks have mostly built on Dynamic Source Routing (DSR), monitoring node behaviour with a watchdog component. DSR is a protocol developed for routing in mobile ad-hoc networks and was proposed for MANET by [Broch'04]. Dynamic source routing is a Source routed On-Demand routing protocol in Ad Hoc networks. Source Routing is a technique by which the sender of a packet determines the complete sequence of nodes through which the node has travel to reach its final destination. The sender of the packet explicitly mentions the list of all nodes in the packet's header, identifying each forwarding 'hop' by the address of the next node to which to transmit the packet on its way to destination host. In this protocol the nodes don't need to exchange the routing table information periodically and thus reduces the bandwidth overhead in the network. Each mobile node participating in the protocol maintains a 'routing cache', which contains the list of routes that the node has learnt. Whenever the node finds a new route it adds the new route in its 'routing cache'. Each mobile node also maintains a sequence counter 'request id' to uniquely identify the requests generated by a mobile host. The pair < source address, request id > uniquely identifies any request in the ad hoc network. The protocol does not need transmissions between hosts to work bi-directionally. The main phases in the protocol are Route Discovery process and Route Maintenance process.

#### a. Route Discovery

Route discovery allows any host to dynamically discover the route to any destination in the Ad Hoc network. In DSR, a source initiates a route discovery process when the source wants to send a packet to a destination to which it doesn't have a valid route. The source, if it has a valid route in its routing cache then it uses it otherwise it sends a route request packet by broadcasting it to the neighbours. The route request packet contains the source address, request id and a route record in which the sequence of hops traversed by the request packet before reaching the destination are noted down. A node upon getting a route request packet does the following:

1. It checks to see if it has the pair <initiators address, request id> in its list of recently seen requests if so discards the packet.
2. Otherwise, if this host's address is already present in the route record of the request packet then it discards the packet. This eliminates the looping problem.
3. Otherwise, if the destination the source is looking for matches with its address then it sends the route reply packet to the initiator containing the list of nodes the request packet has traversed before it reached the destination.
4. Otherwise, it appends its own address to the route request packet and rebroadcasts it. The route request travels the Ad Hoc network until it reaches the destination node.

**b. Route Maintenance**

Route maintenance is a procedure for monitoring the correct operation of routes in use. The host that uses the route does this maintenance. Since the nodes do not exchange any routing information in this protocol the route maintenance procedure monitors the operation of the route and informs the source of any errors. If a host node detects that its next hop neighbouring node is not working, then it will send an error packet containing its address and the address of the hop that is not working. A node upon receiving the route error packet removes the hop in error from its routing cache. Acknowledgements are used to verify the correct operation of the route. The route maintenance can be provided by using either hop-to-hop or by using end-to-end acknowledgements. Figure 2-2 shows the propagation of the Route Request and the building of route entry from the source 'S' 1 to the destination 'D' 7.

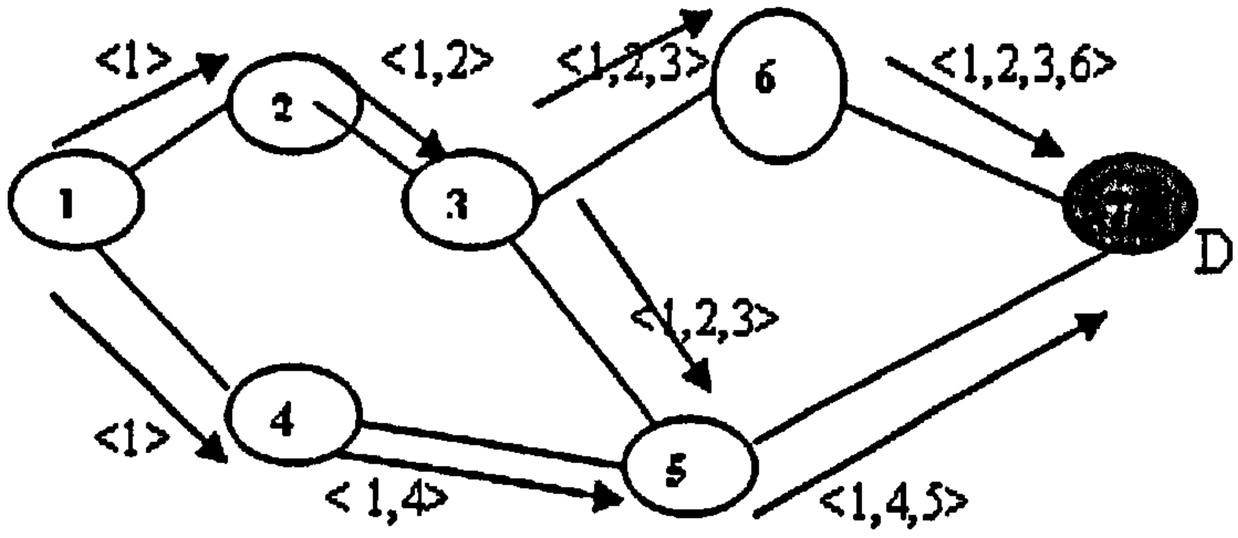



**Figure 2-2: Propagation of the DSR Route Request**

1. It checks to see if it has the pair <initiators address, request id> in its list of recently seen requests if so discards the packet.
2. Otherwise, if this host's address is already present in the route record of the request packet then it discards the packet. This eliminates the looping problem.
3. Otherwise, if the destination the source is looking for matches with its address then it sends the route reply packet to the initiator containing the list of nodes the request packet has traversed before it reached the destination.
4. Otherwise, it appends its own address to the route request packet and rebroadcasts it. The route request travels the Ad Hoc network until it reaches the destination node.

**b. Route Maintenance**

Route maintenance is a procedure for monitoring the correct operation of routes in use. The host that uses the route does this maintenance. Since the nodes do not exchange any routing information in this protocol the route maintenance procedure monitors the operation of the route and informs the source of any errors. If a host node detects that its next hop neighbouring node is not working, then it will send an error packet containing its address and the address of the hop that is not working. A node upon receiving the route error packet removes the hop in error from its routing cache. Acknowledgements are used to verify the correct operation of the route. The route maintenance can be provided by using either hop-to-hop or by using end-to-end acknowledgements. Figure 2-2 shows the propagation of the Route Request and the building of route entry from the source 'S' 1 to the destination 'D' 7.

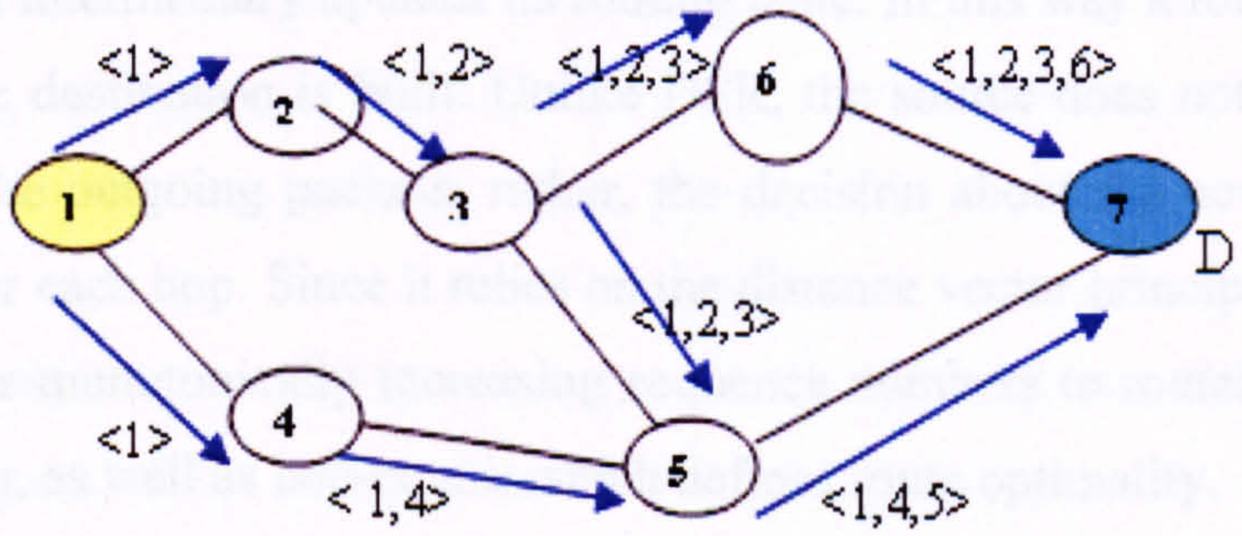



**Figure 2-2: Propagation of the DSR Route Request**

In case of hop-to-hop acknowledgements the hop in error is indicated in the route error packet. But in case of end-to-end acknowledgements the source node assumes that the last hop of the route to the destination is error. Figure 2-3 shows the propagation of the Route Reply containing the route entry from the destination 'D' 7, to source 'S' 1.

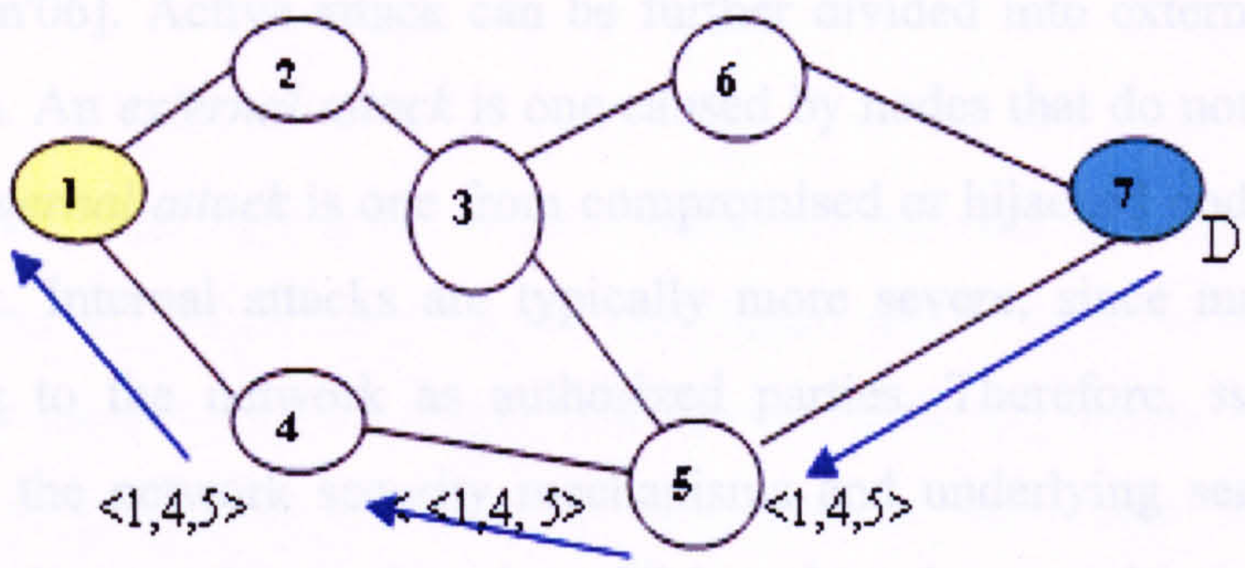



**Figure 2-3: Propagation of the DSR Route Reply**

### 2.2.3 AODV (Ad hoc On-Demand Distance Vector)

AODV [Perkins'99, Perkins'03] is a hop-by-hop routing protocol. When a node needs to send a data packet to a destination to which it has no route, it has to broadcast a RREQ to all its neighbours, then each neighbour does so until reaching the destination (or a node with a valid route to the destination). This node sends a RREP packet that travels the *inverse* path until reaching the source. Upon the reception of this reply each intermediary updates its routing table. In this way a route between the source and the destination is built. Unlike DSR, the source does not put the whole route within the outgoing packets; rather, the decision about the next hop is made separately after each hop. Since it relies on the distance vector principle [Perkins'99], AODV assigns monotonically increasing sequence numbers to routes, which define route freshness, as well as hop-count, which defines route optimality.

### 2.2.4 Routing Security Issues and Attacks in MANET

Security always implies the identification of potential attacks, threats and vulnerabilities of a certain system. Security attacks in MANET are divided into two categories: *active* and *passive* attacks [Nguyen'06]. Active attacks are performed by malicious nodes to harm the entire network operation intentionally, and include denial of service (DoS), tunnelling (wormhole attack), black hole, and impersonation [Hu'04, Nguyen'06]. Active attack can be further divided into external attacks and internal attacks. An *external attack* is one caused by nodes that do not belong to the network. An *internal attack* is one from compromised or hijacked nodes that belong to the network. Internal attacks are typically more severe, since malicious nodes already belong to the network as authorized parties. Therefore, such nodes are protected with the network security mechanisms and underlying services. On the other hand, passive attacks are done by selfish nodes whose goal is just to use their limited resources only for their own benefit. That some nodes could be selfish is a reasonable assumption, especially in the open MANET environment, since nodes owned by different commercial entities always attempt to maximize their own interests. They do not want to use their resources to support global connectivity, even though all nodes benefit from such a commitment in the long run. Among various resources associated with nodes, energy is one of the most important, so it needs to be conserved as much as possible. In terms of energy consumption, data transmission is the most expensive function in the MANET environment. To send a bit over 10 or 100 m distance, nodes consume energy that can perform thousands to millions of arithmetic operations [Yang'04]. Thus, nodes may not forward others' packets and simply discard them on purpose. Or they may excessively reduce transmission power to save energy, resulting in network partitioning. Any such feature of nodes is called selfishness.

Next, we describe some types of active attacks [Zhou'99, Deng, H.'02, Perkins'03] easily performed against a MANET in the network layer.

**Black hole**: Black hole problem in MANET [Deng, H.'02] is a serious security problem to be solved. In this problem, a malicious node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. In a flooding based protocol, if the malicious reply reaches the requesting node before the reply from the actual node, a forged route then will be created. This malicious node then can choose whether to drop the packets to perform a denial-of-service attack or to use its place on the route as the first step in a man-in-the-middle attack.

**Denial of service**: The DoS attack results when the network bandwidth is hijacked by a malicious node. It has many forms: the classic way is to flood any centralized resource so that the network no longer operates correctly or crashes. For instance, a route request is generated whenever a node has to send data to a particular destination. A malicious node might generate frequent unnecessary route requests to make the network resources unavailable to other nodes.

**Routing table overflow**: The attacker attempts to create routes to nonexistent nodes. The goal is to have enough routes so that creation of new routes is prevented or the implementation of routing protocol is overwhelmed.

**Energy consumption**: Energy is a critical parameter in the MANET. Battery-powered devices try to conserve energy by transmitting only when absolutely necessary. An attacker can attempt to consume batteries by requesting routes or forwarding unnecessary packets to a node.

**Impersonation**: A malicious node may impersonate another node while sending the control packets to create an anomaly update in the routing table.

**Information disclosure**: The malicious node may leak confidential information to unauthorized users in the network, such as routing or location information. In the end, the attacker knows which nodes are situated on the target route.

## 2.3 MANET Security Attributes

Security is the combination of processes, procedures, and systems used to ensure confidentiality, authentication, integrity, availability, access control, and non-repudiation.

- **Availability** is to keep the network service or resources available to legitimate users. It ensures the survivability of the network despite malicious incidents. Lack of availability occurs through denial of service (DoS) attacks. In MANET many of the security breaches are targeted to cause DoS attacks.

- **Confidentiality** is to keep the information sent unreadable to unauthorized users or nodes. MANET uses an open medium, so usually all nodes within the direct transmission range can obtain the data. One way to keep information confidential is to encrypt the data, and another technique is to use directional antennas. In many applications of MANET like transformation of military secrets during war, confidentiality is a major concern.

- **Integrity** is to be able to keep the message sent from being illegally altered or destroyed in the transmission. When the data is sent through the wireless medium, the data can be modified or deleted by malicious attackers. The malicious attackers can also resend it, which is called a replay attack.

- **Authentication** is to be able to identify a node or a user, and to be able to prevent impersonation. In wired networks and infrastructure-based wireless networks, it is possible to implement a central authority at a point such as a router, base station, or access point for authentication. But there is no central authority in MANET, and it is much more difficult to authenticate an entity. Without authentication an attacker can impersonate as an authenticated node and thus gain control over the entire network.

- **Non-repudiation** is related to a fact that if a node sends a message, the node cannot deny that the message was sent by it. By producing a signature for the

message, the entity cannot later deny the message. In public key cryptography, a node A signs the message using its private key. All other nodes can verify the signed message by using A's public key, and A cannot deny that its signature is attached to the message. It is particularly useful for detecting compromised nodes.

- **Access control** is to prevent unauthorized use of network services and system resources. Clearly, access control is tied to authentication attributes. In general, access control is the most commonly thought of service especially in individual computer systems.

## 2.4 Cryptography

Cryptography is the science of secret writings. Cryptanalysis is the art or science of "breaking" the cipher texts without knowing the key used for decrypting. Those who "practice" cryptography are called cryptographers and those who "practice" cryptanalysis are called cryptanalysts [Stallings'05]. Nowadays cryptography is used for securing messages, certification, services and mechanisms used for electronic equipment networks. There are two types of cryptographic systems: symmetric and asymmetric encryption.

### 2.4.1 Symmetric and Asymmetric Encryptions

Encryption is the process of encoding a text so that its original meaning is observed. Decryption is the opposite process, a mechanism to reveal the original message from the encrypted one. The term encipher and decipher are used respectively. The original or unaltered version of the message is termed as plain text and the encrypted message is called ciphertext.

**Symmetric Encryption**

This encryption uses the same secret key for encryption and decryption. The cryptographic algorithms (ciphers) used in symmetric encryption systems are divided into stream ciphers and block ciphers. The stream ciphers may encrypt only a single

bit clearly at a time, while the block ciphers may encrypt more bits (64 or 128 bits) at a time. The most challenging task in symmetric encryption is to distribute and manage the shared secret (Key). DES (Data Encryption Standard) and AES (Advanced Encryption Standard – Rijndael) are examples of symmetric encryption.

**Asymmetric Encryption / Public Key Encryption**

Unlike the symmetric encryption it uses two separate keys for encryption and decryption. So keys come in pairs called private-public key pairs. The sender encrypts the message with his private key. Prior to this operation sender must send its corresponding public key to the receiver. On receiving the encrypted text, the public key is used to decipher the original plain text. The asymmetric encryption incurs quite high computational expense for an attacker, and its application is limited where both security and efficiency are concerned. The best known asymmetric encryption system is RSA (Rivest Shamir Adleman).

Generally, the symmetric algorithms are executed more quickly than asymmetric ones. In practice they are often used together, so that a public-key algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm. This is sometimes called hybrid encryption.

### 2.4.2 Digital Signature

Digital signature is an important cryptographic primitives used for authentication, authorization and non-repudiation [Stallings'05] . Digital signature has the best use of public key cryptography. An asymmetric encryption algorithm such as RSA can be used to create and verify digital signature [Stinson'02]. The simplest form of the protocol works as follows:

Two parties Bob and Alice wish to exchange a signature:

1. Bob encrypts the document with his private key, thereby singing the document.
2. Bob sends the signed document to Alice.

3. Alice deciphers the document with Bob's public key, thus verifying the signature.

The strength of the digital signature lies with the fact that although the public-private key pair for asymmetric encryption is mathematically related, it is computationally infeasible to derive the private key from the corresponding public key. Another fundamental process, termed a "hash function," is used in both creating and verifying a digital signature. Hash functions the security primitives that ensure data integrity. Hash function is often called one-way hash function, because it is a computationally difficult problem to compute the inverse function.

A digital signature must meet the following two properties [Pfleeger'02]

- It must be authentic. If someone R receives a digital signature from S, R must be able to verify that the signature is really from S.
- It must be un-forgeable. If an entity sings a document M with signature S(M), it is not possible for other entity to produce the same pair <M, S(M)>.

In reality digital signature creation and verification are performed using the combination of hash function and asymmetric encryption. To create a digital signature the sender first computes the message authentication code (MAC) or hash of the original message and append the code with the message. Then the hash code is encrypted using asymmetric encryption. On the reception end the receiver uses the same hash algorithm to compute the hash code of the message decrypts the encrypted message using the corresponding public key and compares the hash value.

## 2.5 Key Management Security Issues

Most of the solutions proposed for securing routing and data forwarding that will be descried in the next chapter rely on cryptography described above, and assume the existence of an underlying mechanism for providing and managing keys. Many secure applications and services also use cryptography and rely on this assumption. However, because of the lack of any central infrastructure or administration, key

management is problematic in MANET. There are basically two kinds of key infrastructure. The first involves the private key infrastructure, which establishes common private keys used for symmetric cryptography, such as symmetric group keys used for securing group communications [Maki'00, Yasinsac'02, Chiang'03, Pietro'03, Lazos'03b]. The second kind is the public key infrastructure, which provides a couple of keys (public/private) used for asymmetric cryptography, as in digital signatures. Providing such an infrastructure in MANET is challenging, due to their infrastructure-less nature. Certainly, the role of this infrastructure should be spread out to all mobile nodes (or a subset of them), which form the key infrastructure. Therefore, the MANET key management system should neither trust nor rely on any fixed certificate authority (CA), but should be distributed and self-organised.

### 2.5.1 Private Key Infrastructure

The private key management protocols have been classified into two classes [Steiner'98]: key distribution protocols, which are centralised and based on a trusted third party, and key agreement protocols, which are distributed. The suitable class for our environment in MANET is certainly the second approach. [Diffie'76, Bellovin'92, Steiner'96, Becker'98, Wong'98, Ozaki'99, Sinha'99, Asokan'00, Perrig'01a, Jetcheva'01, Li'02, Naor'02, Staddon'02, Liu, D.'03, Lazos'03a, Kaya'03, Das'03, Zhu'04] are solutions belonging to this class and some of them are especially devoted to MANET.

### 2.5.2 Public Key Infrastructure

The solutions we have mentioned above are related to private key management. We will now discuss the public key management problem. In a public key infrastructure, each node has a public/private key pair. Public keys can be distributed to other nodes, while private keys should be kept confidential to individual nodes. This type of key is essential for any service or application that employs asymmetric cryptography, such as many of the protocols described earlier, which use digital signature. In a traditional public key infrastructure, there is a trusted entity called a

certification authority (CA) that distributes nodes' public keys in *certificates*. The CA has a public/private key pair. The private key is used to sign certificates binding public keys the CA provides for nodes, while the public key is used by nodes to check the certificate's authentication. However, it is problematic in MANET to establish a key management service using a *single* CA. A standard approach to improve service availability is replication, but a naive replication of the CA makes the service more vulnerable, since compromising any single replica that possesses the service private key could lead to the collapse of the entire system. To solve this problem, recent solutions propose [Ostrovsky'91, Zimmermann'95, Zhou'99, Hubaux'01, Yi'03, Capkun'03] to distribute the trust over a set of nodes by letting them share the key management responsibility.

## 2.6 MAC Layer Misbehaviour Issues in MANET

In this section we present a misbehaving activity that threatens one of the most important purposes of MAC protocols, namely fairness in channel access.

### 2.6.1 Misbehaving in Channel Access

Since there is no central authority in MANET, Wireless Medium Access Control (MAC) protocols [Jurdak'04], such as IEEE 802.11, use distributed contention resolution mechanisms for sharing the wireless channel. The contention resolution is typically based on cooperative mechanisms that ensure a reasonably fair share of the channel for all the participating nodes. In this environment, some selfish hosts in the network may misbehave by failing to adhere to the MAC protocol, with the intent of obtaining an unfair share of the channel. The presence of selfish nodes that deviate from the contention resolution protocol can reduce the throughput share received by conforming nodes. The IEEE 802.11 MAC protocol [Gast'02], which is the standard MAC protocol for wireless networks, has two mechanisms for contention resolution: a centralized mechanism called PCF (Point Coordination Function), and a fully distributed mechanism called DCF (Distributed Coordination Function). PCF needs a centralized controller (such as a base station) and can only be used in infrastructure-based networks; thus, it is not to be considered in the ad hoc mode. In contrast, DCF

is widely used in infrastructure-based wireless networks as well as in ad hoc wireless networks. DCF uses the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) option for resolving contention among multiple nodes accessing the channel. A node (sender) with data to transmit on the channel selects a *random backoff* value from range (0; CW), where CW (contention window) is a variable maintained by each node. While the channel is idle, the backoff counter is decremented by one after every time slot (a fixed interval of time), and the counter is frozen when the channel becomes busy. The node may access the channel when the backoff counter is decremented to zero. After the backoff counter is decremented to zero, the sender may reserve the channel for the duration of the data transfer by exchanging control packets on the channel. The sender first sends a RTS (request to send) packet to the receiver, then the receiver responds with a CTS (clear to send) packet. This RTS-CTS exchange is optional in IEEE 802.11; it aims to ensure the channel reservation for the duration of the data transmission. Both of the packets contain the proposed duration of the data transmission. Other nodes that overhear either the RTS or the CTS (or both) are required to defer transmissions on the channel for the duration specified in the RTS/CTS. After a successful RTS/CTS exchange, the sender transmits a DATA packet, which will be acknowledged by an ACK. If the node's data transmission is successful, the node resets its CW to a minimum value (CWmin); otherwise, if the sender does not receive the CTS, then CW is doubled, but it should not exceed a maximum value of CWmax. A misbehaving node may obtain more than its fair share of the bandwidth by:

- Selecting backoff values from a different distribution with smaller average backoff value than the distribution specified by DCF (e.g., by selecting backoff values from the range (0,CW/4) instead of the range (0,CW)) [Kyasanur'03].

- Using a different retransmission strategy that does not double the CW value after collisions.

We note that it is not beneficial for a selfish node to not delay at all or to choose a very small constant period, since this may result in a very high collision rate, and thus the loss of the packets it sends. Such selfish misbehaviour can seriously degrade the throughput of well-behaved nodes. For instance, simulation results obtained by Kyasanur and Vaidya [Kyasanur'03] show that for a network containing eight nodes sending packets to a common receiver with one of the eight nodes misbehaving by selecting backoff values from the range (0,CW/4), the throughput of the other seven nodes is degraded by as much as 50 percent. There is no published solution proposed to this complex problem, except the solution proposed by [Kyasanur'03].

## 2.7 Misbehaviour Issues in Wireless Sensor Networks (WSN)

Wireless sensor networks, a special class of ad hoc networks applied to monitoring physical environments, have recently emerged as an important application of the ad hoc network paradigm. This technology has mainly been made possible by the convergence of micro-electromechanical systems technology, wireless communications, and digital electronics, enabling the construction of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate un tethered in short distances, thus forming the sensor network [Akyildiz'02, Akyildiz'05].

A sensor network consists of hundreds to thousands of tiny devices equipped with signal processing circuits, microcontrollers, and wireless transmitters/receivers, in addition to embedded sensors. Nodes may be randomly and densely deployed over the sensing field, leading therefore to a need for auto-organization capability. Potential applications of sensor networks include, but are not limited to, geophysical monitoring (seismic activity), precision agriculture (soil management), habitat monitoring (tracking of animal herds), target tracking in battlefields, and disaster relief networks [Xu'02].

Early research efforts have focused on the development of a new network protocol stack, trying to meet performance requirements that are more stringent than in other

ad hoc networks, including energy efficiency, auto-organization, scalability to a high number of nodes, etc. However, most applications of sensor networks face acute security concerns, including packets dropping, eavesdropping, forgery of sensor data, denial of service attacks, and the physical compromise of sensor nodes [Wood'02].

It has been noticed that little research work has been conducted to investigate the development of security analysis models for ad hoc and sensor networks, especially those used for the quantitative performance evaluation of encryption algorithms, in terms of communication overhead and computational cost [Ganesan'03]. [Xie'02, Ganesan'03, Venugopalan'03] provided models that allow designers to project computational limitations and determine the threshold of feasible encryption schemes under a set of constraints for a given embedded architecture such as sensor nodes.

Although "key management" is important for ensuring confidentiality and authentication, it still remains an unsolved problem in WSNs, mainly due to Key Pre-Deployment, Shared Key Discovery and Path-key Establishment. To overcome the shortcomings of traditional pre-deployed keying approaches, essentially the large size of the loaded key ring on each node, several alternatives have been proposed such as [Hofflein'98, Eschenauer'02, Chan'03, Zhu'03, Pietro'04, Gaubatz'04]. At the routing level, many sensor network routing protocols do not consider security as a primary goal. Consequently, these protocols are more susceptible to powerful attacks than in general ad hoc networks. Powerful and dangerous security attacks can be launched against sensor networks. Sinkhole and Hello Flood are examples of such attacks. [Perrig'01b, Deng, J.'03a, Deng, J.'03b] have proposed solutions to deal with these attacks.

### 2.7.1 Misbehaviour and security in WSN Data Aggregation

Data aggregation (or data fusion) is a key emerging theme in the design and development of WSNs. In this process, intermediary nodes called "aggregators" collect the raw sensed information form sensor nodes, process it locally, and forward only the result to the end-user. This important operation essentially reduces the amount of transmitted data on the network and thus prolongs its overall lifetime, the

most critical design factor in WSNs. However, this functionality is made even more challenging due to the hostile deployment environment, which makes possible the physical compromise of aggregators and some of the sensor nodes. Indeed, possible threats can vary from denial-of-service attacks that try to stop completely this service to stealthy attacks where the attacker's purpose is to make the user accept false aggregation results. This latter is more difficult to detect. For data aggregation validity assurance, [Du'03] have proposed the use of redundant data fusion nodes as witnesses. These nodes conduct the same data fusion operations as aggregators, but send the result as a Message Authentication Codes (MAC) to the aggregator itself instead of sending it to the base station. In order to prove the validity of the aggregation results, the aggregator has to forward the received proofs from witness nodes along with its calculated result to the base station. If a compromised aggregator wants to send invalid fusion data, it has to forge the proofs on the invalid results. The aggregation result is confirmed when *n* out of *m* witness proofs agree with the aggregators results, otherwise this latter is discarded and the base station polls one of the witness node to send it the valid aggregation result. This solution is efficient when witnesses are supposed to be trusted enough; otherwise it requires an important additional overhead to attain acceptable aggregation results using the voting scheme. Moreover, the authors have not addressed issues about choosing witness nodes. In [Przydatek'03] the authors have proposed a security framework based on an aggregate-commit-prove approach to verify that the answer given by aggregators is a good approximation of the true value even if the aggregators and a fraction of the sensor nodes can be corrupted. In this approach the aggregator commits to the collected data by constructing a Merkle Hash-tree [Merkle'80]. The commitment ensures that the aggregator uses data provided by the sensors, and acts as a statement to be verified by the base station about the correctness of the aggregation results. Although the authors have proposed concrete protocols for securely computing the median, average, and some other types of specific aggregation operations, we think the proposed scheme remains somewhat generic, and may not be flexible enough to support other types of in-network processing, such as in tinyDB [Madden'02]. The in-network processing is one of the key issues that have to be considered in all layers of the WSN's network protocol stack in order to

minimize energy consumption. However, this operation cannot be efficiently done without being secured. Therefore, secure in-network processing should consider keying schemes that are more energy-efficient. Moreover, multi-tiered hierarchical aggregation approaches, such as in [Deng, J.'03c, Deng, J.'03a, Deng, J.'03b] would be the most efficient scheme when the WSN contains a high number of sensor nodes. For that, more research work should be undertaken on how to securely and efficiently construct such schemes and dynamically choose aggregation nodes.

## 2.8 Summary

In this chapter we have presented MANET architecture and its different security issues, and we have shown that the special features of this new architecture make it more vulnerable to threats, especially node misbehaviour and we noticed that solutions developed for standard networks are often either unsuitable or not directly applicable in this environment. We discussed several problems related to different layers in MANET. Starting from the network layer we discussed how important it is to protect this layer, as it is the provider of two important services, namely routing and data forwarding. We then discussed the routing issues and types in MANET, and we explained that reactive types of protocols are more adaptable to MANET environments than proactive ones. This is because, in reactive protocols a route is only calculated when it is needed, and there is no need to keep routing-information all the time to all nodes.

Due to this fact we presented two of MANET main reactive routing protocols namely Dynamic Source Routing (DSR) and Ad hoc On-Demand Distance Vector (AODV) protocols. The primary differences between AODV and DSR are: (1) DSR sources determine the whole path to the destinations, while in AODV the routing decision is made hop by hop; and (2) unlike DSR nodes, which can keep multiple paths in the routing cache, AODV nodes record the information of only a single route in the routing table. These two features of DSR are useful for increasing path reliability and overcoming misbehaving nodes.

We then highlighted and classified the security threats and attacks on routing protocols and list most common attacks namely, Denial of Service (DoS), Black Hole, Routing Table Overflow, Energy Consumption and Information Disclosure. We also emphasised the fact that security is the combination of processes, procedures, and systems used to ensure confidentiality, authentication, integrity, availability, access control, and non-repudiation. Cryptographic systems in terms of symmetric and asymmetric encryption have been discussed. Generally, the symmetric algorithms are executed more quickly than those asymmetric. In practice, symmetric algorithms are often used together with asymmetric algorithms.

Next we moved down one step to the MAC layer and discuss its security issues in MANET. In particular, we presented the selfishness on channel access misbehaviour, which affects the fairness and significantly affects the network efficiency. The only solution proposed in the literature was presented and discussed. In our discussion we illustrated how this solution may wrongly accuse well-behaving nodes, and how it is unable to detect what we called cooperative misbehaviour. This problem represents a fruitful field of research. Finally, security issues related to Wireless Sensor Networks (WSNs), which is a special type of Ad Hoc network, was outlined. This hostile environment makes adaptation of existing protocols, first proposed for general ad hoc networks, a challenging task. We believe the design of novel security mechanisms that take into account the unique features of WSNs, such as their new communication paradigm, would be a more thoughtful mechanism.

Security in MANET remains an interesting research field that includes many research topics. In the following chapter we will give more attention to one of these topic related to an emergent security problem caused by nodes misbehaviour on packet forwarding. We will survey the literature and related works and discuss the existing solutions that specifically aim at detecting such misbehaviour when it appears in MANET.

# 3. MANET NODES MISBEHAVIOUR DETECTION MECHANISEMS

In this Chapter we survey the literature and related work. We present and discuss the existing solutions that aim at detecting misbehaviour on packet forwarding when it appears in the network. This Chapter also points out the main drawbacks of the existing work and issues that need to be addresses.

## 3.1 Detection Solutions against Nodes Misbehaviour in MANET

In this section we present and discuss a number of solutions that aim at detecting selfish misbehaviour on packet forwarding when it appears in the network. These solutions can be classified into two main techniques, Reactive and Preventive [Djenouri'05a]. The reactive solutions divided into two main classes, monitoring and reputation-based solutions. The monitoring class includes basic approaches that focus on the monitoring phase and suggest techniques to control the forwarding process. Reputation-based solutions propose mechanisms to isolate the nodes detected as selfish. However, these solutions incorporate a monitoring component that uses some of the promiscuous mode monitoring technique. On the other hand, preventive techniques proactively try to mitigate nodes misbehaviour or its effects, either by motivating nodes to cooperate or by taking measures to prevent packets from being dropped before sending them.

### 3.1.1 Reactive Solutions

Here we present and discuss reactive solutions that aim at detecting selfish misbehaviour on packet forwarding when it appears in the network. As we will see, the detection may be limited to the route including the selfish node, or may give deeper information and identify the selfish node. Upon the detection of a selfish node, routing through this node will be avoided. More stringent solutions suggest punishing these misbehaved nodes by excluding them from the service, some of them allow the redemption and the reintegration of punished nodes. The reactive solutions split up into two main classes, monitoring and reputation-based solutions [Djenouri'05a]. The reactive mechanisms are not independent of the routing protocol and operate as an extension of it. Many mechanisms work only with one particular routing protocol

#### A. Monitoring-based Solutions

We will present here five monitoring approaches, two of them are based on the promiscuous mode monitoring, while the others rely on the employment of acknowledgments (ACKs). As we will see, the advantages of the promiscuous monitoring over the ACKs based monitoring employment is that the former imposes no additional overhead for monitoring, and allows monitoring of both unicast and broadcast packets. However, the promiscuous mode monitoring has many troubles regarding the accuracy of detection. The troubles of promiscuous mode monitoring include its failure to detect the misbehaviour in cases of collisions, partial collusion, and power control employment.

##### 1) End-to-End ACKs

This mechanism consists of monitoring the reliability of routes by acknowledging packets in an end-to-end manner, to make the routing protocol reliable (like TCP). That is, the destination node acknowledges the successfully received packets by sending feedback to the source. A successful reception implies that the corresponding route is operational, while a failure in the ACK reception after a timeout is interpreted as an indication that the route is either broken, compromised, or includes selfish nodes. For each route the routing protocol maintains a rating

reflecting the route's reliability, which is updated each time a piece of data (a set of data packets) is transmitted across the route as follows: It is increased for each successful reception (when the source receives the ACK of that piece), and decreased for each failed piece (when a timeout expires without receiving an ACK). When the path rating of a given route decreases below a defined threshold, assumed to be high enough to overcome the losses due to collisions, this route will not be used any more. Moreover, the routing protocol may rely on this rating as a metric and choose the most reliable routes. The ACKs must be signed to ensure no-repudiation, otherwise a selfish node may misbehave by not forwarding packets and sending back a falsified ACK to the source without being detected. Note that it is beneficial to a selfish node to perform like this, since an ACK costs much less than a piece of data packets. The signature of the ACKs requires an end-to-end security association between the source and the destination. The major problem of this technique is the lack of the misbehaving node detection. This technique may detect routes containing misbehaving or malicious nodes, and those which are broken, but without any further information regarding the node causing the packet lost. However, this technique helps to avoid sending packets through unreliable routes, and it can be combined with other more sophisticated techniques. It is used in [Awerbuch'02, Papadimitratos'03] along with another technique, namely data dispersion and probing which will be presented later in section (c) of the preventive solutions in this chapter. Note that this mechanism is also used in [Conti'05], where the authors propose a cross-layer mechanism that exploits TCP ACKs instead of adding explicit ACKs at the network layer, which reduces the overhead. This mechanism, however, is not combined with any detective technique in this solution, since the latter aims only at avoiding unreliable routes.

**2) Watchdog**

The first paper addressed the problem of nodes misbehaviour in MANET is [Marti'00]. The authors define a watchdog concept, which is a basic technique on which many further solutions rely. It aims at detecting misbehaving nodes that do not forward packets, by monitoring neighbours in the promiscuous mode. Suppose node S sends packets to D using a route including (possibly amongst others) respectively

three intermediate nodes: A, B, and C. When A transmits a packet to B to forward to C, A can check whether B forwards each packet by analysing packets it overhears during a given timeout. If A overhears a packet it is monitoring during the fixed timeout then it validates its forwarding. Otherwise it raises a rating regarding B, and will judge that B is misbehaving and notify S as soon as the rate exceeds a given threshold. This monitoring is generalised for each pair of hops in the source route. The solution also includes the pathrater component that selects routes based on the link reliability knowledge. The watchdog is able to detect misbehaving nodes in many cases, and requires no overhead when no node misbehaves. It allows monitoring all packets regardless whether they are directed or broadcast. Nonetheless, the watchdog fails to detect the packet loss due to collisions, partial collusion, and power control employment. After a collision at C, B could circumvent retransmitting the packet without being detected by A. B could also circumvent the watchdog by partially dropping packets, viz. at low rate than the configured accusation threshold. The watchdog fails when two successive nodes collude to conceal the misbehaviour of each other, that is, B could collude with C and do not report to A when C misbehaves. Furthermore, the watchdog technique may cause false detections when the configured threshold fails, and especially when the monitored node uses the power control technique [Doshi'02b, Doshi'02a] to preserve its power. By using the power control technique, nodes in MANET can preserve their power, by only transmitting packets from one node to another using controlled power according to the distance separating them from each other. For example, when C is closer to B than A and B transmits packets using a controlled power according to the distance separating it from C, A could not overhear B's forwarding and may accuse it wrongly.

The power control technique has been used by many routing protocols proposed after the watchdog's proposal in the field of power consumption optimization, such as [Doshi'02b, Doshi'02a, Krunz'04, Jung'05, Djenouri'06a]. Another serious problem with this solution is that it does not punish the detected misbehaving nodes. Upon the detection of misbehaviour, the detector informs the source node, thereby the rating regarding the misbehaving is updated. Despite this rating update ensures that

transmissions through the misbehaving node is avoided, no measure is taken against this node.

**3) Activity-Based Overhearing (ABO)**

In [Kargl'04] the authors propose the termed Activity-Based Overhearing, which is a generalisation of the watchdog. In this technique, a node constantly monitors in the promiscuous mode the traffic activity of all its neighbours, and oversees the forwarding of each packet whose next forwarder is also in its neighbourhood. This can increase the number of observations and improve the watchdog efficiency. It also mitigates the collusion problem. Nevertheless, this technique suffers from all the other problems of the watchdog, especially the one related to the power control technique as it relies on the promiscuous mode monitoring.

**4) Two-Hop ACK**

The authors in [Djenouri'05] propose a monitoring approach based on feedback called two-hop ACK. In the context of three aligned nodes, A, B, and C, such that A monitors B's forwarding to C, node C acknowledges packets sent from A by sending this latter via B a special ACK that travels two hops. Node B could, however, escape from the monitoring without being detected by simply sending A a falsified two-hop ACK. Note that performing in this way is power economic for B, since sending a short packet like an ACK consumes less energy than sending a data packet. To avoid this vulnerability, the authors use an asymmetric cryptography based strategy, and suggest that A generates a random number and encrypts it using C's public key, then A validates B's forwarding if and only if it receives later the random number it generated with the two-hop ACK. Otherwise, it notices packet dropping for B after a timeout. This random number received at A is ciphered by C with A's public key, C does so (ciphers the random number with A's public key) after deciphering the number with its private key. This way B could not falsify a valid two hop ACK, unless it gets or breaks C's private key. Like the watchdog, A accuses B as soon as the number of two-hop ACK detected dropped exceeds a given threshold. Since the validation at A is related to C's reception and not only to B's forwarding, the solution is independent of the power control usage, thus solves the watchdog's problems related to this issue. Unlike the watchdog, the two-hop ACK ensures that

after a collision at C, B could not escape from retransmitting the packet without being detected. The major drawback of this solution is its communication overhead, since a two-hop ACK is required for each data packet on each couple of hops. Although, the problem related to the overhead has been treated by the authors in their more recent work [Djenouri'06b] by the so-called Random Two-Hop ACK protocol, but still relatively high. In the random two-hop ACK protocol, instead of asking an ACK for each packet, the monitor node (node A) does this randomly with a probability continuously updated according to the behaviour of the monitored node (node B), in such a way to give more trust (low probability) to well behaving nodes. The Random Two-Hop ACK protocol is not independent of the routing protocol and operates as an extension of DSR.

**5) Probing**

Previously we have seen that the end-to-end ACK approach allows to monitor routes and to detect unreliable ones containing misbehaving or failed nodes, but fails to detect the appropriate nodes responsible of the unreliability. All the other monitoring solutions, however, directly monitor nodes. The probing approach could be viewed as a combination of route and node monitoring. This approach incorporates commands into data packets to acknowledge them. These commands are called probes and intended for selected nodes. Probes are launched when a route that contains a misbehaving node is detected (but not the ID of that node). [Awerbuch'02] was the first to use this mechanism. The protocol is based on the end-to-end feedback to monitor routes, thus requires the destination to return an acknowledgment ACK to the source for every successfully received data packet. The source keeps track of the number of recent losses (ACKs not received over a window of recent packets). If the number of recent losses exceeds the acceptable threshold, the protocol registers a fault between the source and the destination and starts a dichotomic search [Ferré'05] on the path, in order to identify the faulty link. The end-to-end ACK employed could be considered as the route monitoring phase, and the dichotomic search as the node monitoring on suspicious routes. The source controls the search by specifying a list of intermediate nodes on the future data packets. Each node in the list, in addition to the destination, must send an ACK for the packet. These nodes are called probed

nodes. The list of probes defines a set of non-overlapping intervals that cover the whole path, where each interval covers the sub-path between the two consecutive probes that form its endpoints. When a failure is detected on an interval, the interval is divided into two by inserting a new probe. This new probe is added to the list of probes appended to future data packets. The process of sub-division continues until a fault is detected on an interval that corresponds to a single link. This solution suffers from many drawbacks. In addition to the high cost of the communication overhead, there is no reliable detection of the dropper. A selfish node could analyse each packet it receives before deciding either to forward this packet or not. When it gets a probe packet it would notice that a probing is under way, and would consequently choose to cooperate and forward packets for a limited time, until the probe is over.

In [Kargl'04] the authors propose an enhanced probing approach called iterative probing. It defers from the previous solution in the fact that each command is addressed to one node instead of a set of nodes. Therefore, the command contains one encrypted node ID added to a special field in data packets. If a data packet includes no probing command then the field will contain a random number, such that a recipient cannot distinguishe data packets including probing from regular data packets, unless it is the destination of the probing command. The solution suffers from the problem of high overhead, for an H hops route, O(H) ACK transmissions is required for the first phase and O(log(H)) ACK transmissions to detect a misbehaving node. Overall, O(H + log(H)) is the overhead communication complexity of the solution when a misbehaving appears. This solution is also unreliable. It allows to detect the link containing the selfish node but cannot distinguish which of the two nodes forming the link is actually the misbehaving one, since there is no knowledge of the selfish node behaviour upon the reception of a probing (either it sends back the ACK or not). To mitigate this problem Kargl et al. [Kargl'04] proposed the unambiguous probing. The principle of this mechanism is simple and can be summarised as follows: Assume after an iterative probing a link (Xi,Xi+1) will be detected. To determine which one of the two suspicious nodes is guilty (selfish), the source node asks the node Xi−1 to check if it can overhear the forwarding of Xi. If so then Xi+1 is the guilty, otherwise the guilty is Xi. This

mechanism (unambiguous probing) suffers from the watchdog's problems, as it relies on the promiscuous at the predecessor of the suspicious link. Note that like two-hop ACK, probing was proposed and is applicable only to directed packets.

### B. Reputation-based

The reputation of a given community member can be defined as the amount of trust granted by the other members regarding its well-behaviour on a given function, according to their experience with it. Members that helpfully contribute to the community life get good reputation among community's members, while others who refuse to cooperate are badly reputed and gradually excluded from the community. In our context, the reputation of a node is the trustworthiness the other ones grant to it regarding its cooperation and participation in forwarding packets. This definition is large such that including both solutions that evaluate nodes' reputation by real values or Boolean values (well-behaving vs. misbehaving), provided that they punish bad reputation nodes.

Reactive reputation-based solutions are more elaborate than the previous monitoring-based solutions, and deal with the post-detection issues. Still, to detect selfish nodes they simply incorporate approaches proposed by those basic monitoring solutions. Each node keeps track of each other's reputation according to the behaviour it observes, and the reputation information may be exchanged between nodes to help each other inferring the accurate values. There is a trade-off between efficiency in using available information and robustness against mis-information. If ratings made by others are naively considered, the reputation system can be vulnerable to false accusations or false praise. However, if only one's own experience is considered, the potential of learning from experiences made by others goes unused, which decreases the efficiency. In the following, we present four solutions based on this general principle of reputation.

#### 1) Signed Token

In [Yang'02] the authors describe a unified network layer solution, based on the approach of mutually according admission in neighbourhood using signed tokens. It

aims at protecting both the routing and the data forwarding. Threshold cryptography-based signature [Shamir'79] and the watchdog technique [Marti'00] are at the core of this solution. The solution is structured around four closely interacted components: i) neighbour verification that describes how to verify whether each node in the network is well-behaving or selfish, ii) security enhanced routing protocol which enhances AODV [Perkins'99, Perkins'03] and extends to the termed AODV-S that explicitly incorporates the security information in routing, iii) neighbour monitoring that is based on the watchdog to describe how to monitor the behaviour of each node in the network, and how to detect packet droppers, iv) and finally the v)intrusion reaction which describes how to alert the network and isolate the misbehaving, and serves as a bridge between neighbour verification and neighbour monitoring. Nodes in a neighbourhood mutually accord participation admissions, and nodes without up-to-date admissions are excluded from any network service. Each node has a token issued by its local neighbours allowing it to participate in the network operations, which implements the concept of participation admission. The token has a period of expiration, whose value depends on how long the holder node has been behaving well (its reputation). This latter renews (updates) the token before its expiration. Nodes in a neighbourhood collaboratively monitor each other to detect any misbehaviour.

This solution employs asymmetric cryptography. There is a global key pair SK/PK (Secret Key and Public Key). Each token carried by a node is signed with SK and broadcast periodically in the hello message to ask for a new validation. Note that the solution uses a hello protocol. PK is known by all nodes, but none has the SK. Indeed, each node has a partial key, which is as a part of SK and participates by providing a partial signature of order K, thereby K different partial signatures are sufficient to provide the right signature. In other words, SK is divided among nodes in such a way that K different signatures with K different partial keys are necessary and sufficient to make a signature equivalent to that made by SK. This technique is called polynomial secret sharing [Shamir'79]. To decide whether to provide a partial signed token for the requestor or not, the requestor's historical behaviour is considered, which is drawn according to information collected using the promiscuous

monitoring and detections of neighbours as well. Once a node is detected as selfish the detector informs its neighbours, and the selfish is isolated as soon as K different nodes detect it. Isolating a node in a neighbourhood is achieved by not providing it with tokens. Although the authors do not evoke the notion of reputation, we categorize this solution in the reputation-based class since each node is granted or denied services in its neighbourhood according to its past behaviour. The reputation value of each node could be simply considered Boolean, i.e. well behaving or selfish. Therefore, well-behaving nodes will be served and granted tokens, while misbehaving ones will be isolated. Since the detected misbehaving are isolated and excluded from any network's service, the lack of a punishment mechanism against detected misbehaving nodes problem of the previous basic solutions is resolved. However, this solution has many disadvantages. First, all the watchdog's problems described previously remain untreated, since the neighbour monitoring component completely relies on it. The second disadvantage of this solution is that it prevents a node which has less than K neighbours to communicate, and poses a critical issue on the choice of the parameter (threshold) K for the sharing of the secret key. The choice of low K weakens the key, whereas the choice of high values requires high connectivity, which is not always ensured in MANET.

**2) CORE**

Michiardi and Molva [Michiardi'02a, Michiardi'02b] suggest a generic reputation-based mechanism termed CORE, supposed to be easily integrated with any network function. Unlike the previous solution this one gives more precise definitions to the notion of reputation, and defines three types of reputations: i) subjective reputation that is calculated directly from a node observations, and gives more relevance to the past observations in order to minimize the influence of sporadic misbehaviour in recent observations, ii) indirect reputation, which is calculated based on the information (observations) provided from other nodes, and iii) functional reputation that combines the subjective and indirect reputation. Each node maintains the three reputations for each other in a reputation table that is updated in two different situations; during the request phase of a given function, and during the reply phase corresponding to the result of the function execution. In the first phase, only

subjective reputation related to misbehaviour is updated (relying on negative information provided from the monitor component). Whereas, in the second phase only indirect reputations are updated positively. That is, a reply message containing a list of all the entities that correctly behaved is supposed to be transmitted back to the source node at the end of the function execution, so that the indirect reputations of these well-behaving nodes are increased.

CORE is implemented with DSR, and uses the watchdog for monitoring and collecting direct observations, thus both directed and broadcasted packets could be monitored. It can be applied to packet forwarding function, both on data and route request packets. For the route discovery function, the aim is to detect misbehaving nodes that do not participate in this function and do not forward route request packets. During the request phase of the route discovery, the negative rating factor of the next provider may be observed by the requestor's watchdog, like in [Marti'00], while the identity of the nodes that participate in the function are reported to the initiator during the reply phase. The routing service will be denied to route requests issued from nodes classified as misbehaving, i.e. nodes whose functional reputation values become negative (< 0). Similarly, the CORE scheme can be used to monitor the data packet forwarding function during the first step (negative rating observation). But as opposed to the route discovery function, the data packet forwarding function does not include separate operations that can be qualified as request and reply phases, which harden the indirect reputation updates. However, the authors propose to add end-to-end ACKs, the transfer of which can be considered as the reply phase.

The signed token mechanism [Yang'02] problem in terms of preventing nodes with less than K neighbours to communicate described previously, does not exist in this solution. Also, in contrast to the previous solution nodes' observations are propagated beyond neighbourhoods. However, only the positive observations (of well-behaving) are propagated but not the negative ones. The purpose is to provide robustness for the solution and prevent the vulnerability of rumours propagation which can cause DOS (Denial Of Service) attacks. This reduces the potential of

learning from observations made by others and can decrease the efficiency of misbehaviour detections in the network. Contrary to the previous solution where the isolation is performed collectively by all nodes in neighbourhoods, the isolation in CORE is performed unilaterally by each node basing merely on its own view of nodes' behaviour. This could represent a potential threat of possible false accusations, as when an isolator does not forward packets for another node unilaterally isolated, other neighbouring nodes (that are not isolating the appropriate node) would consider this as illegal behaviour. Further, the solution does not allow redemption after detection, as when a node is excluded by another node it will not be asked to execute the service for this detector and will never be able to redeem and increase its reputation with it. If the nodes exchange their own experiences with each other (their views of reputations and not only observations), such a redemption would be possible. Moreover, all the watchdog's drawbacks related to detections are present in this solution, since the solution relies on the watchdog mechanism for monitoring.

**3) CONFIDANT**

CONFIDANT is another reputation-based solution, proposed in [Buchegger'02b, Buchegger'02a] [Buchegger'03]. It consists of four components present in each node. The first one is the monitor which is very similar to the watchdog [Marti'00]. It registers the deviations from the normal behaviour and calls the reputation system as soon as a given misbehaviour occurs. The trust manager is the second component that deals with the incoming and the outgoing ALARM messages. ALARM messages are sent by the trust manager of a node to warn others of misbehaving nodes, i.e. the protocol is based on negative information propagation. Outgoing ALARMS are generated by the node itself according to its experience observations, or after a misbehaviour report reception. The recipients of these ALARM messages are called friends, which are considered to be configured on a user-to-user basis. Incoming alarms, originate from either outside friends or other nodes, are checked for trustworthiness before triggering a reaction. The trust manager uses a filtering of incoming ALARM messages according to the trust level of the reporting node. To

define trust levels, a general mechanism similar to the trust management used in PGP for key validation and certification has been proposed.

In their recent work [Buchegger'04], the authors propose a modified Bayesian mechanism that gives less importance to past observations than recent ones, and allows redemption. The third CONFIDANT's component is the Reputation System that manages the node's view on reputations of the others. Each node reputation is represented by a rating that is changed according to a rate function, assigning different weights to the type of behaviour detection, i.e. the greatest weight for own experience, a smaller weight for observations in the neighbourhood, and the smallest one to reported experience. The rationale for this weighting scheme is that nodes trust their own experiences and observations more than those of other nodes. Once the rating of a node exceeds a configured threshold, the path manager is called for action. This latter is the last component, it is responsible for punishing the misbehaving nodes by not relaying any packet for them, as well as deleting paths containing misbehaving nodes and path re-ranking according to nodes trustworthiness.

Unlike the previous reputation-based solution (CORE), with CONFIDANT reliable negative information are propagated beyond the neighbourhood. To mitigate the vulnerability to DoS (Denial of Service) attacks by propagating rumours, the trust manager is proposed along with the rate function that assigns different weights to the types of behaviour detections in such a way to give more importance to local observations when computing the reputation rating. Moreover, the path manager component clarifies punishments against detected misbehaving. The simulation results [Buchegger'02b] show a significant improvement in term of throughput compared to the standard DSR (with which CONFIDANT has been implemented). Nevertheless, like the previous solution the isolation is performed independently by the path manager of each node. Recall that this could represent a potential threat of possible false accusations, as when an isolator does not forward packets for another node unilaterally isolated, other neighbouring nodes would observe that and consider it as illegal behaviour when they are not isolating the appropriate requestor. Also, all

the watchdog's drawbacks presented previously remain untreated in this solution, since the monitor component fully relies on this technique.

**4) OCEAN**

Bansal and Baker propose OCEAN [Bansal'03], a scheme for robust packet-forwarding. OCEAN, similarly to CORE and CONFEDENT schemes, is based on nodes' observations. In contrast to previous mechanisms, no rating is exchanged and every node relies on its own information, so the trust management is avoided. The rating is based on a counter that counts the positive and the negative steps a node performs and based on a faulty threshold, the node is added to a faulty list. In the method for route selection, a DSR node appends an avoid list to every generated RREQ and a RREP based on this list. A second-chance mechanism is provided to give nodes that were previously considered misbehaving another opportunity to operate. OCEAN suffers from similar drawbacks as CONFEDENT and CORE.

**5) SORI**

The Secure and Objective Reputation-Based Incentive (SORI) scheme was proposed by [He'04]. It targets the non-forwarding misbehaviour type and uses a watchdog-like mechanism for monitoring. The reputation system keeps counting of the packets forwarded both by and for neighbouring nodes. Reputation ratings consist of the ratio of these counts, taking into account the confidence in the rating proportional to the number of packets requested for forwarding. Nodes propagate reputation ratings locally; this second-hand information is weighted by credibility, which is derived from the ratio above. The response is given by packet dropping with a probability determined by reputation. SORI additionally employs hash-chain-based authentication for propagated reputation ratings. SORI mechanism is designed to treat generously the nodes that do not intentionally drop packets. It also has a complementary security mechanism which proposed to deal with a node that uses the following attacks: 1) impersonation of an adjacent node's id, ranked with a good reputation, in order to send more packets, and, 2) impersonating a distant node's id, ranked with a good reputation, to broadcast fake observation information in order to boost its reputation. This mechanism is based on a one-way hash chain and Message

Authentication Codes (MACs). SORI takes no countermeasures to prevent collusion. Finally, it suffers from the Watchdog drawbacks.

**6) Friends and Foes**

Contrary to CONFIDANT, friends and foes [Miranda'03] gives as much importance to the past observations as to the present ones. Thus, it uses a long-lived memory. In this solution nodes are permitted to publicly claim that they are unwilling to forward packets to some nodes, as each node maintains basically three sets: a set of friends to which it is willing to provide services, a set of foes to which it is unwilling to provide services, and finally a set of nodes known to act as if it is their foe (they do not provide services packets for it) named set of selfish. These three sets are periodically broadcast in the neighbourhood. Each node also maintains other variables for its neighbours, especially its view of their friends and foes, that are updated according to its experience and to the messages it receives periodically from its neighbours. When a node is asked to forward a packet it does so only when the asker is a friend, and count accordingly a credit for this friend. Also, every node chooses routes such that the next forwarder is its friend, then monitors the forwarding using the watchdog technique. It deletes a credit for the monitored node if this latter is perceived to correctly forward the packet, and puts it in the selfish set as soon as the number of packets it drops exceeds a given threshold. The solution allows redemption and permits a selfish node to be reintegrated by broadcasting a special packet (SelfState) acknowledging that it has behaved selfishly with the appropriate nodes. To prevent abusing this mechanism, the selfish node is first charged with penalties; it must broadcast two SelfState packets to consume additional energy, and the maximum value of its credit (the maximum number of packets it can send without providing forwarding services) is decreased by all neighbours. In addition to data packets, the authors propose the use of this solution to secure DSR control packets against selfish dropping.

This solution defines a robust method of redemption that allows selfish nodes reintegration, while preventing these latter from abusing nodes' tolerance. Nonetheless, it suffers from some problems. First, it has all the watchdog problems

on which it relies for monitoring. The second problem is related to the overhead. The authors argued that the solution does not cause significant overhead because control packets of each node are merely sent in its neighbourhood. However, these packets are broadcast periodically, which could be significant in networks with high connectivity. Moreover, because of the fact that each node keeps only information about its current neighbours, and subsequently if the information of nodes leaving its neighbourhood are arisen, then a mobile selfish node will take advantages and can easily circumvent without being detected. Finally, the solution is integrated with DSR, and is used to secure DSR's control packets from dropping. However, a basic principle of the solution is that each forwarder chooses the next one among its friends. Therefore, routing is made hop-by-hop and the solution is not applicable to a source routing protocol as DSR. Indeed, any reactive hop-by-hop routing protocol could be integrated with this solution, such as AODV.

**7) Context-Aware Detection**

With this mechanism by Paul and Westhoff [Paul'02], accusations of nodes are related to the context of a unique route discovery process and to a stipulated time period. To detect attacks in the route discovery phase source and destination use un-keyed hash chains and promiscuous mode of link layer to observe malicious acts of neighbourhood nodes. Observers of the attacker independently communicate their accusation to the source node. The source node executes an inference scheme based on majority voting to rate an accused. Source node can later on advertise these rating along with adequate proofs to trusted nodes. Such ratings are used by the knowledgeable nodes to deny any future service to the attackers. In contrast to watchdog and pathrater, several types of misbehaviour are detected. The decision of how to treat nodes in the future, the response, is based on accusations of others, whereby a number of accusations pointing to a single attack, approximate knowledge of the topology, and context-aware inference enable a node to rate an accused node. Accusations are sent to the source, which infers based on majority voting and can inform trusted nodes.

**8) Dependency Graphs approach**

Dependency Graphs approach is proposed in [Badonnel'05a, Badonnel'05b] to estimate Ad-hoc Node Influence and to detect the most influential routing nodes in the ad-hoc network. This method can also be applied to detect misbehaving nodes performing for instance flooding in the network. However this work lacks of clarity and verification especially in terms of the detection of the flooding node. The method used in [Badonnel'05b] is inspired by the work described by Mark Burgess in [Burgess'04] related to the estimation of nodes influence in fixed wired networks. In Burgess's method a simple starting definition of well-connected could be 'of high degree', i.e. count the neighbours. This method is inspired from research in social sciences, where social relationships are studied in order to detect individuals capable to influence important parts of a social network.

### 3.1.2 Preventive Solutions

So far, we have presented reactive solutions that aim at detecting selfish misbehaviour on packet forwarding when it appears in the network. Another class of solutions includes approaches that proactively try to mitigate the misbehaviour or its effects, either by motivating nodes to cooperate or by taking measures to prevent packets from being dropped before sending them. This is helpful to reduce the problem but does not eliminate it completely. Thus, a reactive solution which detects such misbehaviour remains essential. In this section we present three approaches classified as preventive: economic-based solutions, data dispersal, and game theory based solutions.

**1) Economic-based**

In the following we present two economic-based solutions, inspired by some economic principles, which they project on to the forwarding service in MANET.

**a) Nuglets**

Buttyan and Hubaux [Buttyan'01] propose an economic-based approach stimulating nodes to cooperate for packet forwarding in MANET, which they model and analyse in [Buttyan'03]. They introduce what they call virtual currency or nuglets, along with

mechanisms for charging the service usage. The basic idea of this technique is that nodes which utilize a service must pay for it (in nuglets) to the provider ones. This makes nuglets essential for utilising the network, and renders each node interested in increasing its stock of nuglets by providing services for other nodes. Besides stimulating for the provision of services, this mechanism can also force nodes to make a moderate usage of the network services, since they are charged. Nuglets are represented by counters at nodes, each one's value corresponds to the wealth of the holder. In order to prevent a node from illegitimately increasing its own counter, this latter is maintained by a trusted and tamper-resistant hardware module, called security module. Only this module can directly perform operations on the counter. Nuglets loaded in a packet are protected from illegitimate modification and detachment from its original packet by cryptographic mechanisms. The physical and data link layers (where the security module is built) are assumed to be robustly protected, such that users cannot modify them. Furthermore, the neighbourhood of a node is assumed not to change very fast, so as to make it feasible for a node to keep track of its neighbours by running a hello protocol. Besides discovering its neighbours, the security module uses the hello protocol (like the signed token described before) to establish and maintain security associations with the security modules of the neighbouring nodes.

As for packet forwarding charging, the authors suggest three models: Packet Pursue Model (PPM), Packet Trade Model (PTM), and a hybrid one. In the first model the source is charged. It estimates the required nuglets on each hop and puts the total number of estimated nuglets in the packet, then each forwarder acquires the required nuglets from the packet. The required nuglets charged by a forwarder may depend on many things, such as the amount of energy used for the forwarding operation, the current battery status of the forwarder, and its current nuglets number. If a packet has not enough nuglets to be forwarded then it is discarded. The advantage of this model is that it may deter nodes from sending useless data and overloading the network. However, the drawback is that it is difficult to estimate the total number of nuglets required for a packet to reach to a given destination. If the source under-estimates this number then the packet will be discarded and the source loses its investment in

this packet, whereas an overestimation causes a wasting of the precious nuglets. On the other hand, in the PTM approach the packet does not carry nuglets, but it is traded for nuglets by intermediate nodes on each hop. Each intermediary buys it from the previous one for some nuglets (except the first intermediary that receives the packet for free from the source), and sells it to the next one (or to the destination) for more nuglets. This way, each intermediary that provides a service by forwarding the packet increases its number of nuglets, and the total cost of forwarding the packet is covered by the packet's destination. In contrast to the previous model, in this one the source does not need to know in advance the number of nuglets required to deliver a packet. Furthermore, letting the destination pay for the packet forwarding makes this approach applicable in the case of multicast packets. However, a serious disadvantage is that this approach does not deter nodes from overloading the network. Another disadvantage is of overhead, since a price negotiation is required on each hop for each packet. The two models can be combined in the following way: the source loads the packet with some nuglets before sending it, the packet is handled according to the PPM until it runs out of nuglets, then it is handled according to the PTM until the destination buys it. This hybrid model gets over the packet loss problem of PPM.

Nuglets is a new economic-based approach that motivates and obliges nodes to cooperate and forward packets for each other, because when a node behaves selfishly it will be unable to send its own packets. Moreover, this solution allows the nodes redemption, since a node which is unable to send its own packets because it runs out of nuglets is not excluded from being asked to participate in the data forwarding service and earning nuglets. But this approach suffers from some disadvantages. If a well-behaved node is not asked to route enough packets then it cannot send enough packets, and will be unfairly excluded. A node may be excluded from the routing process because of its position (it has few neighbours and belongs to just few routes) or because of the communication patterns of its neighbours (they have no communications with nodes to which it has routes). Furthermore, this technique does not prevent a node with enough nuglets from misbehaving, especially if it has not enough packets to send. Another issue related to this technique is that its robustness

totally relies on tamper-resistant hardware, but this is known to be a difficult problem.

**b) SPRITE**

[Zhong'03] propose another economic-based solution termed SPRITE, in which each node has a virtual credit maintained and continuously updated by a central authority called Credit Clearance Service (CCS). The principle is simple; when a node sends its own messages (as a source) it loses credits, and gains credits when it forwards messages for other nodes. To implement this each forwarder is assumed reporting to the CCS for each message it forwards a receipt, a signed small message derived from the original one. This reporting is assumed to be performed whenever the node switches to a fast connection with a backup power. When the CCS gets reports related to a receipt, it charges the source of the message and compensates the intermediate nodes. The credit that an intermediary receives depends on whether its forwarding has been successful, and whether the message has reached its final destination. Forwarding is considered successful if the next node on the route reports a valid receipt. Signing receipts prevents nodes from forging them, so none can report a receipt without really receiving a message. However, as soon as a node receives a message, it can easily report the receipt without forwarding the message.

The compensation strategy takes this problem into account, and prevents reporters that provide receipts of messages which do not reach the finale destination (messages not reported by the destination) from earning credits. The authors provide a modelling and a formal proof of the solution, which shows that the solution is cheat-proof (under a set of conditions). That is, truth telling (reporting receipt only when forwarding a message, and not denying any forwarding) is the optimal strategy for every node. The proof also illustrates that the solution is collusion-resistant. Further, the solution was extended with little modifications to broadcast control packets (like route request of the routing protocol), for which the CCS computes a tree based on receipts it receives before updating credits. This way, redundancy is avoided.

Like Nuglets, SPRITE is an economic-based strategy that motivates nodes to collaborate. However, the major advantage of SPRITE is that it does not require any tamper-resistant hardware. Also, virtual money in this solution are considered as credits and are not held in packets, contrary to Nuglets. Consequently, the strategy of charging the source is efficient for SPRITE, since the problem of packet dropping due to virtual money lack presents with the packet pursue model of Nuglets does not exist here. Remember that the source-charging strategy has the advantage of preventing nodes from sending useless data that overload the network, and makes them rational when using the network services. Further, the proposed compensation strategy overcomes collusion (on falsely reporting receipt), providing that the destination well-behaves. Nevertheless, the elimination of the tamper-resistant dependency was ensured by using a central authority (CCS) that manages credits, which makes the solution centralized, and thus introduces another drawback. Distributing the CCS is mandatory for this solution to be applicable in MANET, basically featured by the total decentralization. Another disadvantage of this solution is that it assumes the cost of reporting a receipt to be negligible, and requires the reporting to be performed when the node switches to a fast connection and gets backup power, which is not always possible in MANET.

**c) Data Dispersal**

This scheme [Papadimitratos'03] is based on Rabin's algorithm [Rabin'89] and takes advantage of the existence of multiple routes from a source to a destination, to increase the reliability when transmitting packets. It consists of adding redundancy to the message to be sent, then the message and the redundancy are divided into a number of pieces and dispersed on the available routes, so that even a partial reception can lead to the successful reconstruction of the message at the receiver. Note that node-disjoint routes ensure more efficiency. This technique can overcome partial packets loss that can occur due to misbehaviour on some used routes. Also this approach is based on a mathematical framework. The redundancy factor is a crucial parameter for this solution. Increasing this ratio ensures more reliability, since few pieces among the overall sent pieces would be required to reconstruct B, but high values of this ratio cause significant overhead. On the other hand decreasing the

redundancy factor reduces the overhead, but gives less reliability. Therefore, the choice of this parameter is a trade-off issue. It should strike a balance between reliability and overhead. Even though this mechanism does not prevent nodes from misbehaving and does not motivate nodes to cooperate, unlike the previous ones, it is helpful to reduce the selfish misbehaviour effects on the communication reliability, and can be combined with a reactive solution. In [Papadimitratos'03] the authors propose SMTP, a solution that uses this mechanism. However, this solution has the end-to-end feedback technique drawbacks presented previously, since it relies on it.

**2) Game Theory Based**

In this approach the forwarding process is viewed as a game, where nodes have to continually decide whether to forward or not to forward packets. The purpose of this approach consists of defining strategies to ensure fairness to all nodes. Since users may be selfish, there is no guarantee that they will follow a particular strategy unless they are convinced that they cannot do better by following some other strategy. In the game theory terms a strategy which constitutes a Nash equilibrium [Myerson'91] needs to be identified. Nash equilibrium can be defined as a strategy profile having the property that no player can benefit from unilaterally deviating from the strategy [Srinivasan'03]. In other words it is a feature which ensures that if a cheating player tries to deviate from the strategy whereas all the others follow it, the cheater cannot receive more benefits than the others.

Some solutions based on this approach have been proposed, such as [Srinivasan'03, Wang'06]. For instance, in [Srinivasan'03] nodes are distributed among classes according to their energy constraints and their expectation of lifetime. The source node asks intermediate ones to relay packets before sending them, then each node has to decide whether to accept or reject forwarding packets for this source. If one node refuses to forward packets then it returns a negative ACK back to the source, and consequently the session is blocked. Otherwise, the request is forwarded until reaching the last router (destination's predecessor) which sends a positive ACK back to the source. A node that has relayed much more traffic than the amount that has been relayed for it (according to a defined factor) refuses to participate in the session.

A node that has relayed more traffic than a defined amount also rejects the participation. In other cases, the node agrees to forward packets. It has been proved that the proposed algorithm leads to a Nash equilibrium. That is, if all nodes accurately execute the algorithm then any individual deviation from a node will not allow it to reach a greater throughput than the so-called Pareto optimal value, reached by all well-behaved nodes in the Nash equilibrium. This solution, as well as all the ones based on game theory, trusts the ACKs of intermediate nodes. Indeed, a selfish node may agree to participate in a session and to forward packets, in order to give impression that it executes accurately the protocol, but actually would not forward packets when it receives them. The approach needs to be combined with a reactive monitoring solution for resolving this problem.

## 3.2 Intrusion Detection System (IDS)

An intrusion may be defined as "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource" [Heady'90], or "any unauthorized or unwanted activity on a system or a network" [Lee'03]. An IDS may also be defined as "a system that tries to detect and alert on attempted intrusions into a system or a network" [Lee'03]. The history of security research has taught us a valuable lesson: no matter how many intrusion prevention measures are inserted in a network, there are always some weaknesses in the systems that one could exploit to break in [Zhang'03]. These weaknesses include design and programming errors and various social engineering penetration techniques as well. Hence, intrusion prevention measures (proactive solutions) cannot eliminate attacks, and they must be fortified with IDSs. An IDS presents a second wall of defence and is essential for any high-survivability network. There are mainly two classes of IDSs, Anomaly detection and Misuse detection explained below.

### 3.2.1 Anomaly detection

These IDSs consider activities that deviate significantly from the established normal usage profiles as anomalies, i.e., possible intrusions, where "normal" patterns are defined beforehand. The main advantage of anomaly detection is that it does not require prior knowledge of intrusions and can thus detect new intrusions. The main

disadvantage is that it might be unable to describe what the attack is and might have a high false positive rate. An example of this type of IDS in traditional networks is IDES [Lunt'92].

### 3.2.2 Misuse detection (signature-based)

These IDSs rely on the use of specifically known patterns of well known unauthorised behaviour and attacks to match and identify *known* intrusions. The main advantage of this technique is that it can accurately and efficiently detect instances of known attacks. Its main drawback is that it lacks the ability to detect the truly innovative (i.e., newly invented) attacks, whose patterns are unknown. IDIOT [Ilgun'95] and STAT [Kumar, S.'95] are examples of signature-based IDSs in traditional networks. Another classification of traditional IDSs is based on the type of audit data used. This class includes Network-based IDS and Host-based IDS. Network-based IDS normally runs at the gateway of a network, where it captures and examines packets that go through the network hardware interface. On the other hand, Host-based IDS relies on operating system audit data to monitor and analyse the events generated by programs or users on a host.

Recently some IDSs have been proposed [Anjum'03, Huang'03, Kachirski'03, Zhang'03, Liu, Y.'06, Karim'06] for MANET. Most of these IDSs are distributed, host-based, anomaly-based, and cooperative. The cooperation, however, may be fully and equally distributed among nodes, or it may be based on hierarchal node organization. The existing intrusion detection techniques mostly deal with nodes misbehaviour problem at the application layer but not cross other layers, and most of them based on trace analysis of historical data.

## 3.3 Summary

In this chapter we surveyed the literature and related works. We presented and discussed the existing solutions that aim at detecting misbehaviour on packet forwarding. These solutions classified into two main techniques, Reactive and Preventive. Reactive solutions aim at actively detecting the misbehaviour when it appears, while preventive ones try to either proactively prevent any misbehaviour by

motivating and forcing nodes to cooperate, or to take precautions to avoid packets from being lost before sending them. The reactive solutions divided into two main classes, monitoring and reputation-based solutions. The monitoring class includes basic approaches that focus on the monitoring phase and suggest techniques to control the forwarding process. The major drawback related to these monitoring solutions is the post-detection issues, i.e. punishment and selfish nodes knowledge exchange between nodes. Reputation solutions on the other hand are more detailed and principally deal with these issues. However, all the reputation-based solutions in this chapter used the Watchdog [Marti'00] for their monitoring approach; subsequently inherit all its drawbacks. We realize that designing a solution basing on a more reliable monitoring approach represents an open research topic. Preventive techniques proactively try to mitigate the misbehaviour or its effects, either by motivating nodes to cooperate or by taking measures to prevent packets from being dropped before sending them. We noted that the major drawback of these techniques is that it totally trusts nodes; thus it needs to be combined with a reactive monitoring technique. We also discussed Intrusion Detection System (IDS) and its two main classes Anomaly and Misuse detections. We pointed out IDS weaknesses that include design and programming errors as well as various social engineering penetration techniques.

This chapter pointed the main drawbacks of the existing work and issues that need to be addresses as:

- Most of the existing approaches have high cost in terms of communication overhead produced.
- Existing detection mechanisms are not independent of the routing protocol and operate as an extension of it. Many mechanisms work only with one particular routing protocol.
-

- Existing monitoring/detection solutions suffer from the post-detection drawback, in terms of punishment and selfish nodes knowledge exchange between nodes.

- Most of the existing mechanisms use monitoring approaches that depend on promiscuous monitoring, which has many drawbacks regarding the accuracy on detections, especially when employing the power control technique.

- Techniques using proactive approaches trust all nodes and do not prevent nodes from overloading the network, thus they can not work effectively alone and they require to be combined with a reactive monitoring technique.

- Most of the existing solutions are applicable only for a small MANET with limited number of nodes and as such scalability has not been addressed, especially when dealing with wireless sensors network which has a large number of nodes.

- Energy saving has not been considered properly, and as such many existing approaches have a high-energy consumption.

In the following Chapter we will present our new framework for misbehaviour detection and its different components. We will also discuses the aim and requirements of our new framework based on shortcomings in the related work.

# 4. SESSIONS-BASED MISBEHAVIOUR DETECTION FRAMEWORK (SMDF)

Having introduced existing misbehaviour detection mechanisms in MANET, we now present our new framework. In this chapter, we discuss our novel framework objectives, requirements and techniques that address the gaps in the related work. We will start with the overall aims and specific objectives of our work. Then we will identify specific requirements, issues and challenges important when designing an effective misbehaviour Detection Framework in stationary MANET and static wireless sensor network. Next, we create a structured overview on what kinds of attacks in Ad hoc networks our new framework will target. Finally, we present our new framework Sessions-based Misbehaviour Detection Framework (SMDF) and briefly describe its different components.

## 4.1 Aims and Objectives

In this section, we discuss the overall aims and specific objectives of our work. Our aims were born from the security emerging of the new security threats and attacks that targeting mobile ad hoc networks. The evolving of MANET and its widely advance applications such as wireless sensors network in today's IT and Telecommunication industries trigger our attentions to find effective solutions to such attacks. Security research in finding low cost and effective security mechanisms to detect attacks is still not enough in this particular evolving area. Much less progress has been made on providing new and efficient detection approaches.

The aim of our research is to provide a set of security mechanisms to overcome nodes misbehaviour problem in MANET. The major research objectives that we address in the problem area are:

- To develop novel detection mechanisms that can monitor and detect nodes misbehaviour in stationary MANET and wireless sensor network. These mechanisms will detect node misbehaviour with low cost in terms of reducing the amount of communication overhead. These mechanisms can be evaluated by comparing them to existing approaches to show how much overhead reduction it achieves. This can be carried out through simulation.

- To design a decision scheme to determine and judge whether nodes misbehaved deliberately or not. This scheme will be responsible for ensuring that nodes are not wrongly accused of misbehaviour. It will also allow node redemption before taking decision. Such mechanism can be enhanced from the existing mechanisms and be mathematically evaluated.

- To design an isolation scheme that can deal with misbehaving nodes that has been charged by the decision scheme. This isolation scheme will be able to punish misbehaving nodes by isolating them from the network, so they can not harm the network or attacking it again. This scheme can be implemented mathematically.

- To integrate the developed schemes into a framework that can efficiently monitor, detect, make decisions and isolate misbehaving nodes, to prevent security attacks such as Data Dropping attack and Black-hole attacks from targeting MANET and wireless sensor network.

## 4.2 Framework Requirements

The next stage is to identify requirements, issues and challenges when designing an effective misbehaviour Detection Framework in stationary MANET and static wireless sensor networks. Our framework requirements designed under the assumptions that MANET is stationary and that misbehaving nodes (selfish and/or malicious) are dropping only data packets and not control packet.

The requirements for the new system are specified as follows:

- The new framework should use little system resources at low cost to run and should not degrade the system performance by introducing significant communication overhead and high power consumption. The overhead reduction has to be significant in comparison with existing approaches. By achieving this, the new framework will have novel aspect as it can be efficient at very low cost in comparison with other existing approaches.

- The new framework should support cross-layer collaboration in order to reach valuable optimizations. This will require that the new framework being an active acknowledgment mechanism, which will ensure the correct delivery, and will take place only when data packets are sent contrary to most of the existing mechanisms.

- The new framework has to be reliable in terms of minimizing detections that are false positives or false negatives. If the new framework achieved an optimal lower value of false positives or false negatives among other existing approaches, then it will be considered as a novel achievement. This also will reflect that the new framework is fair in terms of not wrongly misjudged well-behaved nodes in the network.

- The new framework has to be precise and accurate in maximizing its true positives detection. If the new framework achieved the optimal higher rate of accuracy in comparison with existing approaches, then it will be consider as a

novel achievement. Such high accuracy will required that the new framework has to be capable of detecting node misbehaviour with partial dropping, where malicious nodes selectively forward some packets and drop others. Also capable of detecting dishonest nodes and nodes misbehaviour in the presence of collisions and most importantly it has to be capable of detecting node misbehaviour when limited transmission power is available or when using the power control technique which most of the existing approaches are not capable of detecting it.

- The new framework is not required to be part or an extension of particular routing protocol, but it can work independently with any routing protocol. If achieving such independency the framework will include novel characteristic in comparison with all of the current existing approaches that are not independent of the routing protocols.

- The new framework has to be flexible and has the capability of adding new components to it or removing existing component from it as necessary. This will add extra novelty to the new framework in that it can be updated with new components either from existing approaches or completely a new one. It will also allow the framework to be transparent in terms of integrating with other mechanisms as it is capable of accepting their components to be integrated to it.

- The new system has to be scalable to support large numbers of nodes to reflect stationary wireless sensors network. Most of the existing approaches support limited number of nodes mostly up to 50 nodes. Therefore by increasing this number the new framework will has the advantages of supporting larger number of nodes which will increase its suitability for different kinds of MANET including wireless sensors network.

- The new system has to use key management method to accurately enforce authentication, confidentiality and integrity by cryptography; requiring

distribution/exchange of encryption key information. So the message receiver must be able to determine the actual originator of message, and to verify the node's identity. The new system also has to perform availability by reaching all necessary recipient nodes.

- After detecting and deciding a node is selfish and/or malicious, the new system has to isolate such node from the network so they can not harm it again. Redemption should not be allowed after the decision stage but rather before it. By doing this we prevent the misbehaving nodes from lunching another malicious attack on the network.

### 4.2.1 Issues regarding Punishment and Reward Requirements

There are two ways to enforce a desired behaviour in the network [Yau'03]: punishing misbehaving nodes or encouraging well-behaving nodes. Commonly, the nodes are more sensitive to punishment than to rewards, so we focus on effective punishment more than on reward. Punishment of the misbehaving nodes (which do not forward packet properly) is done by dropping all their packets - both control and data packets. The more nodes that identify a misbehaving node and punish it, the more useful the punishment is. A question that arises is whether to accept rating information from such nodes, or just ignore it. Traffic of misbehaving nodes, which pass through intermediate good nodes that are not aware of the misbehaviour, is also an open issue that should be decided. An appropriate punishment would drop the misbehaving node's traffic, whether it is obtained directly or indirectly. Such a policy, however, may cause suspects in well behaving nodes. Punishing liars is another issue. It is reasonable to penalize nodes that do not report honestly, to encourage proper information distribution. However, it may discourage other nodes from reporting on misbehaving nodes that have not been detected yet. Punishment of liars is commonly implemented by ignoring their reports. It may also be enhanced further to packet dropping, but then the problem of incorrect suspicions arises again.

### 4.2.2 Types of Attacks

Before developing a security framework that prevents selfish or malicious nodes from harming the network, it is advisable to first create a structured overview on what kinds of attacks in Ad hoc networks the system will target. This way we can later verify what attacks are actually prevented by our security system and where there are still open problems.

The security system we are to propose is targeting the following attacks:

**1) Dropping Data Packets Attack**

Since packets follow multi-hop routes and pass through other nodes, a malicious or selfish node can participate in routing, include itself in routes, and drop all packets it gets to forward. To do this, the malicious node first attacks the routing protocol to gain participation in the routing, using one or more of the attacks presented previously. This attack is launched by both selfish and malicious nodes, and it has the same effects as the selfish misbehaviour.

**2) Black-hole Attack**

In this attack, a malicious node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. In flooding based protocol, if the malicious reply reaches the requesting node before the reply from the actual node, a forged route has been created. This malicious node simply drops data packets quietly, modify data content, replay, or flood data packets; they can also delay forwarding time-sensitive data packets selectively or inject junk packets. They can also choose whether to drop the packets to perform a denial-of-service attack or to use its place on the route as the first step in a man-in-the-middle attack. This attack launches by malicious nodes only.

## 4.3 Sessions-based Misbehaviour Detection Framework (SMDF)

Our solution to the Misbehaviour problems in MANET is a new Sessions-based Misbehaviour Detection framework (SMDF) [Fahad'07b]. It consists of three new

components integrated together to detect and deal with nodes Misbehaviour in MANET. The first and most important component of the framework is the novel Detection component. For this component we have developed a novel Sessions-based Misbehaviour Detection Protocol (SMDP) [Fahad'06, Fahad'07a, Fahad'07b]. The second component of the new framework is the Decision Component [Fahad'07b] which will judge whether the nodes misbehave intentionally or not. The third and final component of our framework is the Isolation component [Fahad'07a] which will penalize nodes who are judged to have misbehaved. Figure 4-1 above shows our framework SMDF and its components.

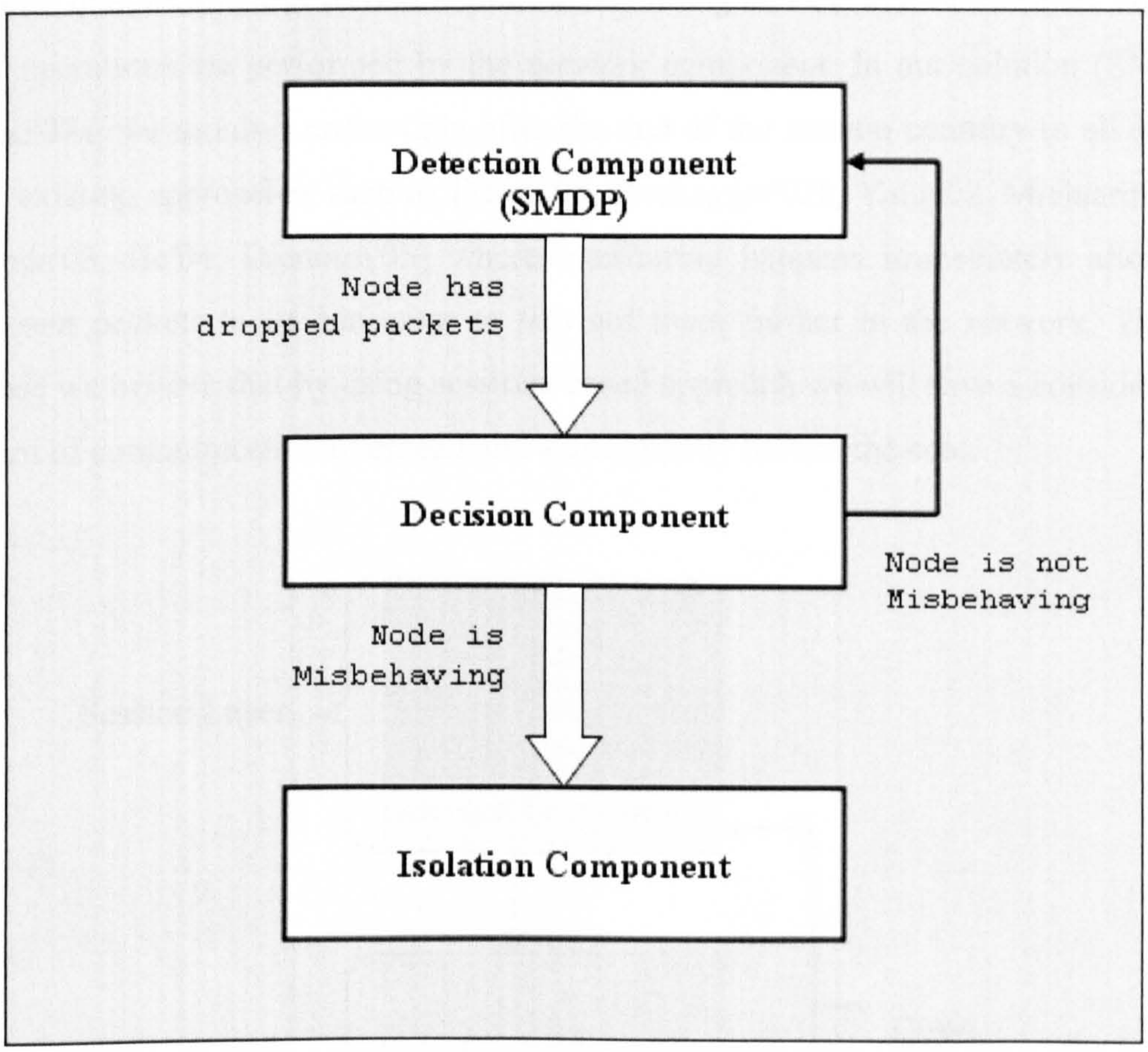



**Figure 4-1: SMDF Framework**

In our solution Sessions-based Misbehaviour Detection Framework (SMDF) [Fahad'07a] each node in the route session monitors all of its direct neighbours (i.e.

neighbours within a one hop communication), and checks whether they correctly forward packets. We define a session as the continuous traffic sent from the source node to the final destination node. The routing protocol has to be aware of the beginning and the end of each session. This has been done through cross-layer collaboration between the session layer and the network layer, shown in figure 4-2. Cross-layer is a paradigm in wireless network architecture design that takes into accounts the dependencies and interactions among layers, and supports optimisation across traditional layer boundaries [Conti'04]. In our framework it means the exchange of information between the session layer and the network layer. As a result, our protocol has two components, a session component and a network component. The first one informs the second about the beginning and the end of sessions. All the other operations are performed by the network component. In our solution (SMDF) [Fahad'07a] we monitor nodes only after the end of the session contrary to all of the other existing approaches such as [Marti'00, Buchegger'02b, Yang'02, Michiardi'02a, Miranda'03, He'04, Djenouri'05] where monitoring happens immediately after the node sent packets to its successor to forward them further in the network. This is because we believe that by using sessions based approach we will save a considerable amount of communication overhead and subsequently reduce the cost.

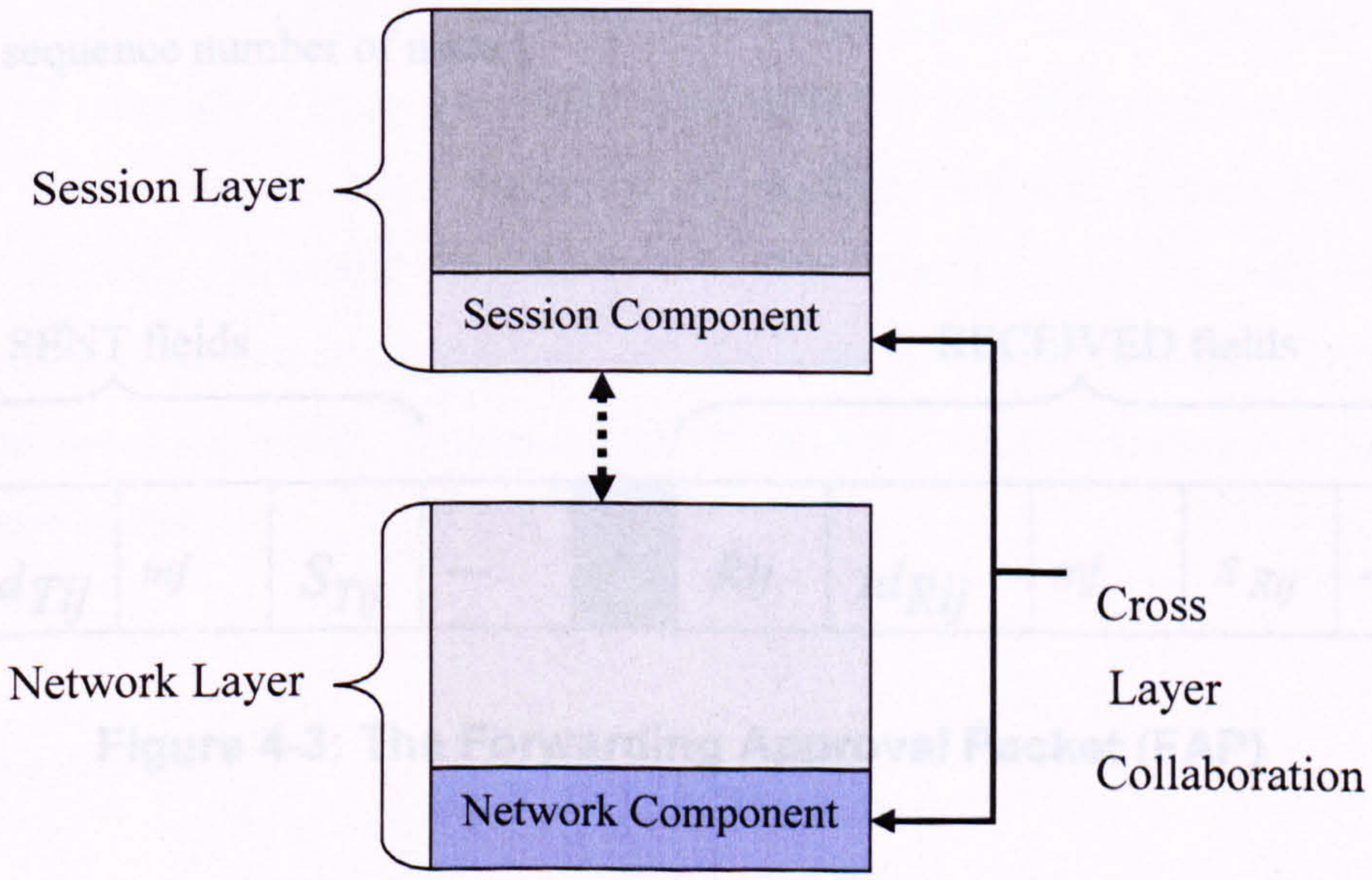



**Figure 4-2: SMDF Cross-Layer Collaboration**

After the end of each session, each node included in a path used by the session (apart from the originated source node and the final destination node) sends two *cryptographically signed* (i.e. using asymmetric encryption) packets. One to its successor containing the number of packets it has sent to it, we denote by NPS, and the other one to its predecessor containing the number of packets it has received from it, denoted NPR. The source node will send only the number of packets it has sent NPS to its successor, and the final destination node will send only the number of packets it has received NPR from its predecessor. NPR and NPS contain the sequence numbers of their sender, which is the number maintained by each node and monotonically increased (by 1) after including it in a packet. This prevents using an NPS or NPR more than once by selfish or malicious nodes. After sending and receiving this information, each node builds and broadcasts to all of its one-hop neighbours a Forwarding Approval Packet (FAP) shown in figure 4-3, which is divided into SENT/RECEIVED fields. Each field involves one neighbour participating in the session, and contains the following attributes:

$Tij$ / $Rij$ : Number of packets node '$i$' has sent/received to/from neighbour 'j'.

$id_{Tij}$ / $id_{Rij}$ : Node identification number (ID) of the sender/receiver node.

$S_{Tij}$ / $S_{Rij}$ : A node signature for authentication.

$mj$: The sequence number of node j.

SENT fields | RECEIVED fields

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $Tij$ | $id_{Tij}$ | $mj$ | $S_{Tij}$ | --- | | $Rij$ | $id_{Rij}$ | $mj$ | $S_{Rij}$ | ---- |

**Figure 4-3: The Forwarding Approval Packet (FAP)**

Note that contrary to almost all the other solutions, our new framework can work independently of the routing protocol, as it does not need to know the two-hop neighbor to monitor its successor. It does it locally with its neighbours as it will be seen in the detection component in the following section. Therefore, there is no requirement of a source routing protocol. Cooperation among nodes is a primary requirement for the network functioning that cannot directly be assumed. Providing service to each other consumes resources, which are generally limited on ad hoc nodes. Furthermore, nodes try to maximize their own utilities in a self-interested way.

Each component in SMDF provides different functionalities and can work individually. However, and all of these components integrated as one framework to provide efficient and robust solution against node misbehaviour in MANET. Our SMDF has the advantages of being independent of the routing protocols, as well as transparent in terms of its capability to integrating with other mechanisms or routing protocols when it is required. The other advantages of SMDF are its flexibility and capability of adding new components to it or removing existing components from it when required to enhance its efficiency.

The detection component of SMDF contains our novel Sessions-based Misbehaviour Detection Protocol SMDP to detect selfish or malicious nodes that drop packets partially or completely to launch either black-hole or data dropping attacks. For the decision component we have enhanced an existing Bayesian approach to decide whether the node deliberately misbehaved or not. For the Isolation component, we have modified an existing approach and used an Observation-Based Protocol to isolate misbehaving nodes. It uses neighbouring observations experience to isolate misbehaved nodes.

## 4.4 Summary

In this chapter we have presented our framework and its cross layer collaboration. First we have described our research objectives that form a comprehensive set of

support mechanisms and schemes. We discussed the gaps in the current knowledge that this thesis will address in the requirements review. We identified requirements, issues and challenges important when designing an effective misbehaviour Detection Framework in stationary MANET and static wireless sensor network. Before developing a security framework that prevents selfish or malicious nodes from harming the network, it is advisable to first create a structured overview on what kinds of attacks in Ad hoc networks the system will target. Our new security framework is targeting at two major attacks in MANET and wireless sensor networks namely dropping data packets attack and black-hole attack.

We have presented our framework Sessions-based Misbehaviour Detection Framework (SMDF) and its cross layer collaboration between the session layer and the network layer. By applying a cross layer design, we increased optimisation in our security framework. SMDF has three components, namely the detection component, the decision component and the isolation component. Each component in SMDF provides different functionalities and can work individually. However, all of these components integrated as one framework to provide efficient and robust solution against node misbehaviour in MANET.

The major advantages of our SMDF include its capability of working either independently or integrating with other routing protocols. The new framework is also flexible and extensible as it has the capability of adding new components to it or removing existing components from it as necessary. Moreover, the new framework design to be transparent in terms of integrating with other mechanisms as required.

In the following chapter we will fully explain and discuses the different components of our SMDF.

# 5. SMDF COMPONENTS

In this chapter, we present in details our new framework Sessions-based Misbehaviour Detection Framework (SMDF) components. We will start with our detection component, which performs through the new novel Sessions-based Misbehaviour Detection Protocol, SMDP, where we explain the concepts of the monitoring method that our protocol used, and the algorithm to do so. Then it will follow up with two case studies in order to illustrate how our SMDP works. Next we explain our modified Bayesian approach in our decision component. Finally, description of our isolation component where misbehaving nodes will be penalised and punished for their misbehaviour.

## 5.1 SMDF Detection Component

After receiving a Forwarding Approval Packet FAP (described in the previous chapter) broadcasted from its one hop neighbour, our detection component represented through our Sessions-based Misbehaviour Detection Protocol (SMDP) will start working. Each node checks the authenticity of each $T_{ij}$ and $R_{ij}$, respectively in the FAP using digital signature. It also checks that none of the sequence number has already been used. For this it keeps the last sequence number of each other node, so that the new received number should be greater than the previous one. Any failure in one of the previous verifications results in considering the appropriate number of packets to be zero, meaning do not accept such information.

If there are no packets dropped the following equation holds:

$$\sum_{i \in I} T_{ij} = \sum_{i \in I} R_{ij} \qquad (1)$$

Thus far, nodes are assumed to not deny the sending and the reception of packets, and accordingly they correctly send the NPS and notably NPR packets, and include all the receptions in the FAPs as well. Now we deal with situations where selfish nodes lie. Assume that there is no more than one such node in a neighbourhood, and we do not consider collusions. If a well-behaving node does not receive NPR or NPS from a neighbouring node, it simply leaves the corresponding signature field empty in the FAP it sends. The neighbours receiving such a packet with an empty signature assume that either the node of the appropriate field or the FAP sender is misbehaving. They keep their IDs for further investigations. This will be enhanced in the following.

We first deal with the situations where nodes do not lie, and all the required signatures are put in the FAP. From equation (1) we consider the following:

$$\sum_{i \in I} T_{ij} = T \;\; \& \;\; \sum_{i \in I} R_{ij} = R$$

If $R-T=0$ then the node is forwarding packets correctly. Otherwise, $(R-T)$ packets has been dropped.

The following steps in figure 5-1, explain our algorithm for detection component when executed by nodes in ad hoc network sessions:

*After the end of each session in the network:*
***If*** *nodes in the session are not the originated source nor the final destination* ***Then***
*Each node sends two signed packets;*
*NPS to its successor;*
*And NPR to its predecessor;*
***Else***
***If*** *node is the originated source* ***Then***
*Sends only NPS to its successor;*

***End if***

***Else***

***If*** *node is the final destination* ***Then***

*Sends only NPR to its predecessor;*

***End if***

***End if***

***When*** *all nodes in the session completed sending and receiving of NPS and NPR:*

***If*** *nodes in the session broadcasted a FAP to their all one hop neighbours* ***Then;***

*For a set of nodes $I$ that surround a single neighbour $j$ ;*

***If*** *each node authenticated all $T_{ij}$ and $R_{ij}$ fields inside the FAP* ***Then***

***If*** *there in no packet drops* ***Then***

$$\sum_{i \in I} T_{ij} = \sum_{i \in I} R_{ij} ;$$

***End if***

***Else***

$$\textbf{If} \sum_{i \in I} T_{ij} = T \;\; \& \sum_{i \in I} R_{ij} = R \;\; \textbf{Then}$$

***If*** *R-T > 0* ***Then***

*(R-T) of packets will be monitored dropped by node j;*

***End if***

***End if***

***End if***

***End if***

**Figure 5-1: Detection Component Algorithm**

Now we treat the cases where a FAP's SENT field regarding some node, for example *X* lacks a signature. Lack of a signature in a RECEIVED field is of no impact if the sender of the FAP has correctly forwarded packets and shows proofs (signatures in the SENT fields). The previous sums (*T* and *R*) are calculated as before, and if *R-T>0,* this number (*R-T*) of packets will be considered dropped. But in addition, the node will not be immediately considered forwarding the *T* packets. In fact, either *X* is denying the reception of packets, or the sender of the FAP has dropped packets and

is lying. The two nodes' IDs as well as the appropriate number of packets (claimed in the SENT field that lacks a signature) are safeguarded in what we call the suspicious set. Later, if one of these two nodes will be considered as suspicious in another experience, it will be charged of dropping packets (both in the first and the second experiences), and the innocent's id will be released from the suspicious set. In the following section we will provide two case studies to clarify the above explanation.

### 5.1.1 Detection Component Case Study 1 (well-behaved nodes)

To illustrate how our novel monitoring approach works consider the following case study as shown in figure 5-2 where an ad-hoc network is shown as a set of 25 nodes (5x5 nodes) in a squared grid surface. Node mobility is supposed to be low enough so that relative positions of nodes do not vary during the sessions.

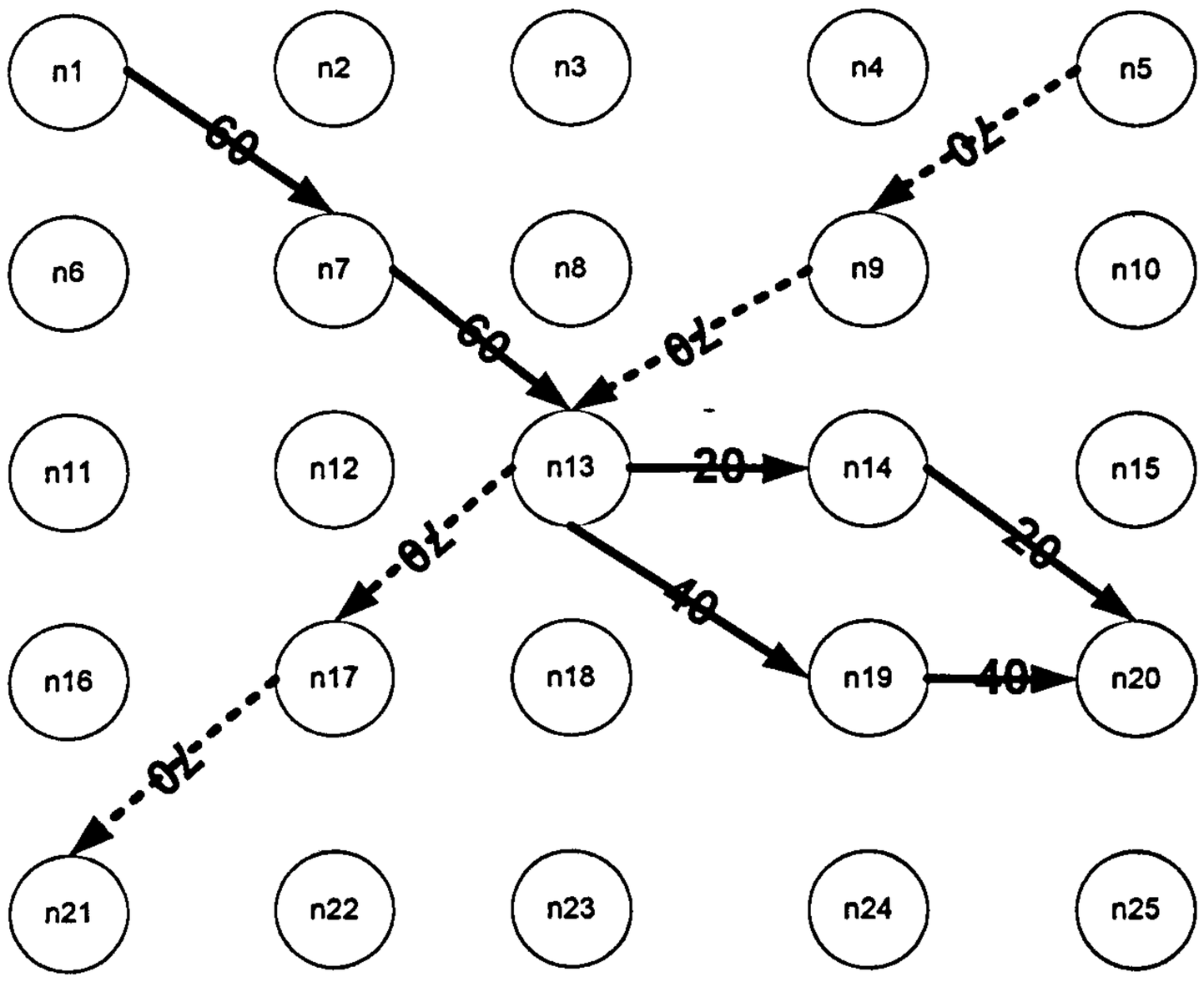



**Figure 5-2: MANET Two Sessions Case Study**

There are two sessions running. The first one shown as a solid arrow in figure 5-2, starts at n1 (session source) and ends at n20 (session final destination), and includes in total 60 packets. These packets are sent from n1 to n7, which forwards them to n13, and then 20 packets are routed through n14 and the remaining 40 through n19. The second session is shown as dashed arrows in figure 5-2, starts form n5 (session source) and ends at node n21 (session final destination). The total number of packets of this session is 70. Node 5 sends the 70 packets to node 9 to forward them to node n13, then from n13 to n17 and finally the latter forwards them to the session final destination n21.

Suppose all nodes are well-behaved. After the end of the first session which starts at n1, each of the nodes n7, n13, n19, n14, sends a signed packet including the number of packets it has received, and another signed packet including the number of packets it has sent. n1 sends only the number of packets it has sent (it does not receive any packet as it is the originated source), while n20 sends only the number of packets received (as it is the final destination).

After the end of the first session, node n13 will send the following signed packet to node 19:

| | | | | |
|---|---|---|---|---|
| Tx | 40 | n13 | m13 | S13 |

Where, Tx is the type of the packet (Tx stands for a packet that includes the number of packet sent and Rx for a packet that includes the number of packets received), 40 is the number of packets sent from node n13 to n19, n13 is the ID of the sender, and finally S13 is a signature of node n13 applied on the packet.

n13 will also send the following signed packet to node n14:

| | | | | |
|---|---|---|---|---|
| Tx | 20 | n13 | m13 | S13 |

And finally it will send the following signed packet to its predecessor n7:

| Rx | 60 | n13 | m13 | S13 |
|---|---|---|---|---|

Node n13 will also receive the following packet from n7:

| Tx | 60 | n7 | m7 | S7 |
|---|---|---|---|---|

And the following packet from n14:

| Rx | 20 | n14 | m14 | S14 |
|---|---|---|---|---|

And finally the following packet from n19:

| Rx | 40 | n19 | m19 | S19 |
|---|---|---|---|---|

After receiving from its neighbours the number of packets it has received and sent, n13 will broadcast the following FAP:

| 40 | n19 | m19 | S19 | 20 | n14 | m14 | S14 | |
|---|---|---|---|---|---|---|---|---|
| 60 | n7 | m7 | S7 | | | | | |

When receiving this packet, neighbouring nodes will check first the authentication of each $T_{ij}$ and $R_{ij}$ in the FAP. Then they will calculate the following:

$$\sum_{i \in I} T_{ij} = 40+20 = 60, \quad \sum_{i \in I} R_{ij} = 60$$

Based on this, neighbouring nodes of node 13 will detect that this latter is forwarding packets correctly without any dropping. On the other hand, the same nodes i.e. n7, n19 and n14 build FAP packets using the packets sent from n13 and their neighbours as well, then broadcast them. Subsequently, they will be evaluated by their neighbours in the same way that n13 has been evaluated.

### 5.1.2 Detection Component Case Study 2: (Selfish and Liar Nodes)

Note that thanks to the sequence number, fields used to construct the FAP cannot be reused. For instance, in a future session involving nodes n13 and n19, the former cannot drop packets and reuse the field (40, n19, m19, S19), as when neighbours receive such a field they remark that m19 has not increased, and consequently do not accept that n13 forwarded 40 packets to n19.

Now we consider the situation where node n13 is selfish. It drops packets received from n7, then it can either put a field with an empty signature, or simply deny the reception of packets from n7 (not sending FAP, neither NPR to n7). Note that it cannot claim forwarding packets to both n19 and n14 with empty signatures, as in this case it will be suspicious simultaneously with the two nodes, thus it will be immediately detected.

Assume it claims forwarding the 60 packets to one of the nodes, such as n14. It then sends the following FAP:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 60 | n14 | m14 | | | 60 | n7 | m7 | S7 |

When receiving such a packet, the neighbours will put nodes n14 and n13 in their suspicious set, along with the number 60. Next, when n13 drops packets of the second session, during which it receives packets from n9, either by sending a FAP with an empty signature regarding n17, or simply denying the reception from n9 and not sending neither the NPR to n7 nor the FAP. In the first case it will be suspicious with n17 then immediately detected by neighbours, after checking their suspicious

sets. Node n17 will not be put in the suspicious sets in this case, and n14 will be removed from the sets. Whereas in the second case, it will be suspicious with n9 when this latter sends its FAP including n13 with an empty signature in the SENT field. n13 will be charged instead of n9, and n14 will be released. In the two cases, n13 will be charged of dropping 130 packets (the sum of the numbers of the two sessions 70+60). If in the earlier session n13 denies the reception of packets from n7, it will be simply suspicious with this latter (instead of n14), when it send its FAP including a SENT field regarding n13 with an empty signature. Identically to the previous scenario, n13 will be detected and n7 released at the end of the second session.

### 5.1.3 Case Studies Analysis

First, we consider the previous example in figure 5-2, then we will generalise the results to infer the communication complexity. In the first session, i.e. the one starting at n1 and ends at n20, there are 6 nodes participating in this session. Let this number be denoted by *h*. At the end of the session, each of the nodes n1, n7, n13, n14 and n19 will send a packet to its successor containing the number of packets it has sent. This makes a total of 5 packets, which is *h-1*.

Also, each of the nodes n7, n13, n14, n19 and n20 will send a packet containing the number of packets it has received from its predecessor. Overall, 5 packets of such a kind will be sent. That is, *h-1*. After receiving the Rx and Tx packets (explained previously), each of the intermediate nodes (n7, n13, n14 and n19) builds and broadcasts a FAP packet to its direct neighbours, resulting in 4 transmissions, which is h-2.

Generally speaking, we have:

- *h-1* packets containing the number of packets sent, i.e. all the nodes, except the destination, send one packet including such an information.
- *h-1* packets containing the number of packets received, all the nodes, except the source, send one packet including such an information.

- *h-2* FAP packets. That is, every intermediate node (neither the source nor the destination) broadcasts such a packet. Overall, we have *3h-4* transmissions which is in term of complexity:

$$\approx O(3(h-1))$$

As we have mentioned, SMDP is operational and can detect misbehaviour when employing the power control technique, contrary to the watchdog-based solutions [Marti'00, Michiardi'02a, Buchegger'02b]. Compared with the random two-hop ACK [Djenouri'06b], our solution is low cost. Using power control technique will make our solution more efficient then the others.

The communication complexity of that solution is:

$$O((h-1)nP_{trust})$$

Where n is the number of packets, and $P_{trust}$ is an intrinsic parameter of the solution. The mathematical study performed in [Djenouri'06b] illustrates that the best value of this parameter is 0.5. Thus, our solution outperforms this one (in terms of overhead reduction) by 6 packets/session. Thus, the reduction factor of the communication overhead is n/6. To explain how we obtained the n/6 reduction factor we do the following:

we calculated previously that our solution's communication complexity is *3*(h-1)*, and the one of Two Hop ACK [Djenouri'06b] is $(h-1)*n*P_{trust}$.

When replacing $P_{trust}$ by 0.5, we get *(h-1)*n*/2. When calculating the reduction factor of our new solution, which is the communication complexity of Two Hop ACK over the one of our solution we obtained:

$$((h-1)*n/2) / (3*(h-1)) = n/6.$$

As for probing, the communication complexity of that solution is:

$$O((h-1)n).$$

Our solution outperforms it by 3 packets/session. The reduction factor of the communication overhead is n/3.

To explain how we obtained the n/3 reduction factor we do the following:

$$((h\text{-}1)*n) / (3*(h\text{-}1)) = n/3.$$

### 5.1.4 Optimised SMDF Using Sessions Aggregation

Our solutions SMDF can be optimised even further to reduce the communication overhead. This can be done by aggregating sessions. When using this approach, nodes that are involved in more than one session could wait a certain time until all sessions end before sending the FAP to their direct neighbours. For example, n13 in figure 5-2 can wait until both sessions end, then sends one aggregated FAP to its neighbours regarding the two sessions, instead of sending two FAPs separately. The aggregated packet is:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 40 | n19 | m19 | s19 | 20 | n14 | m14 | S14 | 70 | n17 |
| m17 | S17 | | 60 | n7 | m7 | S7 | 70 | n9 | m9 | S9 |

In this way we reduce the communication overhead even further. This optimisation is beneficial for well-behaving nodes. A selfish node, however, has no interest of aggregating FAPs, since lying in such a packet will inevitably include two nodes, which allows to directly detect it.

## 5.2 SMDF Decision Component

In this component we develop a modification approach of the standard Bayesian method. As described above the Decision Component of the new framework will judge whether the nodes misbehave intentionally or not. First we give an overview of

Our solution outperforms it by 3 packets/session. The reduction factor of the communication overhead is n/3.

To explain how we obtained the n/3 reduction factor we do the following:

$$((h\text{-}1)*n) / (3*(h\text{-}1)) = n/3.$$

### 5.1.4 Optimised SMDF Using Sessions Aggregation

Our solutions SMDF can be optimised even further to reduce the communication overhead. This can be done by aggregating sessions. When using this approach, nodes that are involved in more than one session could wait a certain time until all sessions end before sending the FAP to their direct neighbours. For example, n13 in figure 5-2 can wait until both sessions end, then sends one aggregated FAP to its neighbours regarding the two sessions, instead of sending two FAPs separately. The aggregated packet is:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 40 | n19 | m19 | s19 | 20 | n14 | m14 | S14 | 70 | n17 |
| m17 | S17 | | 60 | n7 | m7 | S7 | 70 | n9 | m9 | S9 |

In this way we reduce the communication overhead even further. This optimisation is beneficial for well-behaving nodes. A selfish node, however, has no interest of aggregating FAPs, since lying in such a packet will inevitably include two nodes, which allows to directly detect it.

## 5.2 SMDF Decision Component

In this component we develop a modification approach of the standard Bayesian method. As described above the Decision Component of the new framework will judge whether the nodes misbehave intentionally or not. First we give an overview of

the original standard Bayesian method, and then we describe our modified Bayesian approach, which we used in our SMDF Decision Component.

### 5.2.1 A Standard Bayesian Framework Overview

The Bayesian approach [Davison'00, Davison'03] is a mathematical estimation method for estimating a parameter the observations of which follow a Bernouli distribution by a Beta distribution. The Bernoulli Distribution [Evans'00] is an example of a discrete probability distribution. A discrete probability distribution is a roster comprised of all the possibilities, together with the likelihood of the occurrence of each. The Bernoulli Distribution is an appropriate tool in the analysis of proportions and rates.

Several distributions such as beta, Gaussian, Poisson and binomial can be used to represent the reputation of a node. However, the beta distribution has been the most promising due to its flexibility and simplicity as well as its strong foundations on the theory of statistics. It is based on probability distributions that are a fundamental concept in statistics. They are used both on a theoretical level and a practical level. One of the important practical uses of probability distributions is in simulation studies with random numbers generated from using a specific probability distribution are often needed.

The general formula for the probability density function [Evans'00] of the beta distribution is:

$$f(x) = \frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}} \qquad a \le x \le b; p, q > 0$$

Where $p$ and $q$ are the shape parameters, $a$ and $b$ are the lower and upper bounds, respectively, of the distribution, and $B(p,q)$ is the beta function. The beta function has the equation:

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$$

The case where $a = 0$ and $b = 1$ is called the standard beta distribution. The equation for the standard beta distribution is:

$$f(x) = \frac{x^{p-1}(1-x)^{q-1}}{B(p,q)} \qquad 0 \le x \le 1; p,q > 0$$

Typically the general form of a distribution is defined in terms of location and scale parameters. The beta is different in that we define the general distribution in terms of the lower and upper bounds [Evans'00]. However, the location and scale parameters can be defined in terms of the lower and upper limits as follows: location = $a$, scale = $b - a$.

On the other hand, the Bayesian approach has the advantage of not needing a memory (i.e. only the latest updates are safeguarded, and not all the observations). For example, node $i$ models the behaviour of node $j$ as an actor in the base system as follows:

Node $i$ thinks that there is a parameter $\theta$ such that node $j$ misbehaves with probability $\theta$, and that the outcome is drawn independently from observation to observation (Node $i$ thinks that there is a different parameter $\theta$ for every different node $j$, and every node $i$ may believe in different parameters $\theta$. Therefore, $\theta$ should be indexed by $i$ and $j$). The parameters $\theta$ are unknown, and node $i$ models this uncertainty by assuming that $\theta$ itself is drawn according to a distribution the "prior" that is updated as new observations become available. This is the standard Bayesian framework. We use the distribution $Beta(\alpha, \beta)$, as is commonly used in [Davison'03, Buchegger'04], since it is suitable for Bernoulli distributions and the conjugate is also a Beta distribution.

The standard Bayesian procedure is as follows:

Initially, the prior is $Beta(1,1)$, the uniform distribution on [0,1]; this represents absence of information about $\theta$ which will be drawn. Then, when a new observation

is made, say with $\delta$ observed misbehaviours and $f$ observed correct behaviours, the prior is updated according to $\delta := \alpha + \delta$ and $\beta := \beta + f$. If $\theta$, the true unknown value, is constant, then after a large number $n$ of observations, $\alpha \sim n\theta$ (in expectation), $\beta \sim n(1-\theta)$ and $Beta(\alpha, \beta)$ becomes close to a Dirac at $\theta$, as expected. The advantage of using the Beta function is that it only needs two parameters that are continuously updated as observations are made or reported.

The Bayesian approach for nodes reputation regarding packet forwarding in MANET has already been used by [Josang'02, Mui'02, Buchegger'04, Djenouri'07], but requires periodic transmissions of huge control packets noticeable in most of them.

### 5.2.2 Our New Modified Bayesian Decision Stage

We have modified Bayesian approach for our new Decision Stage. Our new proposed Bayesian approach is similar to that used in [Buchegger'04, Djenouri'07] but with advantages of lower overhead. The monitoring method in SMDP described above allows the neighbouring nodes to decide whether each monitored node in the session has forwarded packets correctly or not. Therefore, when a monitoring node notices that some packet has been dropped over a link it should not directly accuse the monitored as misbehaving, since this dropping could be caused by collisions or channel conditions. Therefore, a threshold of tolerance should be fixed.

In our new Bayesian approach, well behaving of nodes improves their reputation, whereas intentional or unintentional packet dropping decreases it. Since misbehaving is usually exception rather than the norm, information exchange in our solution is limited to negative impressions. It is simpler and creates no overhead when nodes well-behave.

Each node A thinks that each other node B misbehaves with a probability $\theta$, which is a random variable estimated by a Beta distribution $Beta(a,b)$ described above. Initially with no prior information, $\theta$ is assumed uniform in [0,1], which is identical to $Beta(1,1)$. As observations (that follow a Bernoulli distribution with a parameter $\theta$) are made, a and b are updated as follows:

$$a = a + T, b = b + (R - T)$$

Where R is the number of packets received by the node (as a router), and T is the number of packets forwarded by it during the session, as mentioned in our detection component. The previous sums ($T$ and $R$) are calculated as before in our detection component, and if $R\text{-}T>0$, this number ($R\text{-}T$) of packets will be considered dropped. But in addition, the node will not be immediately considered forwarding the $T$ packets. If $R-T=0$ then the node forwards packets correctly. Otherwise, $(R-T)$ packets are dropped.

After as many observations as the decision could be made ($\theta$ could be approximated by the mathematical expectation $E(Beta(a,b))$), B will be judged. This is denoted by the decision (or stationary) point i.e. $E(Beta(a,b)) > E_{max}$, while the number of observations is expressed by $a+b$. Upon reaching this point, B will be accused of misbehaviour. Note that: $E(Beta(a,b)) = a/(a+b)$.

The following steps shown in figure 5-3 describe the decision algorithm when executed by a node $i$ :

*Received a notification* $(R-T)$ *from the detector regarding node* $j$ $((R-T)=1$ *if node is dropping packets and* $(R-T)=0$ *otherwise):*

$a_j = a_j + (R-T);$

$b_j = b_j + 1 - (R-T);$

$\theta_j = a/(a+b);$

***If** (Decision point reached) **Then***

***If** ($\theta_j > E_{max}$) **Then***

*Put node* $j$ *in the suspicious set;*

*Launch Observation_REQ against node* $j$ *;*

***End if***

***End if***

**Figure 5-3: SMDF Decision Component Algorithm**

$E_{max}$ could be fixed to 0.5 (i.e. 50% of misbehaviour), or for more efficiency it should be estimated empirically for each network as follows:

1) Make simulations with no misbehaving and calculate E at each node for different scenarios that estimate the network.

2) Retrieve the maximum value in all scenarios from the decision point then consider it as $E_{max}$.

In mathematical estimation methods, the decision (stationary) point is the one upon which the difference between two subsequent observations could be negligible. One usual choice is that fulfilling the following condition:
$Var(Beta(a,b)) < \varepsilon$.

Such that $Var$ is the mathematical variance and $\varepsilon$ is a very small positive.

Note that:

$$Var(Beta(a,b)) = \frac{a \times b}{(a+b+1) \times (a+b)^2}$$

However, this choice is inappropriate here, since $Var(Beta)$ is not monotonous with $a+b$. We use the following variance like function, which is indeed decreasing with $a+b$:

$$Max\left(\frac{b}{(a+b) \times (a+b+1)}, \frac{a}{(a+b) \times (a+b+1)}\right)$$

When enough observations with regard to a given monitored node are collected such that the judgment point is reached, the monitoring node will accuse the monitored one as soon as the estimated probability ($E(Beta(a,b))$) exceeds the configured maximum tolerance threshold, i.e. $E(Beta(a,b)) > E_{max}$.

$$E(Beta(a,b)) > E_{max} \longleftrightarrow \frac{a}{a+b} > E_{max} \longleftrightarrow a > \frac{b \times E_{max}}{1 - E_{max}} :$$

This latter $\left(\frac{b \times E_{max}}{1 - E_{max}}\right)$ represents the tolerable number of packets a node is allowed to drop without being accused. This maximum tolerable threshold is proportional to b, the number of packets forwarded. The more a node forward packets, the more its tolerable threshold increases. Forwarding packets after unintentional or intentional droppings that do not result in an accusation would decrease *E*, which allows redemption. This redemption could not be possible when setting the tolerable threshold to a fixed number of packets. In our SMDF the redemption is just before Decision, a node that forwards packets will need much more packets to be dropped before being accused compared to the one that does not forward, so it is like the forwarding redeem its dropping. However, there is no redemption after the decision.

In [Buchegger'04], every node periodically broadcasts in its neighbourhood its view of $\theta$ regarding all the other nodes. Nodes use these information (known as second hand information) to update their own opinion on nodes' behaviour. To decide about the acceptance of the provided information, each node performs complicated tests on the trustworthiness of the provider. The problem with this proactive solution is that it causes an increase in the amount of overhead generated, even if nodes well-behave. This overhead is also noticeable in [Djenouri'07] when misbehaviour detected. Our approach is rather reactive, thus no such information are exchanged. Indeed, each node performs monitoring separately and informs the others as soon as a misbehaving node is approved, as we will see in the next section with more details.

## 5.3 SMDF Isolation Component

Our Isolation Component derived from the social sciences principle that a person that accuses another of misconduct must show proof. One possible way to prove the accusation is to get observers against the accused person. In order to mitigate false

detections and false accusations vulnerability, we have proposed a Observation-Based Protocol similar to that used in [Du'03, Djenouri'07] to isolate a detected node. In this protocol, a node that detects and accuses another as misbehaving must approve its accusation before taking any measure against it. It should not isolate the assumed misbehaving unilaterally, because this could result in false detections against it. However, it could avoid routing its own packets through this node.

Isolating a misbehaving node in MANET required two actions. First, not to route packets through it, to avoid losing them; second, do not forward packets for it, in order to punish it. For example, node A that judges some other node B as misbehaving should not isolate it unilaterally, but must ensure its isolation by all nodes. This is because when A unilaterally isolates B, the others could consider A as misbehaving when they realize that it does not forward packets for B.

The way that our proposed Observation-Based Protocol work is describe as follow: Upon detection, the detector informs nodes in its neighbourhood about the dropper (the accused), and asks for observers by broadcasting an Observation REQuest (OREQ) packet. It also puts the detected node ID in a special set called a *suspicious set*. Each node receiving the OREQ investigates the issue as follows:

The packets recipient immediately sends a *signed* Observation REPly (OREP) packet to the accuser in the following two cases:

- if the accused node's misbehaving expectation reach $E_{max}$, or the number of control packets considered dropped reach the configured maximum threshold.
- if its suspicious set includes the accused node.

Otherwise, when it has not enough experience with the accused node (B), and if B is its neighbour then it asks the successor of this latter whether it has received packets forwarded from it, by sending an ACcusation REQuest ACREQ packet (using a route that does not include B). But first, in order to avoid false accusations, the investigator

(i.e. the node received the ACREQ packet from the accuser), should ensure that the accuser has really sent a packet to B to be forwarded to the appropriate successor. One possible way to do this is to check whether such a packet has been recently overheard, using the promiscuous mode. The node also should check whether B has sent the accuser an ACK just after overhearing the data, to ensure that the former has really received the packet and that the latter is not impersonating it. If B's successor has not recently received any packet *forwarded* from B, it sends a *signed* ACREP (ACcusation REPly) packet to the investigator, then this latter testifies for the accusation and sends the accuser a signed Observation REPly- OREP packet. The signature of the packets prevents their spoofing, thus no node could testify using the ID of another.

The accuser node has to collect '$S$' different signatures to approve its accusation. Theoretically, $S-1$ is the maximum number of misbehaving nodes that could exist at any time. In practice, however, it is hard to determine such a number, so it should be fixed to strike a balance between efficiency and robustness. Setting $S$ to a high value increases the robustness of the protocol against false detections and rumours, but decreases its efficiency regarding true detections. On the other hand, a low value of $S$ allows high detections, but opens the vulnerability of rumours and increases the unintentional false detections (false positives), since $S$ nodes could collude to accuse maliciously (respectively wrongly) any node.

Once the accuser collects $S$ valid signatures, it broadcasts an Isolation Packet (ISOP) including all signatures through the network to isolate the guilty. This broadcast is not performed until a node is detected and approved as misbehaving. Apart from the monitoring stage, our solution requires no overhead as long as nodes well behave, as no opinions are exchanged periodically. This gives our solution the advantages of being a reactive one, unlike the other reputation-based solutions that were presented before. The following algorithm description in figure 5-4 and case study in section 5.3.1 will illustrates the isolation component further.

*When receive a Observation_REQ sent by node A against node j :*

***If*** *( j ∈ the suspicious set or* $\theta_j \cong E_{max}$ *or Num_Pkt_Drop* $_j \cong$ *Threshold) then*
*send a direct signed Observation_REP to A ;*

***Else***

*If ( j is a direct neighbour of i in the session)* ***Then***
*send ACcusation REQuest toward j 's successor using a route does not include j ;*

***End if***
***End if***

*When receive a ACcusation REQuest sent by B against j where A is the previous hop:*

***If*** *(no packet has been recently forwarded from j including A as the previous hop)* ***Then***
*send B a ACcusation REPly ;*
***End if***

*When receive a ACcusation REPly regarding' A ' accusation:*

*send A a signed undirect Observation REPly;*

*When receive an Observation REPly sent by A against j :*

***If*** *(Observation REPly type = direct)* ***Then***
*num_dirct_Obs = num_dirct_Obs + 1;*

***Else***
*num_undirct_Obs = num_undirct_Obs + 1;*
***End if***

***If*** *(num_dirct_Obs + num_undirct_Obs = S and num_dirct_Obs > 0* ***Then***
*broadcast ISOP to isolate the misbehaving j ;*
***End if***

**Figure 5-4: SMDF Isolation Component Algorithm**

### 5.3.1 Isolation Component Case Study

In this example we assume the short session shown in figure 5-5 and running through n1, n2, n7 and ending in n8. The source originator n1 sent 40 packets to to be delivered to the final destination n8. When n1 accuses n2 for not forwarding all packets to n7 and sends a call for observation, n6 which is a one hob neighbour of n1 and n2 investigates the issue. But before asking n7 it ensures that n1 has really sent the packet and n2 has received it, by checking FAP packets it has received. This is because n6 could not ensure that n2 has received the data packet by just overhearing it. For example, if n6 is closer to n1 than n2, n1 attempting a DoS attack against n2 could send the packet using a power strong enough to be overhead by n6, but not by n2. Requiring the FAP reception from n2 just after the data ensures that n2 has really received the data from n1. To do so, n6 simply safeguards the overheard packets in this case their headers during a short period.

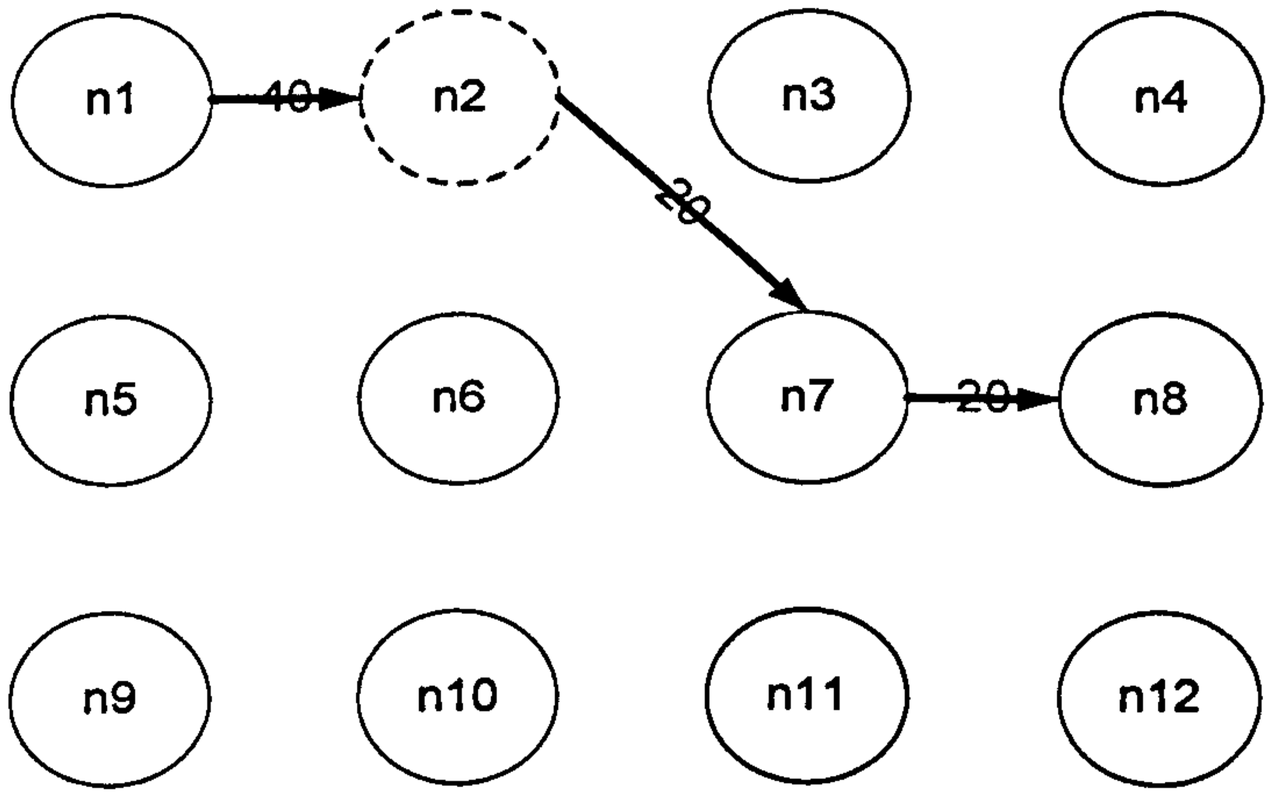



**Figure 5-5: SMDF Isolation Component Case Study**

In this way, a node that asks the accused node's successor has no doubt that the accused node has received a data packet to forward to the successor in question. Any collision at n6 prevents it from testifying, but has no effect on false detections. Upon the reception of the ACREQ, the asked node n7 replies with a signed ACREP packet if it has not received any packet from n2. n1 coincidental collision at n7 at that moment, however, would result in a false reply if n1 is attempting a DoS attack, then in a false testimony. Nonetheless, the requirement of at least one direct observation

(be an observer from its direct experience) prevent wrong accusation caused by this kind of false testimonies. The signature of the packets prevents their spoofing, thus no node could testify using the ID of another.

The accuser node n1 has to collect 2 different signatures to approve its accusation. Once the accuser n1 in this case collects 2 valid signatures from n5 and n6 as they are within one hop of n2, it will then broadcasts an Isolation Packet (ISOP) including all signatures through the network to isolate the misbehaving node, which is in this case n2. This broadcast is not performed until a node is detected through our first detecting component and then approved through our decision component as misbehaving. In theory, as it has been described above, (2-1) is the maximum number of misbehaving nodes that could exist at any time. In reality, it is difficult to determine such a number of signatures, so it should be fixed to keep a balance between efficiency and robustness. Furthermore, setting the number of signatures to a high value will increases the robustness of the protocol against false detections, however, it will decreases its efficiency regarding true detections. In contrast, a low value of this number allows high detections, but opens the vulnerability of rumours and increases the unintentional false detections (false positives), since certain number nodes could collude to accuse maliciously any node.

## 5.4 Summary

The different components of our framework have been fully described in details in this chapter. We started with our detection component through the new Sessions-based Misbehaviour Detection Protocol SMDP, where we explained the concepts of the detection/monitoring that our protocol used, and the algorithm to do so. We have given two case studies in order to illustrate how our SMDP work. We also explained how our detection component can be optimised even further to reduce the communication overhead using aggregating sessions. When using this approach, nodes that are involved in more than one session could wait a certain time until all sessions end before sending the FAP to their direct neighbours. In this way we

reduce the communication overhead even further. It is a trade-off issue as reducing the cost is more valuable and important than increasing the waiting time until the end of all sessions.

Then we went to explain our Decision component. We have proposed a modified Bayesian approach for this component. The advantages of this approach are that it allows redemption before making decisions, and it decreases false accusations due to wireless channel conditions. Following our decision component, we presented and explained our SMDF final component the Isolation component. At this component misbehaving nodes will be penalised and punished for their misbehaviour. For this component, we have modified an existing approach and we used an Observation-Based Protocol to isolate misbehaving nodes. In this protocol, a node that detects and accuses another as misbehaving must approve its accusation before taking any measure against it. This increase fairness and reduce false accusations among nodes. Once the accuser collects enough valid signatures from other neighbouring nodes, it broadcasts an Isolation Packet including all signatures through the network to isolate the misbehaving node.

In the following chapter we will evaluate our proposed framework SMDF using simulation techniques.

# 6. EVALUATION AND SIMULATION RESULTS

The previous chapter described our new framework SMDF to detect misbehaving nodes in MANET. In this chapter we present the evaluation of our work. We apply our methodology for testing and this phase defines the requirements and assumptions. We describe the performance evaluation and we discuss different simulation scenarios that show what happens when we modify the initial system state. We outline the metrics and parameters within our simulator that are the container for the initial data set for any scenario. We show through simulation the effect on node misbehaviour on the network throughput. We analyse overall detection rate performance on the network simulator when using our new framework SMDF. We show how our SMDF framework succeeds in detecting misbehaving nodes at different levels of misbehaviour with low overhead and high rate of detection success. We evaluate the proposed mechanism using simulation techniques by determining the utilization level of network resources achieved using them, and by comparing our results based on simulation models to the best possible deterministic schemes available on the literature. We then take a very broad view of the research and look at the overall achievements including evaluation against our initial requirements specified in chapter 4 and discuss the problems remaining.

## 6.1 Simulation

Simulation is a fundamental tool in the development of MANET protocols, because of the difficulty to deploy and debug them in real networks. The simulation eases the analysing and the verification of the protocols, mainly in large-scale systems. It offers flexible testing with different topologies, mobility patterns, and several

physical and link-layer protocols. However, a simulation cannot provide evidence in real-world scenarios, due to assumptions and simplifications that it makes. Various examinations, such as [Sasson'02], show significant divergences between different simulators that demonstrate an identical protocol. Therefore, the results obtained from the simulations should be evaluated appropriately. Three well-known simulators are used for MANET simulations: NS-2 [NS2'07], GloMoSim and OPNET [OPNET'07]. We chose GloMoSim [GloMoSim'07], because it is a scalable simulator that was designed especially for large wireless networks. It supports thousands of nodes, using parallel and distributed environments.

### 6.1.1 GloMoSim Overview

GloMoSim [Zeng'98, Bajaj'99, Nuevo'03], was designed as a set of library modules, each of which simulates a communication protocol in the protocol stack. The library uses the OSI layer approach shown in table 6-1 and supports multiple protocols in each layer. The layers are separated and each layer has its own API. The layers interact with each other using message-passing approach. A combination of different protocols at various layers into a complete protocol suite, as well as extension with alternative protocols can be done simply. The simulator is built above PARSEC [PARSEC'07], a C-based language that was developed for discrete-event simulations. The simulator enables various scenarios, using configuration files, and allows analysis by a trace file with statistics. The visualization tool of GloMoSim, written in Java, shows the network topology, nodes' mobility and packet transmissions.

### 6.1.2 Validation

The validation is a feedback loop utilising empirical network data and the simulation. For performance evaluation, we collect and analyse the results from the simulation. We compare the performance of our techniques with other results from key literature. We also relate the experimental results to the research objectives, and we discuss the

extent to which the project has succeeded in its goal. The research objectives check against the performance evaluation, and we discuss conclusions. In this way, we are able to reason about the advantages and possible limitations of our techniques.

| Layers | Protocols |
|---|---|
| Mobility | Random waypoint, Random drunken, Trace based |
| Radio Propagation | Two ray and Free space |
| Radio Model | Noise Accumulating |
| Packet Reception Models | SNR bounded, BER based with BPSK/QPSK modulation |
| Data Link (MAC) | CSMA, IEEE 802.11 and MACA |
| Network (Routing) | IP with AODV, Bellman-Ford, DSR, Fisheye, LAR scheme 1, ODMRP, WRP |
| Transport | TCP and UDP |
| Application | CBR, FTP, HTTP and Telnet |

**Table 6-1: GloMoSim OSI Library**

## 6.2 Simulations Parameters

To study the effect of node misbehaviour on MANET and to assess the performance of the proposed detection protocol, we have developed a GloMoSim-based [GloMoSim'07] simulation study. We have simulated a network of 100 nodes, located in an area of 2500 × 2000 $m^2$ where nodes are deployed randomly for 1800 seconds of simulation time. To generate traffic we have used five Constant Bit Rate (CBR) sessions between five pairs of remote nodes, each consists of continually sending a 512 byte data packet each second. On each hop, each data packet is transmitted using a controlled power according to the distance between the transmitter and the receiver.

We have set the seed parameter to number 3, which represent the random number seed used to initialise part of the seed of various randomly generated numbers in the simulation. This can be used to vary the seed of the simulation to see the consistency of the results of the simulation. The two parameters "TERRAIN-DIMENSIONS (2500, 2000)" stand for the physical terrain in which the nodes are being simulated. In our case it represents an area of size 2500 meters by 2000 meters. All range parameters are in terms of meters.

The parameter "NODE-PLACEMENT" represents the node placement strategy, which we set to RANDOM meaning nodes are placed randomly within the physical terrain. We set the nodes mobility parameter "MOBILITY" to NONE, meaning there is no movement of nodes in the model and nodes are static. The PROPAGATION-LIMIT parameter set to -111.0. This value must be smaller than (RADIO-RX-SENSITIVITY + RADIO-ANTENNA-GAIN) explained below, of any node in the model. Otherwise, simulation results may be incorrect. Lower value should make the simulation more precise, but it also makes the execution time longer.

The RADIO-TYPE which is the radio model to transmit and receive packets set to "RADIO-NONOISE" an abstract radio model which is compatible with version (2.1b5) of ns-2 radio model. RADIO-FREQUENCY parameter (in hertz) (Identifying variable for multiple radios set to 2.4e9, and RADIO-BANDWIDTH to 2000000 bit/sec. RADIO-RX-TYPE packet reception model parameter set to SNR-BOUNDED, If the Signal to Noise Ratio (SNR) is more than RADIO-RX-SNR-THRESHOLD (in dB), it receives the signal without error. Otherwise the packet is dropped. RADIO-RX-SNR-THRESHOLD set to 10 dB. The RADIO-TX-POWER parameter is the radio transmition power (in dBm), and it set to 7dBm. The RADIO-ANTENNA-GAIN set to 0.0 dB. The RADIO-RX-SENSITIVITY parameter which represent the sensitivity of the radio set to -91.0 dBm and RADIO-RX-THRESHOLD parameter represent the Minimum power for received packet (in dBm) set to -81.0.

We also set the PROMISCUOUS-MODE parameter to YES and is necessary if nodes want to overhear packets destined to the neighbouring node as the case in the Watchdog. Also this option needs to be set to YES only when DSR is selected as routing protocol. Setting it to "NO" may save a minor amount of time for other protocols.

The NETWORK-PROTOCOL parameter set to IP the only choice. The following parameters determine our interest in the statistics of a single or multiple layers. By specifying the following parameters as YES, the simulation will provide us with statistics for that particular layer. All the statistics are compiled together into a file called "GLOMO.STAT" that is produced at the end of the simulation.

| | |
|---|---|
| APPLICATION-STATISTICS | YES |
| TCP-STATISTICS | NO |
| UDP-STATISTICS | NO |
| ROUTING-STATISTICS | NO |
| NETWORK-LAYER-STATISTICS | YES |
| MAC-LAYER-STATISTICS | NO |
| RADIO-LAYER-STATISTICS | YES |
| CHANNEL-LAYER-STATISTICS | NO |
| MOBILITY-STATISTICS | NO |

Table 6-2 shows the important simulation parameters that have been used in our simulation. These parameters are typical for MANET simulations (see e.g.[Broch'04]) and are used for all following simulations. For the results of the simulation to be meaningful, it is important that the model on which is based the simulator matches the reality as closely as possible. As mentioned in chapter 4 that various examinations, such as [Sasson'02], show significant divergences between different simulators that demonstrate an identical protocol. Therefore, the results obtained from the simulations should be evaluated appropriately.

| Parameter | Value |
|---|---|
| Number of Nodes | 100 |
| Area X (m) | 2500 |
| Area Y (m) | 2000 |
| Traffic Model | Constant Bit Rate (CBR) |
| Sending Rate (Packets/S) | 1.0 |
| Packet Size (Byte) | 512 |
| Simulation Time (S) | 1800 |
| Node Placement | Random |

**Table 6-2: Simulation parameters**

## 6.3 Simulation Metrics

We evaluate our proposed SMDF using the following six metrics:

- **Throughput:** This is the percentage of sent data packets actually received by the intended destinations.
- **Overhead:** This is the amount of control-related transmissions (control packets including FAPs) measured in bytes, and generated during each session in the network. We count the amount of the actual control packets in bytes instead of the number of packets, because it is reflect the real amount of overhead, as you might have small number of packets that generates huge amounts of bytes and vice versa. Control packets are broadcast to all one hop neighbours as in the case of FAPs, which described previously in Chapter 4.
- **True Positive Detection Rate:** The rate of true dropping detection, when nodes correctly detected dropping packets.
- **False Positive Detection Rate:** The rate of false dropping detection, when nodes wrongly accused of misbehaviour, when in fact they are not.
- **Power Consumption Rate:** The effect of node misbehaviour on the detection protocol's power consuming.
- **Scalability:** How scalable the new protocol is if the network number of nodes increased.

## 6.4 Evaluation of the Effect of Node Misbehaviour on MANET Throughput

In order to study how node misbehaviour affects a MANET performance, we have done a number of simulations where we modelled a varying number of selfish nodes. In order to compare the affect of node misbehaviour in the network, we first run the simulation without selfish nodes (i.e. all of the nodes in the sessions are behaving correctly and forwarding packets as required from them without any dropping). Next, we run the simulation and in this case we have injected the network with selfish nodes who misbehave by not forwarding packets they received from other nodes. We have varied the number of selfish nodes from 1 to 20 nodes of the total numbers of 100 nodes. Figure 6-1 shows the results of these simulations. It is obvious that this number has a significant effect on the rate of packets that are successfully delivered in the network. In this simulation we have used DSR routing protocol, the selfish node has not been detected by DSR and no countermeasures are taken.

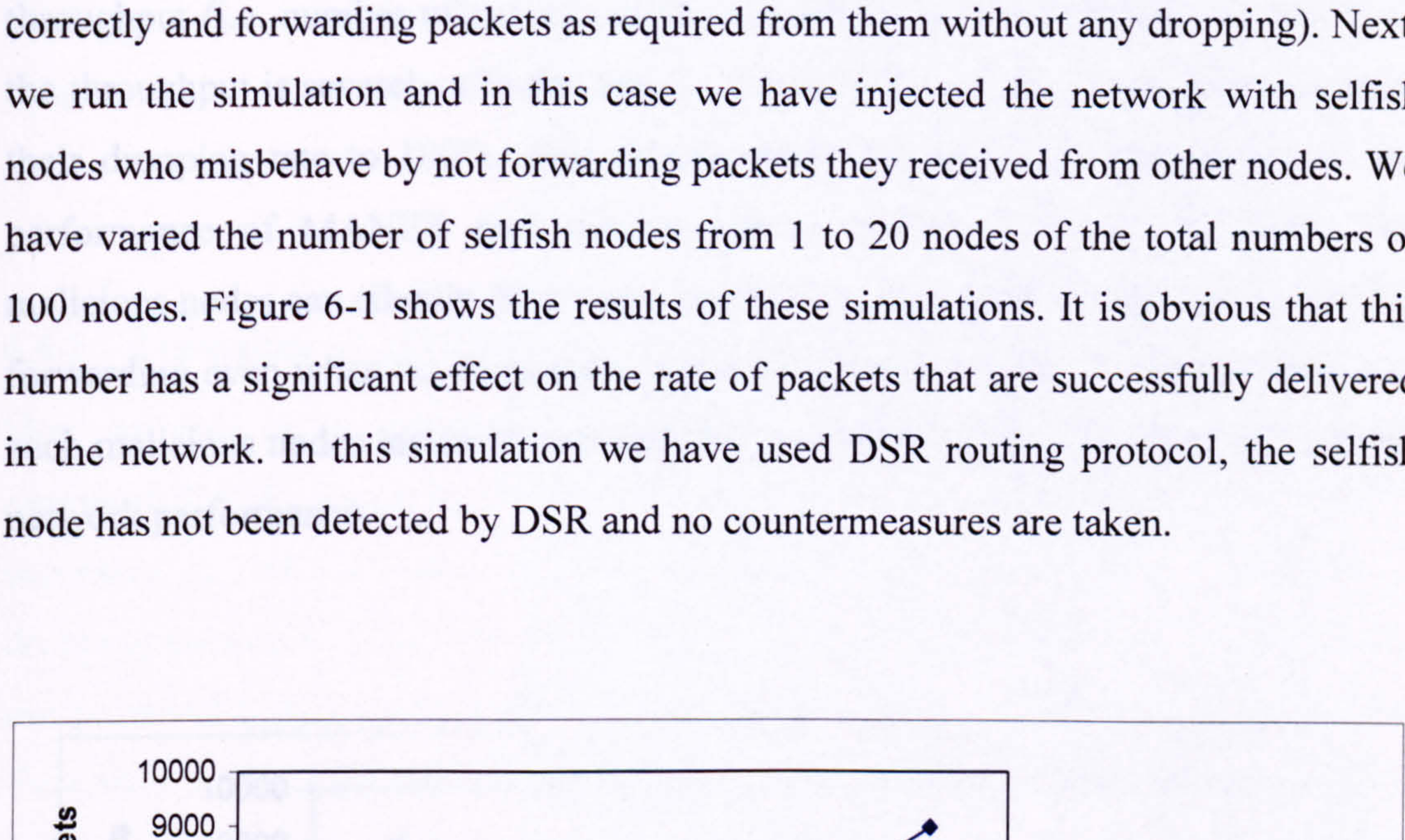



**Figure 6-1: Node Misbehaviour Effects on MANET's Throughput**

## 6.5 Evaluation of the Effect of Packet Dropping Attack in MANET

In this simulation we study the effect of packet dropping attack on MANET. Unlike the previous simulation in figure 6-1 where the dropping rate is fixed to 50%, in this simulation the dropping rate is vary from 0% to 100%. We simulated 20 nodes launching this attack by different rate of dropping as shown in figure 6-2. It can be seen from figure 6-2 that when the dropping rate of the attacker is low, the throughput (i.e. number of packet received) is high. As the dropping rate increased the throughput is severely affected until it reaches 0 as the attacking nodes increased their dropping rate to 100%. This clearly shows the affect of such attack on the performance of MANET and wireless sensor network. The result shows that malicious nodes can silently drop some or all of the data packets sent to it for further forwarding even when no congestion occurs. It also shows that the more number of such malicious nodes inside the network the more the harmful impact on the overall network performance.

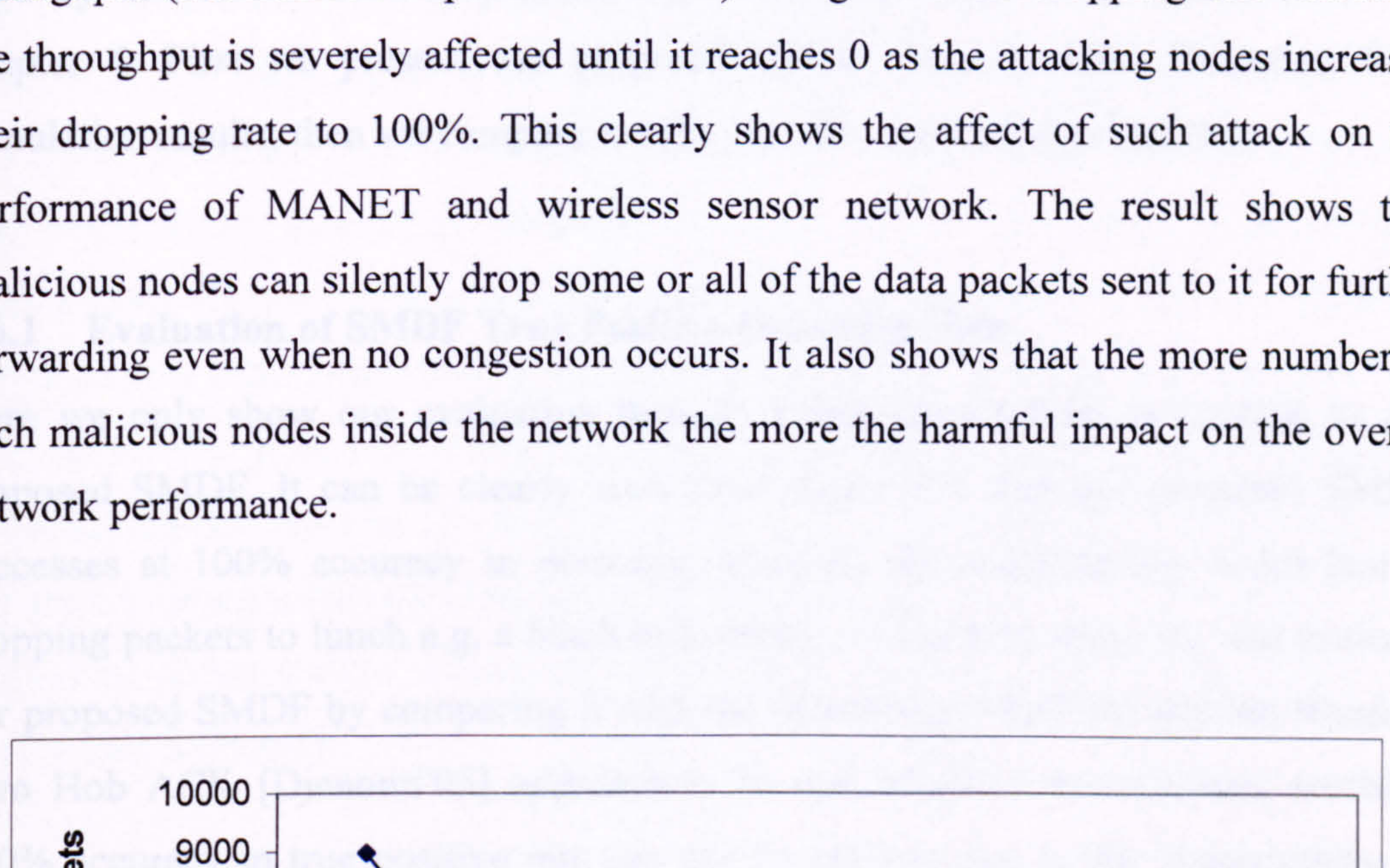



**Figure 6-2: The Effect of Packet Dropping Attack on Throughput in MANET**

## 6.6 Evaluation of the True Positive Detection Rate

This metric is an important one, as it shows how successful our proposed protocol is in terms of detecting misbehaving nodes correctly. We have set the misbehaving rate to be varied from 0% to 20%, which means that 0-20 nodes are misbehaving and dropping packets at the rate of 50% of the overall packets they have received. This scenario reflects the partial dropping case, which is really difficult to detect, and it is used by malicious nodes to perform the black-hole attack as described above in chapter 4. First we present our proposed SMDF True Positive Detection Rate Simulation results, then we compare it with the other existing mechanisms.

### 6.6.1 Evaluation of SMDF True Positive Detection Rate

Here we only show our evaluation through simulations results in regards to our proposed SMDF. It can be clearly seen from figure 6-3 that our proposed SMDF successes at 100% accuracy in detecting correctly the misbehaving nodes that is dropping packets to lunch e.g. a black-hole attack. In the next stage we will evaluate our proposed SMDF by comparing it with the Watchdog [Marti'00] and the Random Two Hob ACK [Djenouri'05] approaches. In real MANET environment reaching 100% accuracy in true positive rate can not be achieve due to the characteristic of MANET described in Chapter 1.

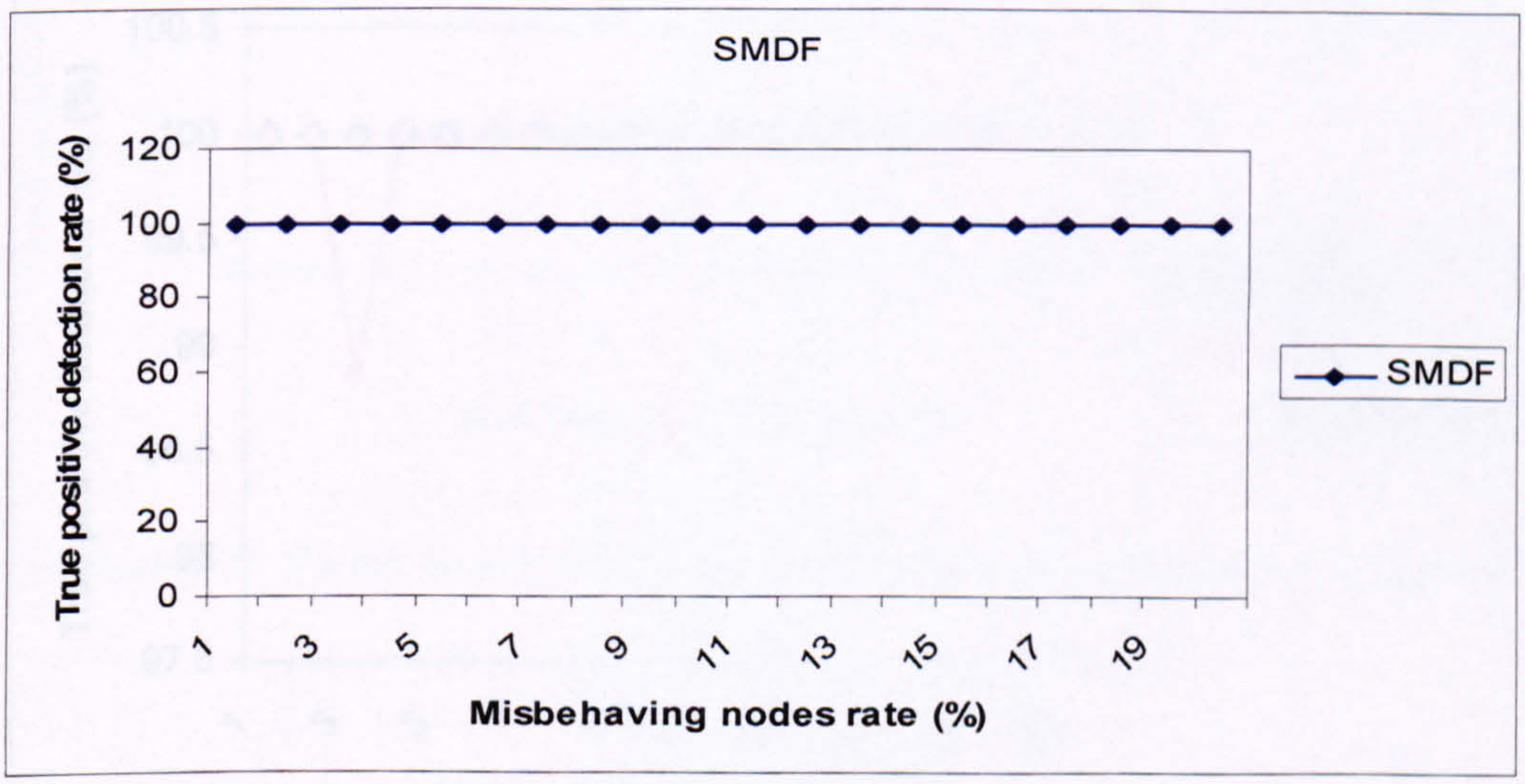



**Figure 6-3: True positive detection vs. Misbehaving rate in SMDF**

## 6.7 Comparison With Existing Approaches for (True Positive Detection Rate)

Having seen true detection positive rate results for our SMDF, we are now comparing it with other mechanisms. Figure 6-4 show a comparison between our proposed SMDF and the Watchdog [Marti'00] and Random Two Hop ACK [Djenouri'05]. We have used same simulation parameters mentioned in section 6.2 and run the simulations using each protocol separately (i.e. WD then Random Two Hop ACK). In addition, as in our SMDF simulation case we have set the POWER COTROL parameter to YES in order to see how the other two protocols perform. The result shows that our proposed SMDF outperformed Watchdog, which suffers from a sharp fluctuation between (98% - 100%), whereas SMDF remains constant at 100%. On the other hand, SMDF has as same true detection rated as the random Two Hop ACK. The pink line which represents WD fluctuated rapidly twice when the misbehaving rate reached 3% and 10% respectively. This is due to the fact that WD use promiscuous mode monitoring which can not detect nodes misbehaviour when employing power control technique. Therefore, in two occasions WD was below our SMDF and R2H ACK.

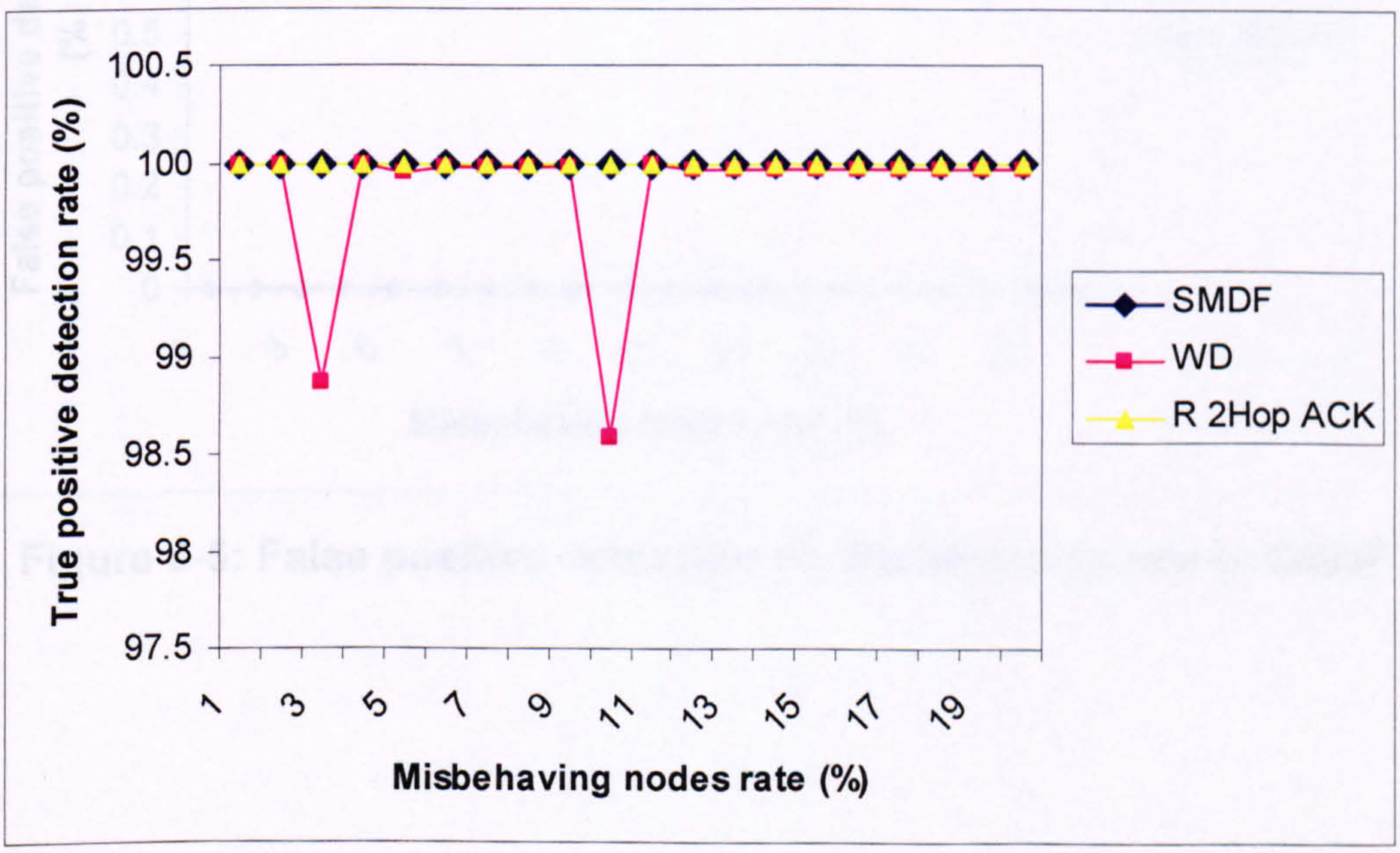



**Figure 6-4: Comparison of True positive detection vs. Misbehaving rate**

## 6.8 Evaluation of False Positive Detection Rate

This metric shows the number of well-behaved nodes falsely classified as misbehaving. The dropping rate remained at 50% as previous scenarios. This scenario gives us an idea whether our proposed SMDF unfairly accused well-behaved nodes of misbehaviour or not. Figure 6-5 shows that SMDF has 0% false positive detection rate in all of the 20 cases we simulate. This means that SMDF have never wrongly accused well behaving nodes in the network that forward packet correctly of misbehaviour. Next we validate this result by comparing it to other existing mechanisms. Again in real MANET environment reaching 0% accuracy in false positive rate can not be achieve due to the characteristic of MANET as well as other network conditions such as collisions, faulty nodes and low connectivity network.

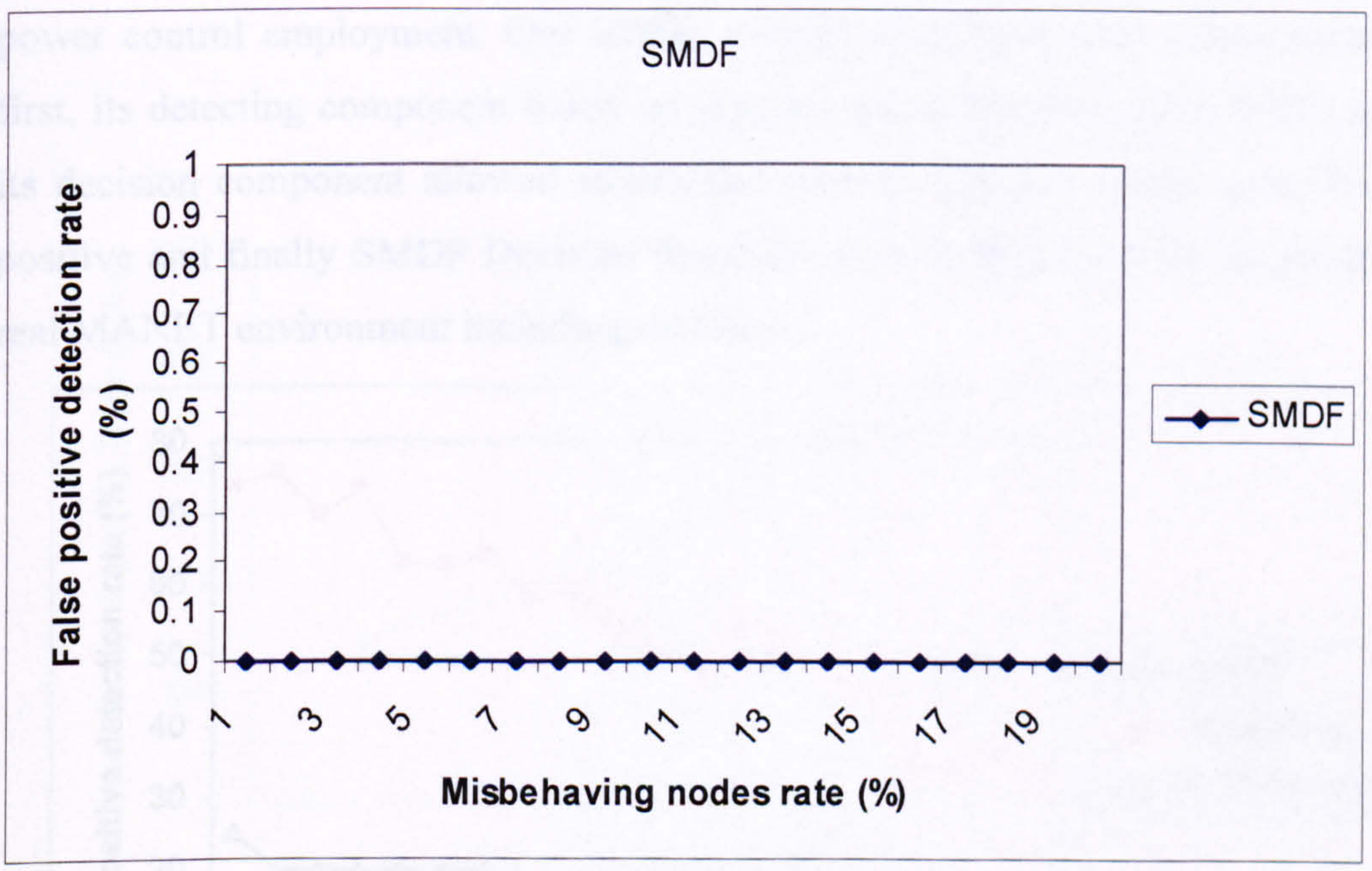



**Figure 6-5: False positive detection vs. Misbehaving rate in SMDF**

## 6.9 Comparison With Existing Approaches for (False Positive Detection Rate)

As we have done in the true detection positive rate comparison, we have compared our SMDF False Positive Detection Rate result with the Watchdog and the Random Two Hop ACK. Figure 6-6 show clearly the considerable advantages of SMDF over both the Watchdog and the Random Two Hop ACK in keeping the false detection rate steady at 0% level. As the highest false detection rate was produced by the Watchdog which was between (35% - 75%), the Random Two Hob ACK performed slightly better than the Watchdog as it fluctuates around 20%. The Watchdog suffered from such a high False Positive Detection Rate because its operation requires the nodes within a MANET to operate in promiscuous mode. As a result the Watchdog failed to detect the misbehaviour in cases of collisions, partial collusion, and power control employment. The Random Two Hop ACK on the other hand has much less False Positive Detection Rate in compare to Watchdog due to its used of power control employment. Our SMDF outperformed both approaches because of first, its detecting component based on sessions and not promiscuous mode, second its decision component allowed redemption before judgment resulting in 0% false positive and finally SMDF Decision threshold set to reflect as much as possible of real MANET environment including collisions.

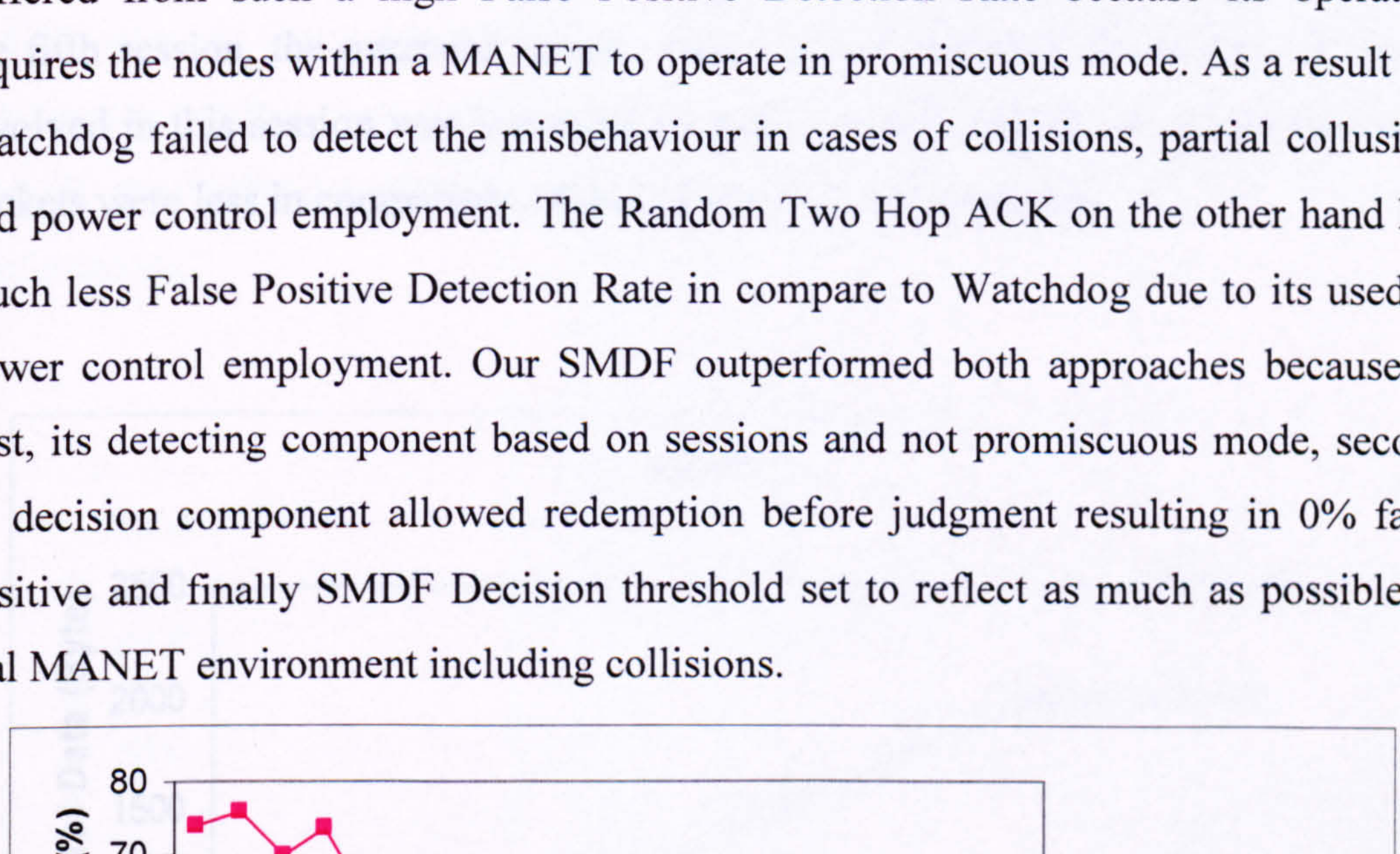



**Figure 6-6: Comparison of False positive detection vs. Misbehaving rate**

## 6.10 Evaluation of SMDF Communication Overhead

In this scenario we simulate the amount of communication overhead generated by our proposed SMDF. We measure the communication overhead in terms of control packets generated throughout the detection stage such as the FAPs packets. Figure 6-7 shows the overhead produced for the whole of the 4 sessions in the simulation. It can be seen that the amount of overhead increased gradually after the first session from just above 500 bytes to 2000 bytes at the fourth sessions. After the fourth session it remains stable at 2000 byte until the end of the fifth session. The gradual increasing of overhead from the first session to the fourth one in Figure 6-7 was due to the exchange of FAPs between the one hop neighbouring nodes in each session. In the fifth session, the overhead remain stable. This is because the number of nodes involved in this session was less than the other. As a result, the exchanges of FAPs packets were less in comparison of that in the other four sessions.

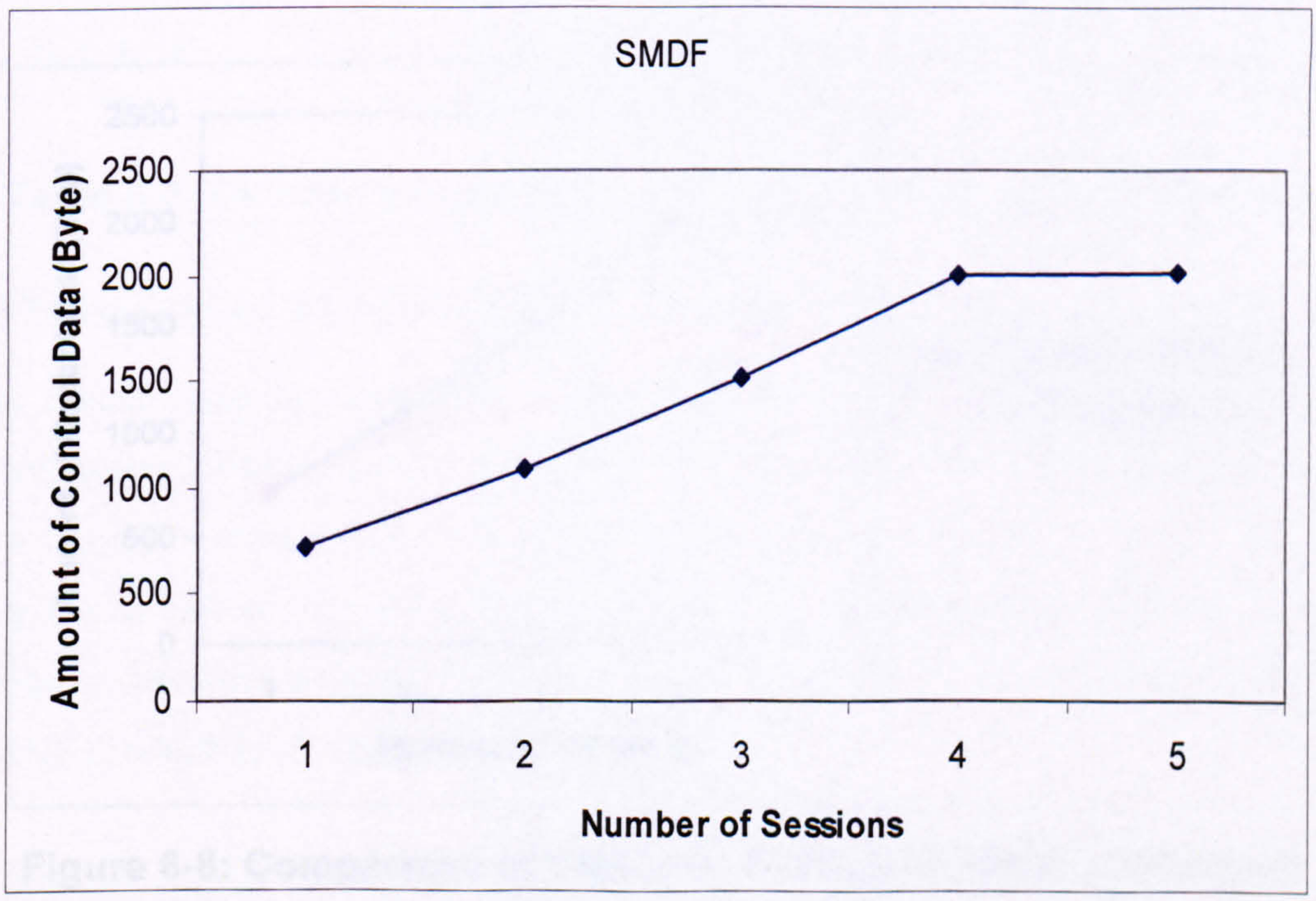



**Figure 6-7: SMDF Communication Overhead**

### 6.10.1 Comparison of SMDF vs. Optimised SMDF with Sessions Aggregation

In this scenario we compare the overhead produced previously by SMDF in figure 6-7 to the optimised SMDF described in chapter 5. The optimised SMDF uses sessions aggregation approach. When using this approach, nodes that are involved in more than one session could wait a certain time until all sessions end before sending the FAP to their direct neighbours. Figure 6-8 shows the comparison between the aggregated SMDF and non-aggregated SMDF in terms of the overhead produced. The figure shows a significant reduction in the amount of communication overhead produced by the aggregated SMDF in comparison with that in non-aggregated SMDF. It can be seen form Figure 6-8 that in aggregated SMDF and non-aggregated SMDF, after the fourth session the overhead remains stable. Again, we believe this is because the number of nodes involved in the fifth session was less than the other. As a result, the exchanges of FAPs packets were less in comparison of that in the other four sessions.

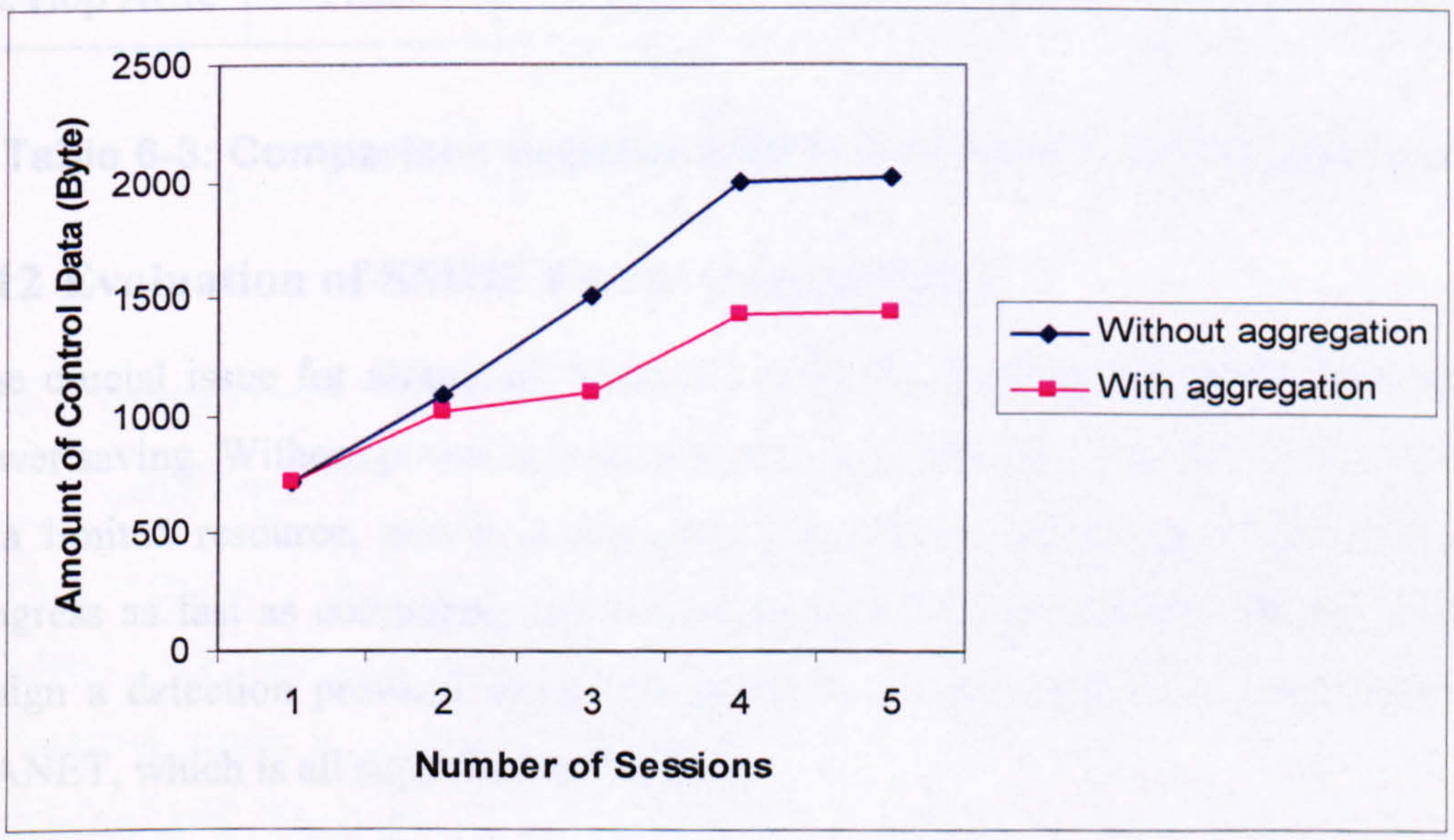



**Figure 6-8: Comparison of SMDF vs. Optimised SMDF (Overhead)**

## 6.11 Comparison of Overhead Reductions Between SMDF and Random Two Hop-ACK

In this scenario, we evaluate SMDF by comparing it with Random 2 Hop ACK in terms of the communication overhead. We will compare both SMDF non-aggregated and aggregated version with Random 2 Hop ACK. Table 6-3 below shows the amount of overhead produced in each mechanism for the 5 sessions we simulate. It can be seen clearly that Random 2 Hop ACK produces a considerable amount of communication overhead compared to both SMDF mechanisms (i.e. aggregated and non-aggregated).

| | **Overhead Amount (Bytes Per Session)** | | | | |
|---|---|---|---|---|---|
| **Sessions** | **1** | **2** | **3** | **4** | **5** |
| SMDF | 724 | 1088 | 1512 | 2012 | 2028 |
| Optimised SMDF | 724 | 1022 | 1096 | 1436 | 1450 |
| R 2 Hop ACK | 11260 | 13300 | 20520 | 42800 | 43860 |

**Table 6-3: Comparison between SMDF and R 2Hop ACK (Overhead)**

## 6.12 Evaluation of SMDF Power Consumption

One crucial issue for almost all kinds of MANET supported by battery powers is power saving. Without power, any mobile device will become useless. Battery power is a limited resource, and it is expected that battery technology is not likely to progress as fast as computing and communication technologies do. Hence, how to design a detection protocol using less power is an important issue, especially for MANET, which is all supported by batteries.

In this scenario we have simulated the amount of energy measured in milliwatt hour (mWhr) (i.e. $10^{-3}$ W). In our proposed SMDP we measured the energy produced with various rate of misbehaviour starting from 1 to 20 misbehaving nodes as shown in figure 6-9. It is apparent that, as the misbehaving nodes increase the energy decrease. This is due to the fact that misbehaving nodes are dropping packets that they should

## 6.11 Comparison of Overhead Reductions Between SMDF and Random Two Hop-ACK

In this scenario, we evaluate SMDF by comparing it with Random 2 Hop ACK in terms of the communication overhead. We will compare both SMDF non-aggregated and aggregated version with Random 2 Hop ACK. Table 6-3 below shows the amount of overhead produced in each mechanism for the 5 sessions we simulate. It can be seen clearly that Random 2 Hop ACK produces a considerable amount of communication overhead compared to both SMDF mechanisms (i.e. aggregated and non-aggregated).

| | **Overhead Amount (Bytes Per Session)** | | | | |
|---|---|---|---|---|---|
| **Sessions** | **1** | **2** | **3** | **4** | **5** |
| SMDF | 724 | 1088 | 1512 | 2012 | 2028 |
| Optimised SMDF | 724 | 1022 | 1096 | 1436 | 1450 |
| R 2 Hop ACK | 11260 | 13300 | 20520 | 42800 | 43860 |

**Table 6-3: Comparison between SMDF and R 2Hop ACK (Overhead)**

## 6.12 Evaluation of SMDF Power Consumption

One crucial issue for almost all kinds of MANET supported by battery powers is power saving. Without power, any mobile device will become useless. Battery power is a limited resource, and it is expected that battery technology is not likely to progress as fast as computing and communication technologies do. Hence, how to design a detection protocol using less power is an important issue, especially for MANET, which is all supported by batteries.

In this scenario we have simulated the amount of energy measured in milliwatt hour (mWhr) (i.e. $10^{-3}$ W). In our proposed SMDP we measured the energy produced with various rate of misbehaviour starting from 1 to 20 misbehaving nodes as shown in figure 6-9. It is apparent that, as the misbehaving nodes increase the energy decrease. This is due to the fact that misbehaving nodes are dropping packets that they should

forward to other nodes. This will result in less transmission of data packets in the network, as a result less power usage. Next we will compare our SMDF energy result with other mechanisms to evaluate it. As the number of misbehaving nodes is 0, SMDF power consumption was just below 450.04 (mWHr) and the network is fully working and all nodes forwarding packets correctly. As the number of misbehaving node increased the power usage decreased with it gradually, as explained above.

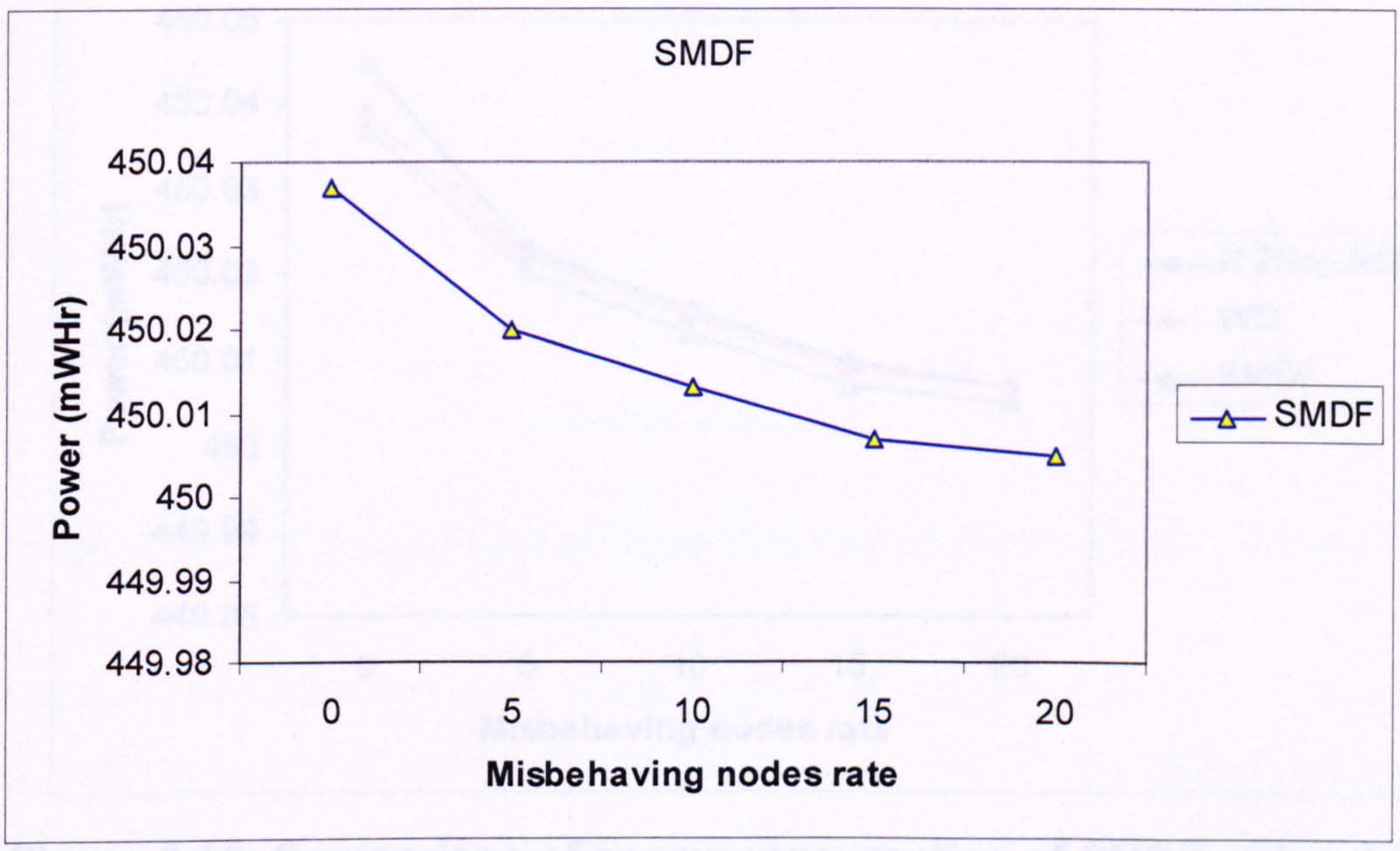



**Figure 6-9: Evaluation of SMDF Power consumption**

## 6.13 Comparison With Existing Approaches for (Power Consumption)

Having seen the SMDF power consumption results in previous section, we now compare them with the Watchdog and Random Two Hop ACK results. The comparison results in figure 6-10 shows that our SMDF clearly outperform both the Watchdog and Random Two Hop ACK in saving energy with less power consumption. There is a very small difference between the Watchdog and Random Two Hop ACK, with slight advantages to Watchdog. This could be due the huge amount of overhead that the Random Two Hop ACK generates. It can be seen from figure 6-10 that when no misbehaviour occur (i.e. the number of misbehaving nodes is 0) SMDF power consumption was just below 450.04 (mWHr) less than both the

Watchdog and Random Two Hop ACK. This is because in SMDF the exchanging of information and control packets between nodes takes place only and the end of the session resulting in saving energy. In contrast, the watchdog exchange of information and control packets happen every each hop, and in Random Two Hop ACK it happen every each two hop resulting in more power consumption than SMDF.

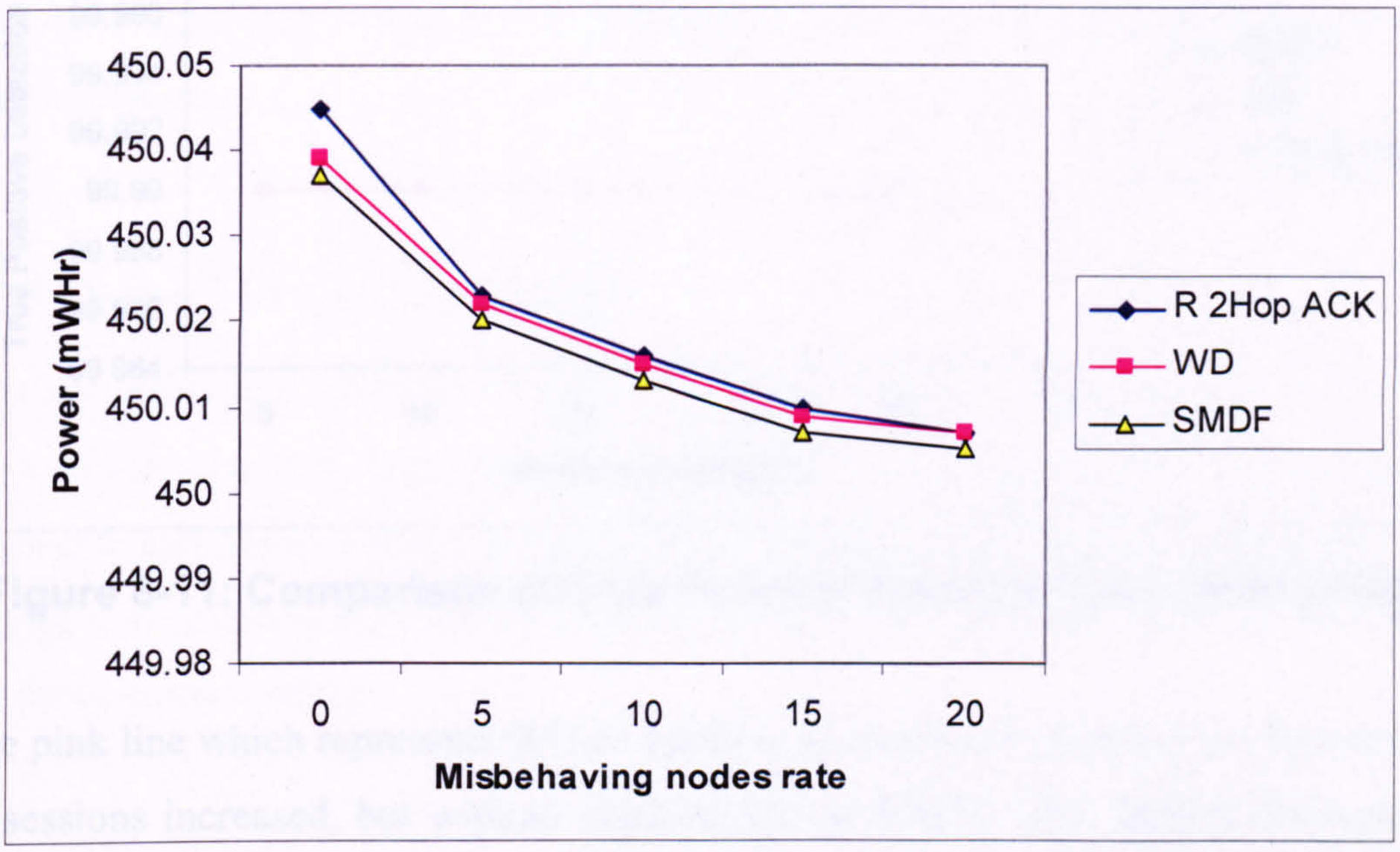



**Figure 6-10: Comparison of power consumption of SMDF with existing approaches**

## 6.14 Comparison with Existing Approaches for Scalability

Our proposed protocol SMDF has already been evaluated using 100 nodes, which is higher than 50 nodes average used in many other existing mechanisms evaluated using simulation. Since the scalability property is one of the desired characteristics especially in wireless sensor network, we have increased our previous number of nodes to 500 to examine our protocol. The main difference between small and large networks is the average path lengths (e.g. 3-4 hops in small network vs. 8-13 hops in large network). We have increased the network sessions from 5 to 50 sessions to reflect the increase in the number of nodes. If our SMDF achieve same accuracy and rate of true and false positives as with the previous 100 nodes scenarios, then we will consider our framework as scalable. It can be seen from figure 6-11 that SMDF is

scalable and still has the true positive detection rate at 100%. Figure 6-12 shows that SMDF has the false positive detection rate at 0% compared with the WD and Random 2 Hop ACK.

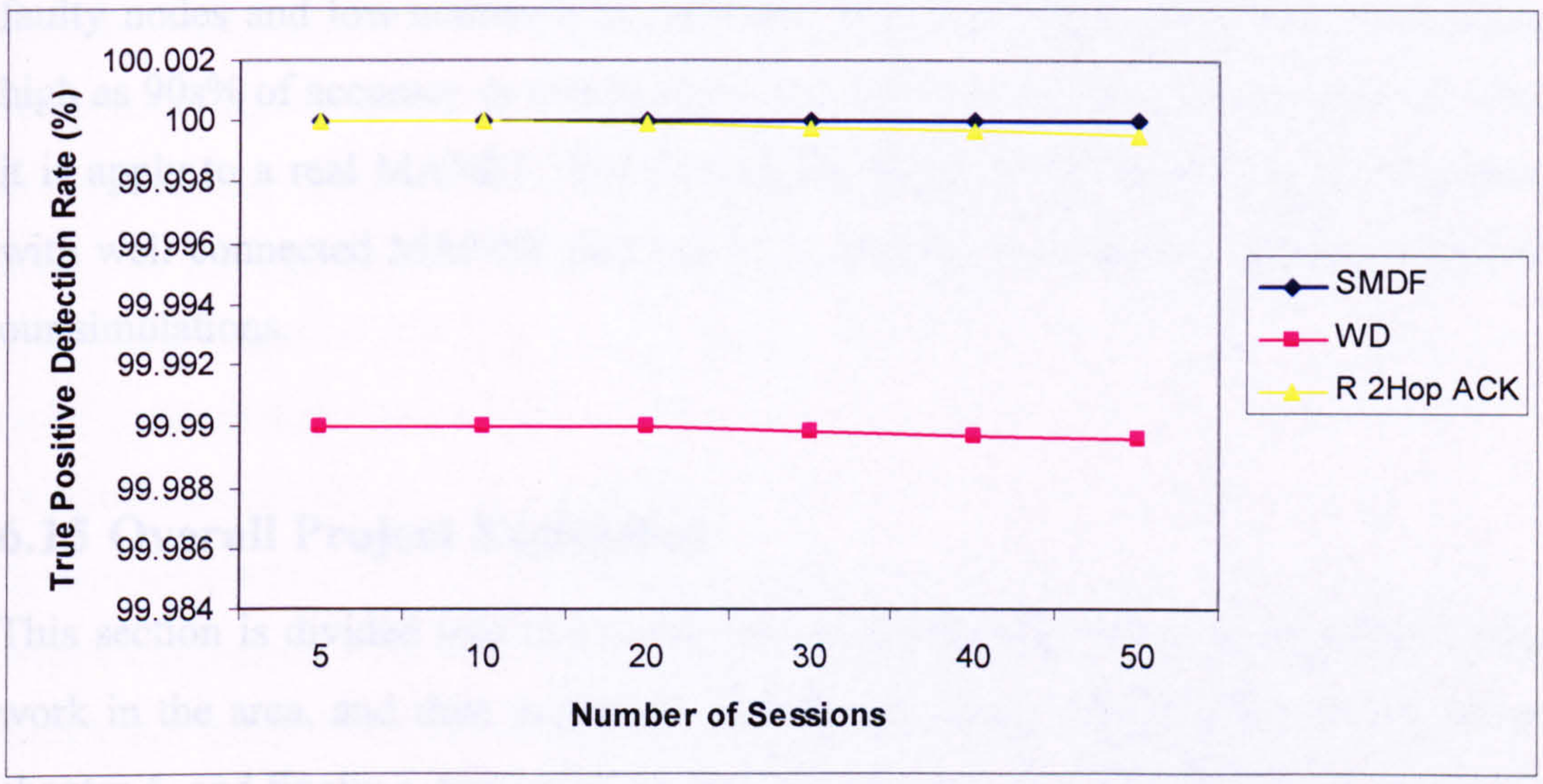



**Figure 6-11: Comparison of True Positive Detection Rate (Scalability)**

The pink line which represents WD in figure 6-12 decreased gradually as the number of sessions increased, but without reaching 0% as SMDF. The gradual decrease of the WD was because as the number of nodes increase to 500 the conectivety of the network increase. As a result the use of promiscuous mode by WD will be more effective in terms of false detection when the network is well connected.

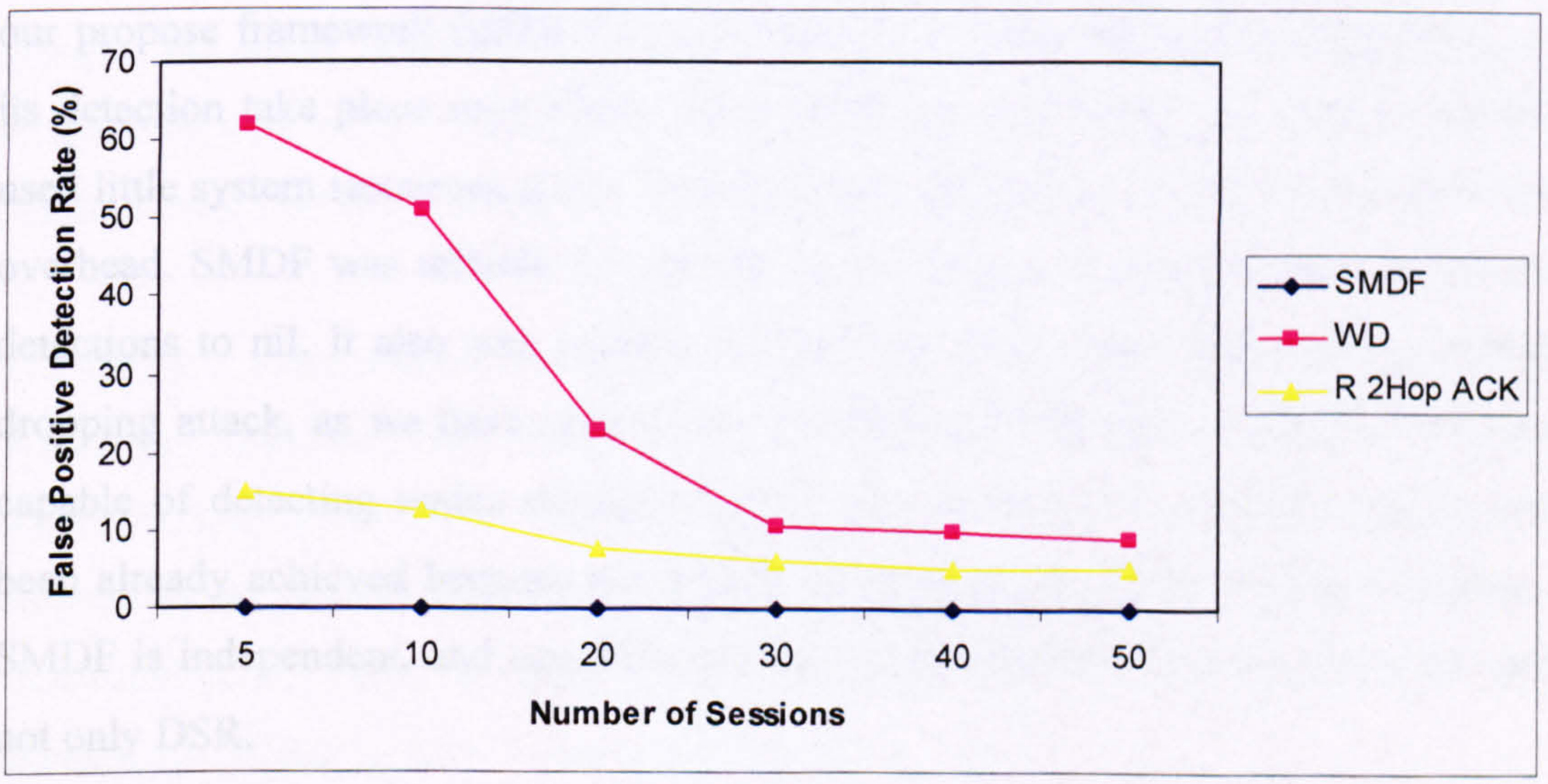



**Figure 6-12: Comparison of False Positive Detection Rate (Scalability)**

As mentioned previously, in a real MANET environment reaching 100% accuracy in true positive and 0% of false positive rates can not be achieve due to the characteristic of MANET as well as other network conditions such as collisions, faulty nodes and low connectivity network. We believe that SMDF can achieve as high as 90s% of accuracy in true positive and as low as 5-10% of false positive when it is apply to a real MANET. We also expect that our SMDF will work effectively with well connected MANET that has large number of nodes as we have shown in our simulations.

## 6.15 Overall Project Evaluation

This section is divided into two parts, first a comparison of our work with existing work in the area, and then evaluation against our initial requirements mentioned in chapter 4, and finally a discussion of the shortcomings of the SMDF. The aim here is to take a very broad view of the research and look at the overall achievements and the problems remaining.

### 6.15.1 Evaluation Against Our Initial Requirements

Our simulation result shows that our proposed framework succeeded in fulfilling our initial requirements we presented in chapter 4. This has been achieved as follows; our propose framework SMDF has used an active acknowledgment mechanism, as its detection take place only when data packets are sent during the session. SMDF used little system resources at low cost, and has produced very low communications overhead. SMDF was reliable in minimizing the false positives and false negatives detections to nil. It also was capable of detecting lying nodes that perform partial dropping attack, as we have seen in our simulation results above. SMDF was also capable of detecting nodes misbehaviour in the presence of collisions, which has been already achieved because our SMDF employing the power control technique. SMDF is independent, and can work with all of the MANET routing protocols and not only DSR.

The simulation results already showed that SMDF is scalable and can work to support up to 500 nodes to reflect stationary wireless sensors network. The simulation shown also that SMDF can integrate with other mechanises and routing protocols such as DSR. Our SMDF performed availability by reaching all necessary recipients nodes involved inside each session in the network. Appropriate authentication, integrity and confidentiality, of the detection mechanisms have been achieved through digital signatures and cryptography primitives (i.e. asymmetric encryption) used by the FAP packets in our SMDF. Our SMDF also support cross-layer collaboration between the session layer and the network layer in order to reach valuable optimisations. Through simulations our SMDF was fair, and precise in determining exactly the misbehaving nodes, and has high rate in its true positive which considered as a novel achievement. Our SMDF has an isolation component which has been evaluated and it successfully isolate misbehaving nodes from the network to prevent harming it again.

### 6.15.2 Comparison with Related Work

As mentioned before, the main objective of our misbehaviour detection framework is to provide a set of components and mechanisms that can detect misbehaviour and mitigated at low energy and communication overhead cost but with high accuracy. The problem of node misbehaviour in MANET has been treated by many research groups, and many mechanisms have been proposed. Our framework shares some similarities with prior work carried out in other projects. In this section we compare our framework with these works.

The first and most famous mechanism in misbehaviour detection in MANET is The Watchdog [Marti'00]. We have compared our SMDF results with the Watchdog results on five of our six different metrics using simulation, and found that SMDF outperform the Watchdog in four of these metrics. The five metrics are True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. Where as the Watchdog has slightly lower overhead than SMDF, but only when there is no misbehaviour in the network. Moreover, all of the other Watchdog drawbacks including partial dropping do not exist in our SMDF.

There are many other detection mechanisms especially the reputation mechanisms such as [Buchegger'02b, He'04, Michiardi'02a, Miranda'03, Yang'02] using the watchdog as their main monitoring component. Consequently, they inherited all the drawbacks that the watchdog suffers, even though their other system components are efficient. This gives our SMDF clear advantages over all of the mechanisms that adopting the watchdog concept in their detection system.

We have also compared our framework with the other types of mechanism that do not use the watchdog as their monitoring component. The most recent solution of these is the Random Two-Hop ACK [Djenouri'05]. Our comparison through simulation showed us that our SMDF outperforms the Random Two-Hop ACK in four of our six simulation metrics. These matrices are Communication Overhead, False Positive Detection Rate, Power Consumption Rate and Scalability. Although, we have similar True Positive Detection Rate as the Random Two-Hop ACK. However, through our simulation comparison we noticed that the Random Two-Hop ACK failed to detect partial dropping in many occasions, and that it can often detect the full dropping case more than the partial one.

Our framework evaluation and comparison with other existing mechanisms shows it performs better and has novel aspects that do not exist in other mechanisms.

### 6.15.3 Discussion

Whilst the SMDF solves an interesting problem with some novel aspects there remains several shortcomings. This section outlines these problems.

Waiting until the end of all sessions to check node misbehaviour reduce the communication overhead considerably as we have seen in the simulation results. However, it will increase the delay before detection. It is a trade-off issue as reducing the cost is more valuable and important than increasing the waiting time. In some application such as video streaming, it is important to detect misbehaviour immediately as it occurred and not wait until the end of all sessions involved in the

network. This shortcoming can be reduced by waiting until the finishing of the first session only, and not all of the sessions.

The SMDF assumes that nodes only drop data packets and not control packets. If the nodes drop the control packets, the SMDF can not detect them, as it only deals with data packets dropping. The increased amount of control packets is not preferable due to the overhead they generate. However, moderate number of control packets is important, and as such, dropping them will affect the performance of the network. Selfish nodes drop both data and control packets whereas malicious node targeting mostly data packets. This is a complex problem and the solution is not obvious. However the design of the framework is such that it would support the addition of a new component to enhance and/or complement the existing one.

We intentionally examine the situation when MANET is stationary and the wireless sensor network is static, as this was the main target investigation of this research. However, it is possible to have mobile sensor nodes. In mobile sensor networks, nodes move freely to get necessary information about a certain event that moves in the nature, such as toxic gas cloud or a radioactive mobile object. In mobile wireless sensor networks applications, the user is also interested in high-level description information about the tracked event. It would be interesting to see how SMDF can be enhanced to support mobile scenarios in different types of both mobile wireless sensor network and mobile ad hoc network. Again, this is a difficult problem and solutions may take the form of Intrusion Detection modules or interaction with the underlying routing schemes to detect mobility.

In our SMDF Decision component we have used a fixed threshold of tolerance over which node will be judged as misbehaving. This is fine for a stationary MANET and static WSN scenarios. However, it is more efficient to use variable threshold which can change according to the network topology and scenarios. The threshold can be also estimated empirically for each network by first, running simulations with no misbehaving and calculate the threshold at each node for different scenarios that

estimate the network. Then, retrieving the maximum value in all scenarios from the decision point and then consider it final threshold.

In some application such as the battlefield or the rescue operations there is no need to run all of the three components in every node. For example isolating and punishing such nodes would not be beneficial. In this case there is no need for the isolation component and it need to be switch OFF. Therefore, deciding when to turn components ON/OFF is another challenges need to be resolved. One suggestion is to add a separate component with 'intelligent' as decision capability to SMDF in order to deal with such situation. More efficient suggestion would be to add to each component of SMDF this intelligent capability to decide itself when and where to function according to the network and nodes status.

For the purpose of evaluation and comparison with other approaches we have used simulations techniques. Performance evaluation through simulations is helpful but will not reflect the reality 100%. For example, in a real MANET environment reaching 100% accuracy in true positive and 0% of false positive rates can not be achieve due to the characteristic of MANET as well as other network conditions such as collisions, faulty nodes and low connectivity network. It will thus be fascinating to see the actual performance of our complete SMDF framework by integrating every component that it consisted of. By doing that we could measure new parameters that will add more understanding of the reality and that can not be performed clearly through simulations. Once the above tasks have been completed successfully, it would be interesting to implement the complete model in an experimental test-bed to see its practical feasibility. This task appears feasible in the near future as the prices of advanced sensors and handheld devices are already decreasing gradually.

## 6.16 Summary

In this chapter, we have described our simulation parameters and identified our simulation metrics that we measured in order to simulate our proposed framework

and evaluated it using simulation techniques, namely GloMoSim. The simulation metrics we used are Throughput, Overhead, True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. Using simulation to carry out experiments evaluation, we showed the significant consequence of node misbehaviour in reducing MANET throughput.

Our proposed framework worked effectively and can be used as suitable and efficient security mechanism for ad hoc network and its special kind static wireless sensors network. Our framework achieved the highest True Positive Detection Rate compared to other existing approaches, which mean it has the highest success rate in detecting misbehaving nodes. It also achieved the lowest False Positive Detection Rate compared to other existing approaches, which means the lowest rate of wrongly misjudging well behaved nodes. SMDF also produced low communication overhead rate as well as less usage of energy compared with other existing approaches.

We analysed SMDF system in the context of other existing systems namely, the Watchdog and the Random Two Hop ACK services, and quantified its performance benefits. There are many other detection mechanisms especially the reputation mechanisms such as [Buchegger'02b, He'04, Michiardi'02a, Miranda'03, Yang'02] that used the watchdog as their main monitoring component. Consequently, they inherited all the drawbacks that the watchdog suffers, even though their other system components are efficient. This gives our SMDF clear advantages over all of the mechanisms that adopting the watchdog as their detection system.

Our framework is also shown to have lower overhead than Random Two Hop ACK. Moreover, SMDF has the lowest percentage (i.e. 0%) of False Positive Detection Rate compared to both the Watchdog and the Random Two Hop ACK systems. It also shows high percentage of True Positive Detection Rate among other schemes. Furthermore, it shows that it is scalable, as we increased the number of nodes from 100 to 500 nodes.

Finally, we took a very broad view of the research and look at the overall achievements including evaluation against our initial requirements specified in chapter 4 and discussed the problems remaining. Our simulation result shows that our proposed framework fully succeeded in fulfilling all framework initial requirements that we presented in chapter 4. Whilst the SMDF solves an interesting problem with novel aspects there remains several shortcomings. These include that SMDF assumes that nodes only drop data packets and not control packets. Also include the delay before detection in terms of waiting until the end of all sessions to check node misbehaviour.

The next chapter is our conclusions and future works chapter, and it is the final chapter of this thesis.

# 7. CONCLUSIONS AND FUTURE WORK

This thesis has presented a new framework for detecting misbehaviour in Mobile Ad hoc Networks and Wireless Sensor Networks. The new framework is an integration of the novel components that we developed during our investigation. The new system aims at providing reliable detection mechanisms that achieving higher levels of accuracy while being at low cost and simple to implement.

This chapter is the conclusion of our work and summary. The chapter is organized as follows. First we present a summary of the thesis in section 7.1. Our main contributions and a summary of the SMDF framework and the new components associated with it are presented in section 7.2. Then a comparison of SMDF with existing approaches discussed in section 7.3. Future work is investigated and proposed in section 7.4, and finally our concluding remarks are provided in section 7.5.

## 7.1 Thesis Summary

Security in Mobile Ad Hoc Networks and Wireless Sensors Network is one of the most important concerns because these systems are more vulnerable to attacks than a wired or infrastructure-based wireless network. Designing an effective security protocol for MANET is a very challenging task. This is mainly due to the unique

characteristics of MANET, namely shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among users, limited availability of resources, and physical vulnerability.

Due to these infrastructures-less features of MANET mentioned above, network-organisation functionality must be performed by the nodes. In particular, data packets sent between distant nodes are expected to be relayed by intermediate nodes, which act as routers and provide the forwarding service. The forwarding service relays the received packets from node to node until reaching their final destination, following routes selected and maintained by the routing protocol. These services (routing and data forwarding) together are at the core of the network layer. All of the existing research solutions in the area of data forwarding detection are only focusing on mobile scenarios in MANET (i.e. when nodes are freely mobile). However, little has been done in the terms of examine and applying such mechanisms to stationary MANET when nodes are static or with very low level of mobility. An example of stationary MANET is a wireless sensor networks for civil and military applications (e.g. security management, surveillance, automation, wildlife and environmental monitoring) that are typically deployed today, and have small to medium scale (tens to hundreds of sensors) a cross small to medium geographical distances. Since every node is potentially a router, this adds new vulnerabilities to the network-layer problems experienced on the Internet. Detection and routing protocols must be simple enough to scale up to large networks such as stationary wireless sensors networks, yet robust enough to cope with failures that occur many hops away from a source.

Our work focuses on the design of a low cost Sessions-based Misbehaviour Detection Framework, SMDF, to detect node misbehaviour in terms of date packets drooping that occur in stationary MANET and wireless sensors network. Our novel system targeting most common attacks that MANET and WSN, suffered namely black-hole and data packet dropping attacks. The breakdown of the thesis is as follows:

Our introduction to the area in Chapter 1 discussed the wider context and outlines the problem of node misbehaviour in Mobile Ad Hoc Networks. It outlined the definition of MANET and its main applications. It also identified MANET's main characteristics that make the design of routing and detection protocol of such kind of network challenging. These characteristics include: (1) Multi-hop routing, (2) Dynamic Network Topology (3) Autonomous terminal (4) Distributed operation (5) Fluctuating link capacity (6) Light-weight terminals. Chapter 1 also highlighted the consequences of node misbehaviour in MANET and its impact on MANET performance.

An overview of MANET architecture and its security issues described in Chapter 2. It first discussed the security issues related to MANET Network layer. It also described two of MANET main routing protocols namely Dynamic Source Routing (DSR) and Ad hoc On-Demand Distance Vector (AODV). We then highlighted the security threats and attacks on routing protocols and list most common attacks such as Denial of Service attack (DoS). Next we moved down one step to the MAC layer and discuss its security issues in MANET, which includes misbehaviour issues in the channel access. At the end of this chapter we listed the main security attributes and considered some methods of achieving them including cryptography and digital signature.

We surveyed the literature and related works relating to MANET misbehaviour in Chapter 3. We presented and discussed the existing solutions that aim at detecting misbehaviour on packet forwarding when it appears in the network. These solutions classified into two main techniques, Reactive and Preventive. The reactive solutions split up into two main classes, monitoring and reputation-based solutions. The monitoring class includes basic approaches that focus on the monitoring phase and suggest techniques to control the forwarding process. Reputation-based solutions propose mechanisms to isolate the nodes detected as selfish. However, these

solutions incorporate monitor components that use only promiscuous monitoring techniques. On the other hand, preventive techniques proactively try to mitigate the misbehaviour or its effects, either by motivating nodes to cooperate or by taking measures to prevent packets from being dropped before sending them. This chapter pointed out the main drawbacks of the existing work and issues that need to be addresses as:

- Most of the existing approaches have high cost in terms of the amount of communication overhead produced.

- Existing detection mechanisms are not independent of the routing protocol and operate as an extension of it. Many mechanisms work only with one particular routing protocol.

- Existing monitoring/detection solutions suffer from the post-detection drawback, in terms of punishment and selfish nodes knowledge exchange between nodes.

- Most of the existing mechanisms use monitoring approaches that depend on promiscuous monitoring, which has many drawbacks regarding the accuracy on detections, especially when employing the power control technique.

- Techniques using proactive approaches trust all nodes and do not prevent nodes from overloading the network, thus they can not work effectively alone and they require to be combined with a reactive monitoring technique.

- Most of the existing solutions are applicable only for a small MANET with limited number of nodes and as such scalability has not been addressed, especially when dealing with wireless sensors network which has a large number of nodes.

- Energy saving has not been considered properly, and as such many existing approaches have a high-energy consumption.

In Chapter 4 we presented our framework Sessions-based Misbehaviour Detection Framework (SMDF) and briefly described its three components. First we have described our research objectives that form a comprehensive set of support mechanisms and schemes. We discussed the gaps in the current knowledge that this thesis will address in the requirements review. We identified requirements, issues and challenges important when designing an effective misbehaviour Detection Framework in stationary MANET and static wireless sensor network. Then we have presented our SMDF and its cross layer collaboration between the session layer and the network layer.

In Chapter 5 the different components of our framework have been fully described in detail. We started with our detection component through the new Sessions-based Misbehaviour Detection Protocol (SMDP), where we explained the concepts of the monitoring that our protocol used, and the algorithm to do so. We have given two case studies in order to illustrate how our SMDP work. Then we explained our modified Bayesian approach for the decision stage. This was followed by description of our Isolation component where misbehaving nodes are penalised for their attacks.

Chapter 6 presented the simulation design, the analysis, results and performance evaluation. We show how selfish misbehaviour can badly affect the performance of MANET in terms of reducing the throughput. We then evaluated our system through simulation with two of the well-known existing systems in terms of six different metrics. We then evaluated our new protocol against the initial requirements listed in chapter 4, and proven that it achieved all of them.

Finally, suggestions for future work and conclusions are presented in this current chapter.

## 7.2 Research Contributions

This thesis contributes to our understanding of Security and Misbehaviour Detection Systems in wireless ad hoc environments in the following ways:

- Our first contribution is to provide a set of requirements for an efficient misbehaviour detection framework in mobile ad hoc networking environments and examine these against existing research in literature [Fahad'06a]. These requirements enable the network providers to operate a secure system whilst consuming low energy and producing low communication overhead. The requirements are similar to those of existing work but have been reconsidered to reflect the changing nature of ad hoc networks, especially as it applies to sensor networks. A survey of research literature in the field revealed that no results completely meet these requirements. These techniques focus on either high accuracy detection rate at huge cost in terms of energy and communication overhead or on poor accuracy detection rate at a medium cost. Others fail because they aim to encourage good behaviour among nodes without fair and firm mechanisms to deal with those who misbehave. Additionally we also bring together relevant ideas of use in search for effective misbehaviour detection in MANET environments.

- Using the set of requirements and inspiration from relevant literature, this thesis proposes a novel solution to accurately and effectively detect and deal with node misbehaviour in mobile ad hoc networking environments including wireless sensor network, and it is called Sessions-based Misbehaviour Detection Formwork (SMDF) [Fahad'07a, Fahad'07b]. The new framework consists of three components, Detection Component, Decision Component and finally Isolation Component. Each component in SMDF provides different functionalities, and all of these components integrated to provide efficient and robust solution against node misbehaviour in MANET. The major advantages of our SMDF included its capability of working either independently or integrating with other routing protocols. The new

framework also extensible and flexible as it has the capability of adding new components to it or removing existing components from it as necessary. Moreover, the new framework is transparent in terms of integrating with other mechanisms as required. The detection component contains our novel Sessions-based Misbehaviour Detection Protocol SMDP to detect selfish or malicious nodes that drop packets partially or completely to launch either black-hole or data dropping attacks. For the decision component we have enhanced an existing Bayesian approach to decide whether the node deliberately misbehaved or not. For the Isolation component, we have modified an existing approach and used an Observation-Based Protocol to isolate misbehaving nodes. It uses neighbouring observations experience to isolate misbehaved nodes. We analysed and evaluated the proposed framework by simulation techniques. Our evaluation was focused on six important parameters, namely Throughput, Overheard, True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. By comparing our results to those of other mechanisms available on literature, we showed that our solution has low cost in terms of communication overhead than other approaches. We showed also that our framework has the lowest False Positive Detection Rate amongst other approaches, and that it has highest value of True Positive Detection Rate compared with other approaches. Our evaluation also showed that our solution has lower energy consumption rate compared with other existing approaches. The experiments showed also that our framework is scalable and can work with higher number of nodes, especially in wireless sensor networks.

- The first new component of our framework we have developed is the Detection Component, which is the most important component in any misbehaviour detection framework. For this reason we have developed a new novel Sessions-based Misbehaviour Detection Protocol (SMDP) [Fahad'06b, Fahad'07a, Fahad'07b]. The SMDP deals with the network in terms of sessions, and uses cross-layer collaboration between the session layer and the

network layer in order to know the start and the end of each session. In SMDP each node in the route session monitors all of its direct neighbours within a one hop communication, and checks whether they correctly forward packets or drop them completely or partially in order to lunch an attack such as black-hole and packet dropping attacks. SMDP is cost effective as it reduces the communication overhead, by using only one hop communication (no flooding), and sending control packets only at the end of sessions, instead of doing so for each packet, contrary the current solutions that exist in literature. The new SMDP also has an advantage of being independent of the routing protocol, as well as its ability to work with any MANET routing protocol, unlike most of the existing mechanism in literature who work as an extension of one particular routing protocol. We evaluated the proposed protocol by simulation and showed that our approach is more efficient and scalable than other approaches found in the literature. It showed also that it has a low cost in terms of both communication overhead and energy consumption compared to other approaches.

- We have developed the second component of our framework (SMDF) which is the Decision Component [Fahad'07a, Fahad'07b]. After detecting those nodes dropping packets using our detection component SMDP, we have used the decision component of the SMDF in order to decide whether the nodes misbehaved or not. As the nodes might drop packets for innocent reasons such as collision or faulty packets, the decision component of SMDF take all of this into account, in order to make a fair decision. For the decision component we have enhanced an existing mathematical estimation method, which has been used in the literature and we have modified it effectively to suit our new framework requirements. It based on Bayesian standard approach, which consists of estimating a parameter the observations of which follow a Bernouli distribution by a Beta distribution. In our approach, well behaving nodes improve their reputation, whereas misbehaving nodes in terms of either intentional or unintentional packet dropping will decreases it. Moreover, our approach allows redemption before making decisions, and

decreases false accusations due to, for example channel conditions or collision. Furthermore, in Bayesian approach only the latest observations are watched over, and not all the observations, as a result it has the advantage of not requiring a memory. We evaluated our proposed approach by simulations and showed that our approach is more accurate in identifying the real misbehaving nodes than existing approaches. It also has lower communications overhead compared to other approaches.

- Having identified the misbehaving nodes locally, we developed our Isolation Component which will then punish them by not routing packets through them and by not forwarding packets for them. For this component, we have modified an existing approach and we used an Observation-Based Protocol to isolate misbehaving nodes. Once a node is judged locally as misbehaving by some other node, this latter must approve its detection to ensure the isolation by all nodes. The Observation-Based Protocol uses neighbouring observations experience to mitigate false detections and false accusations vulnerabilities that exist in other approaches. In this protocol, a node that detects and accuses another as misbehaving must approve its accusation before taking any measure against it. It should not isolate the assumed misbehaving unilaterally, because this could result in false detections against it. However, it could avoid routing its own packets through this node in all cases. The Observation-Based Protocol enforces the accusing node to collect a certain number of observations from neighbouring nodes in terms of signatures before isolating the detected node. Once the accuser node collects this number, it broadcasts an Isolation packet including all observations through the network to isolate the misbehaving node. This broadcast will not be performed until a node is detected and approved as misbehaving. As a result, our solution produces less overhead as long as nodes well-behave, as no opinions are exchanged periodically. Our simulation results suggest that our approach has the lowest percentage in falsely accusing well behaving nodes of misbehaviour compared to other existing approaches. It showed also that it has a low cost in terms of communication overhead.

- Our final contribution is that this research poses some new questions that had not been made explicit before. Among the questions for further work are issues of tackling detection complications in hybrid ad hoc network environments. Two other important issues raised are those of dealing with control packets dropper and mobility handling issues in terms of mobile Wireless Sensor Networks. These questions are examined together with an evaluation of the project in terms of the shortcomings of the framework and comparison with closely related work.

## 7.3 Comparison with Existing Approaches

As mentioned before, the main objective of our misbehaviour detection framework is to provide a set of components and mechanisms that can detect misbehaviour and eliminated at low energy and communication overhead cost but with high accuracy. The problem of node misbehaviour in MANET has been treated by many research groups, and many mechanisms have been proposed. Our framework shares some similarities with prior work carried out in other projects. In this section we compare our framework with these works.

The first and most famous mechanism in misbehaviour detection in MANET is The Watchdog [Marti'00]. We have compared our SMDF results with the Watchdog results on five of our six different metrics using simulation, and found that SMDF outperform the Watchdog in four of these metrics. The five metrics are True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. Where as the Watchdog has slightly lower overhead than SMDF, but only when there is no misbehaviour in the network. Moreover, all of the other Watchdog drawbacks including partial dropping do not exist in our SMDF.

There are many other detection mechanisms especially the reputation mechanisms such as [Buchegger'02b, He'04, Michiardi'02a, Miranda'03, Yang'02] using the watchdog as their main monitoring component. Consequently, they inherited all the drawbacks that the watchdog suffers, even though their other system components are

efficient. This gives our SMDF clear advantages over all of the mechanisms that adopting the watchdog concept in their detection system.

We have also compared our framework with the other types of mechanism that do not use the watchdog as their monitoring component. The most recent solution of these is the Random Two-Hop ACK [Djenouri'05]. Our comparison through simulation showed us that our SMDF outperforms the Random Two-Hop ACK in four of our six simulation metrics. These matrices are Communication Overhead, False Positive Detection Rate, Power Consumption Rate and Scalability. Although, we have similar True Positive Detection Rate as the Random Two-Hop ACK. However, through our simulation comparison we noticed that the Random Two-Hop ACK failed to detect partial dropping in many occasions, and that it can often detect the full dropping case more than the partial one.

Our framework evaluation and comparison with other existing mechanisms shows it performs better and has novel aspects that do not exist in other mechanisms. It also shows that SMDF can be used as suitable and efficient security mechanism for ad hoc network in general and its special kind wireless sensors network.

## 7.4 Future Work

So far in this chapter we have reiterated the project aims, findings and main results and considered the novel contributions of our work. While the contributions of this research are valuable it raises, as research should, some interesting questions. This section deals with, in our view, the more significant of these questions.

**1. Dealing with Control Packets Dropper**

The Sessions-based Misbehaviour Detection Framework (SMDF) we proposed in this work assumes that nodes only drop data packets and not control packets. At the network layer, the routing process can be interrupted through routing control packet modification, selective dropping, table overflow, or poisoning. As such, detecting nodes that drop control packet is highly desirable and challenging. The increase of control packets in MANET as well as wireless sensor networks lead to a high

communication overhead. However, these control packets are important in terms of routing and detecting misbehaviour. For example, if a misbehaving node dropped the control packets that include its neighbouring nodes number of data packets they have received and number of packets they have sent, then will result in wrongly accusing them of misbehaviour. Therefore, it is important to protect control packets and to detect nodes that drop them. To achieve this, a new component has to be added to the detection framework to deal with the control packets separately. The requirements of data packets detection are not appropriate for control packets in that, there are few packets of such kind compared to the first one. Further, dropping control packets should not be tolerated, as it completely excludes selfish nodes from routes. Therefore, any detection mechanism for control packets should be more severe in its judgment regarding this kind of packets. Strategy of dropping up to the tolerable threshold can be used, however it need more investigation since it can not know whether and how much the detection mechanise will notice false observations because of channel conditions.

**2. Detection in Hybrid ad hoc network**

A hybrid ad hoc network is a structure-based network that is a mixture of mobile nodes and fixed nodes that extended using multi-hop communications. Indeed, in this kind of network, the existence of a communication link between the mobile station and the base station is not required. A mobile station that has no direct connection with a base station can use other mobile stations as relays. For large scale sensor networks that may have thousands of nodes in the future, it is more realistic to have a sensor network involves a hybrid of resource-rich specialized nodes in conjunction with small sensor devices [Kumar, R.'03]. Compared with conventional (single-hop) structure-based networks, this new generation can lead to a better use of the available spectrum and to a reduction of infrastructure costs. The coverage of the network is increased while the number of fixed antennas is kept relatively small. Furthermore, the energy consumption of the nodes can be reduced because the signal has to cover a smaller distance. And finally, as the radiated energy is reduced, the interference with other nodes diminishes as well. However, a systematic denial of the packet forwarding service from the mobile nodes would remove all the benefits introduced

by the multi-hop aspect of the communications. It is a challenging task to deal with the problem of packet forwarding denial in such mixture environment. A hybrid framework can be designed to solve the node misbehaviour in such a network. This can be done through a framework, which has separate component for fixed nodes and separate component for the mobile nodes, for the misbehaviour detection. A More efficient suggestion would be to have these components integrated, and can function in both situations i.e. (fixed or mobile) as required. This will save cost and energy.

**3. Detection for Mobile Wireless Sensor Networks**

Even though we assumed in this research that the MANET is stationary and the wireless sensor network is static, it is possible to have mobile sensor nodes. In mobile sensor networks, nodes move freely to get necessary information about a certain event that moves in the nature, such as toxic gas cloud or a radioactive mobile object. In mobile wireless sensor networks applications, the user is also interest in high-level description information about the tracked event. It would be interesting to see how SMDF can be enhanced to support mobile scenarios in different types of both mobile wireless sensor network and mobile ad hoc network. This can be done by adding an Intrusion Detection Component to SMDF, specifically to deal with such mobility problem.

**4. Experimental Test-Bed Evaluation**

In this research we have measured the performance evaluation of our framework SMDF through simulations. Performance evaluation through simulations helpful but will not reflect the reality 100%. It will thus be fascinating to see the actual performance of our complete SMDF framework by integrating every component that it consisted of. Once the above tasks have been completed successfully, it would be interesting to implement the complete model in an experimental test-bed to see its practical feasibility. This task appears feasible in the near future as the price of advanced sensors and handheld devices already decreasing gradually. Most current research in Ad hoc networks and wireless sensor networks utilizes simulation and so there would be great benefit from quality test bed facilities, especially for security researchers.

## 7.5 Concluding Remarks

Wireless networks have become increasingly popular in the past few decades when they are being adapted to enable mobility and wireless devices became popular. It has brought fundamental changes to data networking and telecommunications. One popular type of wireless networks is a mobile ad hoc network (MANET). It is a collections of mobile nodes connected together over a wireless medium. Nodes are computing and communication devices that can be laptop computers, PDAs, mobile phones or sensors.

In MANET, security is one of the most important concerns because a MANET system is much more vulnerable to attacks such as data packet dropping and black-hole than a wired or infrastructure-based wireless network. Designing an effective security protocol for MANET is a very challenging task. This is mainly due to the unique characteristics of MANET, namely shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among users, limited availability of resources, and physical vulnerability.

Existing solutions related to Misbehaviour Detection in MANET were shown to fail to meet all of our requirements, though many are excellent solutions in terms of their intended goals. Four major problems emerged from the literature, first, many detection solutions used one particular monitoring approach in their framework namely the Watchdog which suffers from many drawbacks including its failure to detect misbehaving nodes in cases of collisions, partial collusion, and power control employment power, and as a result their systems inherited same drawbacks even though their isolation component is effective. The other major problem is the high cost in terms of the huge amount of communication overhead that most of the existing solution produced. Finally, they suffer from low scalability and high energy consumption.

The main direction of our work has been to look for an effective approach that can satisfy our initial requirements. The result is a new Low Cost framework entitled Sessions-based Misbehaviour Detection Framework. It consists of three components,

the detection component, the decision component and the isolation component. The detection component contains our novel Sessions-based Misbehaviour Detection Protocol SMDP to detect selfish or malicious nodes that drop packets partially or completely to launch either black-hole or data dropping attacks. For the decision component we have enhanced an existing Bayesian approach to decide whether the node deliberately misbehaved or not. For the Isolation component, we have modified an existing approach and used an Observation-Based Protocol to isolate misbehaving nodes. It uses neighbouring observation experience to isolate misbehaved nodes.

We analysed and evaluated the proposed schemes by simulation techniques. Our evaluation was focused on six important parameters, namely Throughput, Overheard, True Positive Detection Rate, False Positive Detection Rate, Power Consumption Rate and Scalability. By comparing our results to those of other mechanisms available on literature, we showed that our solution has low cost in terms of communication overhead than other approaches. We showed also that our framework has the lowest False Positive Detection Rate amongst other approaches, and that it has highest value of True Positive Detection Rate compared with other approaches. Our evaluation also showed that our solution has lower energy consumption rate compared with other existing approaches. The experiments showed also that our framework is scalable and can work with higher number of nodes, especially in wireless sensor networks. It is important to emphasise that though the proposed framework was developed for Stationary MANET and static wireless sensors network, the ideas by this framework are still applicable for other mobile wireless networks.

To achieve the grand vision of pervasive computing where applications are enhanced through tools such as wireless sensors and integrated using mobile ad hoc networks still requires many problems to be solved. However, remarkable progress has been made in the last decade and we believe our SMDF contribution, addressing fairness within MANET, will help make a step toward this future.

# APPENDIX

## PUBLICATIONS:

T. Fahad, D. Djenouri and R. Askwith. "*On Detecting Selfish Packet Droppers in MANET: A Novel Low Cost Approach*", in *the Third International Symposium on Information Assurance and Security,* Manchester, UK, 2007.

T. Fahad, D. Djenouri and R. Askwith, and M. Merabti. "*A new low cost Sessions-based Misbehaviour Detection Protocol (SMDP) for MANET*", in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07),* Niagara Falls, Canada, 2007.

T. Fahad, R. Askwith, M. Howarth, and G. Pavlou. "*Detecting Selfish Nodes in Wireless Mobile Ad-Hoc Networks*", in *1st Conference on Advances in Computer Security and Forensics,* Liverpool, UK, 2006.

T. Fahad and R. Askwith. "*A Node Misbehaviour Detection Mechanism for Mobile Ad-hoc Networks*", in *The Seventh Annual Postgraduate Network Symposium (PGNet 2006),* Liverpool, UK, 2006.

T. Fahad, S. Yousef, and C. Strange. "*A Study of the Behaviour of the Mobile Agent in the Network Management Systems*", in *The Annual Postgraduate Networking Conference (PGNet 2003),* Liverpool, UK, 2003.

T. Fahad, S. Yousef, and C. Strange. "*The Effect of Mobile Agents in Managing Network Systems*", in *The Third International Conference on Mobile Communication Technologies (3G2003),* London UK, 2003.

# REFERENCES

[Ahmed'05] N. Ahmed, K. S, and J. S, "*The Holes Problem in Wireless Sensor Networks: A Survey*", ACM Sigmobile Mobile Computing and Communications Review, 2005, **9**(2): p. 4–16.

[Akyildiz'02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "*Wireless Sensor Networks: A Survey*", Computer Networks: The Int'l. J. Comp. and Telecommun, 2002, **38**(4): p. 393–422.

[Akyildiz'05] I.F. Akyildiz and X. Wang, "*A survey on wireless mesh networks*", Communications Magazine, IEEE, 2005, **43**(9): p. S23- S30.

[Ang'04] E.Z. Ang, "*Node Misbehaviour in Mobile Ad Hoc Networks*", 2004, National University of Singapore, Technical Report.

[Anjum'03] F. Anjum, D. Subhadrabandhu, and S. Sarkar. "*Signature based Intrusion Detection for Wireless Ad-hoc Networks: A Comparative Study of Various Routing Protocols*", in *Vehic. Tech. Conf., Wireless Security Symp,* Orlando, Florida, USA, 2003.

[Ashwini'05] K. Ashwini, Pandey, and H. Fujinoki, "*Study of MANET routing protocols by GloMoSim simulator*", International Journal Of Network Management, 2005, **15**(6): p. 393–410.

[Asokan'00] N. Asokan and Rinzboorg, "*Key Agreement in Ad Hoc Networks*", Comp. Commun, 2000, **23**(17): p. 1627–1637.

[Awerbuch'02] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. "*An On Demand Secure Routing Protocol Resilient to Byzantine Failures*", in *ACM Wksp. Wireless Security (WiSe),* Atlanta, Georgia, 2002.

[Badonnel'05a] R. Badonnel, R. State, and O. Festor, "*Management of Mobile ad hoc network: information model and probe-based architecture*", International Journal of Network Management, 2005, **15**(5): p. 335-347.

[Badonnel'05b] R. Badonnel, R. State, and O. Festor. "*Management of Mobile Ad-hoc Networks: Evaluating the Network Behavior*", in *IEEE Integrated Network Management,* Nice, France, 2005.

[Bajaj'99] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia, and M. Gerla, "*Glomosim: A scalable network simulation environment*", 1999, UCLA Computer Science Department: Technical Report 990027.

[Bansal'03] S. Bansal and M. Baker, "*Observation-based cooperation enforcement in ad hoc networks*", 2003, Stanford University: Technical Report, URL: http://citeseer.ist.psu.edu/bansal03observationbased.html.

[Basile'03] C. Basile, M.-O. Killijian, and D. Powell, "*A survey of dependability issues in mobile wireless networks*", 2003, National Center for Scientific Research. Technical report: Toulouse, France.

[Becker'98] K. Becker and U. Wille. "*Communication Complexity of Group Key Distribution*", in *5th ACM Conf. Comp. and Commun. Security,*1998.

[Bellovin'92] S.M. Bellovin and M. Merritt. "*Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks*", in *IEEE Symp. Security and Privacy,*1992.

[Broch'04] J. Broch, D.B. Johnson, and D.A. Maltz, "*The dynamic source routing protocol for mobile ad hoc networks. Internet draft*", 2004, IETF, URL: http://www1.ietf.org/mail-archive/web/ietf-announce/current/msg02559.html.

[Bruno'05] R. Bruno, M. Conti, and E. Gregori, "*Mesh networks: commodity multihop ad hoc networks*", IEEE Communications Magazine, 2005, **43**(3): p. 123-131.

[Buchegger'05] S. Buchegger and J.-Y.L. Boudec, "*Self-Policing Mobile Ad Hoc Networks by Reputation Systems*", IEEE Communications Magazine, 2005, **43**(7): p. 101-107.

[Buchegger'03] S. Buchegger and J.Y.L. Boudec, "*A Robust Reputation System for Mobile Ad-hoc Networks*", EPFL IC, Tech. Rep. IC/2003/50, 2003.

[Buchegger'04] S. Buchegger and J.Y. Le-Boudec. "*A Robust Reputation System for p2p and Mobile Ad-hoc Networks*", in *2nd Wksp. Economics of Peer-to-Peer Systems,*2004.

[Buchegger'02a] S. Buchegger and J. Le Boudec. "*Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks.*" in *IEEE Tenth Euromicro Workshop on Parallel, Distributed and Network based,* Canary Islands, Spain, 2002a.

[Buchegger'02b] S. Buchegger and J. Le Boudec. "*Performance analysis of the CONFIDANT protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks*", in *IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobi-HOC),* Switzerland, 2002b.

[Burgess'04] M. Burgess, "*Analytical Network and System Administration. Managing Human-Computer Networks*". 1st ed. 2004: John Wiley & Sons.

[Buttyan'01] L. Buttyan and J.-P. Hubaux, "*Nuglets: A Virtual Currency to Stimulate Cooperation in Self-organized Mobile Ad Hoc Networks*", 2001, Swiss Federal Institution of Technology: Lausanne, Switzerland, Technical Report DSC/2001/001.

[Buttyan'03] L. Buttyan and J.P. Hubaux, "*Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks*", ACM/Kluwer Mobile Networks and Applications (MONET), 2003, **8**(5): p. 579–592.

[Capkun'03] S. Capkun, L. Buttyan, and J.-P. Hubaux, "*Self-organized Public Key Management for Mobile Ad Hoc Networks*", IEEE Trans. Mobile Comp, 2003, **2**(1): p. 52-64.

[Castelluccia'02] C. Castelluccia and G. Montenegro, "*Protecting AODV Against Impersonation Attacks*", ACM SIGMOBILE Mobile Comp. and Commun. Rev, 2002, **6**(3): p. 108–109.

[Chan'03] H. Chan, A. Perrig, and D. Song. "*Random Key Predistribution Schemes for Sensor Networks*", in *IEEE Symp. Research in Security and Privacy*,2003.

[Chen'06] X. Chen, H. Zhai, J. Wang, and Y. Fang, "*A Survey on Improving TCP Performance over Wireless Networks* ", in *Resource Management in Wireless Networking*. 2006, Springer US. p. 657-695. Available on: http:// citeseer.ist.psu.edu/chen05survey.html.

[Chiang'03] T.-C. Chiang and Y.-M. Huang. "*Group Keys and the Multicast Security in Ad Hoc Networks*", in *1st Int'l. Wksp. Wireless Security and Privacy (WiSPr'03)*,2003.

[Conti'04] M. Conti, G. Maselli, G. Turi, and S. Giordano, "*Cross-Layering in Mobile Ad Hoc Network Design*", Computer, 2004, **37**(2): p. 48-51.

[Conti'05] M. Conti, E. Gregori, and G. Maselli. "*Improving the performability of data transfer in mobile ad hoc networks*", in *the Second IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'05)*, Santa Clara, CA, USA, 2005.

[Das'03] S. Das, B. Manoj, and C. Murthy. "*A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks*", in *Mobile Ad Hoc Networks Wksp. (MADNET)*, Sophia-Antipolis, France, 2003.

[Davison'00] A. Davison, "*Bayesian Models, Chapter 11 in Manuscript.*" 2000: Springer.

[Davison'03] A. Davison, "*Statistical Models*". Cambridge Series in Statistical and Probabilistic Mathematics. 2003: Cambridge University Press.

[Deng'02] H. Deng, W. Li, and D.P. Agrawal, "*Routing Security in Wireless Ad Hoc Networks*", in *IEEE Communications Magazine*2002. p. 70-75.

[Deng'03a] J. Deng, R. Han, and S. Mishra. "*Insens: Intrusion-tolerant Routing in Wireless Sensor Networks*", in *23rd IEEE Int'l. Conf. Distributed Comp. Sys. (ICDCS 2003)*,2003.

[Deng'03b] J. Deng, R. Han, and S. Mishra. "*A Performance Evaluation of Intrusion-tolerant Routing in Wireless Sensor Networks*", in *The 2nd International Conference on Information Processing in Sensor Networks IPSN*, 2003.

[Deng'03c] J. Deng, R. Han, and S. Mishra. "*Security Support for In-network Processing in Wireless Sensor Networks*", in *ACM Wksp. Security of Ad Hoc and Sensor Networks (SASN '03)*,2003.

[Diffie'76] W. Diffie and M.E. Hellman, "*New Directions in Cryptography*", IEEE Trans. Info Theory, 1976, **IT-22**(6): p. 644–654.

[Djenouri'05a] D. Djenouri, L. Khelladi, and A.N. Badache, "*A survey of security issues in mobile ad hoc and sensor networks*", Communications Surveys & Tutorials, IEEE, 2005, 7(4): p. 2- 28.

[Djenouri'05] D. Djenouri, N. Ouali, A. Mahmoudi, and N. Badache. "*Random feedbacks for selfish nodes detection in mobile ad hoc networks*", in *The 5th IEEE InternationalWorkshop on IP Operations and Management, IPOM'05, ser. LNCS, no. 3751,* Barcelona, Spain, 2005.

[Djenouri'06a] D. Djenouri and N. Badache, "*New Power-aware Routing for Mobile Ad Hoc Networks*", International Journal of Ad Hoc and Ubiquitous Computing (Inderscience), 2006, **1**(3): p. 126-136.

[Djenouri'06b] D. Djenouri and N. Badache. "*Cross-layer Approach to Detect Data Packet Droppers in Mobile Ad-hoc Networks*", in *the first International Workshop On Self-organized systems IWSOS'06,* Passau, Germany, 2006.

[Djenouri'07] D. Djenouri and N. Badache, "*Struggling Against Selfshness and Black Hole Attacks in MANET*", the Journal of Wireless Communications and Mobile Computing (WCMC), 2007.

[Doshi'02a] S. Doshi and T. Brown. "*Design considerations for an on-demand minimum energy routing protocol for a wireless ad hoc network*", in *In International Conference on Communications (ICC02),* New York, US, 2002.

[Doshi'02b] S. Doshi and T. Brown. "*Minimum energy routing schemes for a wireless ad hoc network*", in *The 22th IEEE Annual Joint Conference on Computer Communications and Networking INFO-COM'02,* San Francisco, California, USA, 2002.

[Du'03] W. Du, J. Deng, Y.S. Han, and P.K. Varshney. "*A Witness-based Approach for Data Fusion Assurance in Wireless Sensor Networks*", in *GLOBECOM '03,*2003.

[Eschenauer'02] L. Eschenauer and V.D. Gligor. "*A Key Management Scheme for Distributed Sensor Networks*", in *CC502,* Washington DC, USA, 2002.

[Evans'00] M. Evans, N. Hastings, and B. Peacock, "*Beta Distribution Chapter*", in *Statistical Distributions*. p.34-42, 2000, Wiley: New York.

[Fahad'06a] T. Fahad and R. Askwith. "*A Node Misbehaviour Detection Mechanism for Mobile Ad-hoc Networks*", in *The Seventh Annual Postgraduate Network Symposium (PGNet 2006)*, Liverpool, UK, 2006.

[Fahad'06b] T. Fahad, R. Askwith, M. Howarth, and G. Pavlou. "*Detecting Selfish Nodes in Wireless Mobile Ad-Hoc Networks*", in *1st Conference on Advances in Computer Security and Forensics Preliminary Programme*, Liverpool, UK, 2006.

[Fahad'07a] T. Fahad, R. Askwith, and D. Djenouri. "*On Detecting Selfish Packet Droppers in MANET: A Novel Low Cost Approach*", in *The Third International Symposium on Information Assurance and Security*, Manchester, UK, 2007.

[Fahad'07b] T. Fahad, R. Askwith, D. Djenouri, and M. Merabti. "*A new low cost Sessions-based Misbehaviour Detection Protocol (SMDP) for MANET*", in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, Niagara Falls, Canada, 2007.

[Ferré'05] S. Ferré and R.D. King, "*A Dichotomic Search Algorithm for Mining and Learning in Domain-Specific Logics*", Fundamenta Informaticae, Advances in Mining Graphs, Trees and Sequences, 2005, **66**(1-2): p. 1-32.

[Ganesan'03] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu. "*Analyzing and Modeling Encryption Overhead for Sensor Network Nodes*", in *WSNA '03: the 2nd ACM Int'l. Conf. Wireless Sensor Networks and Applications*, New York, N1 USA, 2003.

[Gast'02] M.S. Gast, "*802.11 Wireless Networks*". 1st ed. 2002: O'Reilly and Association, mc.

[Gaubatz'04] G. Gaubatz, J.-P. Kaps, and B. Sunar. "*Public Key Cryptography in Sensor Networks Revisited*", in *1st European Wksp. Security in Ad-Hoc and Sensor Networks (ESAS 2004)*,2004.

[GloMoSim'07] GloMoSim website,http://pcl.cs.ucla.edu/projects/glomosim, 2007.

[He'04] Q. He, D. Wu, and P. Khosla. "*SORI: A Secure and Objective Reputation-Based Incentive Scheme for Ad Hoc Networks*", *in IEEE Wireless Communications and Networking Conference WCNC 2004,* Atlanta, GA, 2004.

[Heady'90] R. Heady, G. Luger, A. Macabe, and M. Servilla, "*The Architecture of a Network Level Intrusion Detection System*", 1990, Tech. Rep, CS90-20, Computer Science Department, University of New Mexico.

[Hofflein'98] J. Hofflein, J. Pipher, and J. Silverman, "*Ntru: A Ring-based Public Key Cryptosystem*", Algorithmic Number Theory (ANTS III), 1998, **1423**(LNCS): p. 267-288.

[Hsieh'02] H. Hsieh and R. Sivakumar, "*Transport Over Wireless Networks*", Handbook of Wireless Networks and Mobile Computing, Edited by Ivan Stojmenovic. John Wiley and Sons, Inc, 2002.

[Hu'02a] Y.C. Hu, D.B. Johnson, and A. Perrig. "*Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks*", in *4th IEEE Wksp. Mobile Computing Systems and Applications WMCSA '02*,2002a.

[Hu'02b] Y.C. Hu, A. Perrig, and D.B. Johnson. "*Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks*", in *8th Annual Int'l. Conf. Mobile Comp. and Net. MobiCom 2002*,2002b.

[Hu'03] Y.C. Hu, A. Perrig, and D.B. Johnson. "*Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols*", in *ACM Wksp. Wireless Security WiSe 2003,* San Diego, CA, USA, 2003.

[Hu'04] Y.C. Hu and A. Perrig, "*A Survey of Secure Wireless Ad Hoc Routing*", IEEE Security and Privacy, 2004, **2**(3): p. 28-39.

[Huang'03] Y. Huang, W. Fan, W. Lee, and Y. P.S. "*Cross-feature Analysis for Detecting Ad-hoc Routing Anomalies*", in *23rd Int'l. Conf. Distributed Comp. Sys,* Providence, RI, 2003.

[Hubaux'01] J.-P. Hubaux, L. Bunyan, and S. Capkun. "*The Quest for Security in Mobile Ad Hoc Networks*", in *MobiHoc '01: Proc. 2nd ACM Int'l. Symp. Mobile Ad Hoc Net. and Comp*,2001.

[IEEE'99] IEEE, "*802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification*", 1999.

[IETF'07] The IETF Web site, http://www.ietf.org, 2007.

[Ilgun'95] K. Ilgun, R.A. Kemmerer, and P.A. Porras, "*State Transition Analysis: A Rule-based Intrusion Detection Approach*", IEEE Trans. Software Eng, 1995, **21**(3): p. 181-199.

[Jetcheva'01] J. Jetcheva and D. Johnson. "*Adaptive Demand-driven Multicast Routing in Multi-hop Wireless Ad Hoc Networks", in the ACM Symposium on Mobile Ad Hoc Networking & Computing MOBIHOC 2001.*

[Johnson'96] D.B. Johnson and D.A. Maltz, "*Dynamic source routing in ad hoc wireless networks* ", Mobile Computing, Chapter 5., 1996: p. 153-181.

[Josang'02] A. Josang and R. Ismail. "*The beta reputation system*", in *the 15th Bled Electronic Commerce Conference,* Bled, Slovenia, 2002.

[Jung'05] E.-S. Jung and N.H. Vaidya, "*Power aware routing using power control in ad hoc networks*", ACM SIGMOBILE Mobile Computing Communication Review (MC2R), 2005, **9**(3): p. 7-18.

[Jurdak'04] R. Jurdak, C.V. Lopes, and P. Baldi, "*A Survey, Classification and Comparative Analysis of Medium Access Control Protocols for Ad Hoc Networks*", IEEE Commun. Surveys and Tutorials, 2004, **6**(1).

[Kachirski'03] O. Kachirski and R. Guha. "*Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks*", in *36th Annual Hawaii Int'l. Conf. System Sciences (HICSS'03),* Big Island, Hawaii, 2003.

[Kargl'04] F. Kargl, A. Klenk, S. Schlott, and M. Weber. "*Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks*", in *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004),* Heidelberg, Germany, 2004.

[Karim'06] A.H.M.R. Karim, R.M.A.P. Rajatheva, and K.M. Ahmed. "*An efficient collaborative intrusion detection system for MANET using Bayesian Approach*", in *the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems,* Terromolinos, Spain 2006.

[Kaya'03] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz. "*Secure Multicast Groups on Ad Hoc Networks*", in *1st ACM Wksp. Security of Ad hoc and Sensor Networks SASN'03,* Fairfax, VA, USA, 2003.

[Ko'98] Y.B. Ko and N. Vaidya. "*Location-aided routing, LAR, in mobile ad hoc networks*", in the *Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking ACM/IEEE MOBICOM98,* Dallas, Texas, 1998.

[Krunz'04] M. Krunz, A. Muqattash, and S.J. Lee, "*Transmission power control in wireless ad hoc networks: Challenges, solutions, and open issues*", in *IEEE Network Magazine,* 2004, p. 8-14, **18**(5).

[Kumar'03] R. Kumar, V. Tsiatsis, and M.B. Srivastava. "*Computation hierarchy for in-network processing*", in *the 2nd ACM international conference on Wireless sensor networks and applications,*2003.

[Kumar'95] S. Kumar and E.H. Spafford. "*A Software Architecture to Support Misuse Intrusion Detection*", in *18th National Info. Security Conf,*1995.

[Kyasanur'03] P. Kyasanur and N.H. Vaidya. "*Detection and Handling of MAC Layer Misbehavior in Wireless Networks*", in *IEEE Int'l. Conf. Dependable Systems and Networks (DSN'03),* San Francisco, California, 2003.

[Lazos'03a] L. Lazos and R. Poovendran. "*Energy-aware Secure Multicast Communication in Ad-hoc Networks Using Geographic Location Information*", in *IEEE Int'l. Conf. Acoustics Speech and Sig. Proc,* Hong Kong, China, 2003a.

[Lazos'03b] L. Lazos, R. Poovendran, and G.H. Cirincione, "*Locationaware Secure Wireless Multicast in Ad-hoc Networks Under Heterogeneous Path-loss*", 2003b, University of Washington, Electrical Engineering Department, Tech. Rep. UWEETR-2003-0012.

[Lee'03] Y.H.W. Lee. "*A Cooperative Intrusion Detection System for Ad Hoc Networks*", in *1st ACM Wksp. Security of Ad Hoc and Sensor Networks,* Fairfax, Virginia, USA, 2003.

[Li'02] X.-Y. Li, Y. Wang, and O. Frieder. "*Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Networks*", in *IEEE Int'l. Conf. Comp. Commun. and Networks ICCCN'02,* Miami, FL, USA, 2002.

[Liu'03] D. Liu, P. Ning, and K. Sun. "*Efficient Self-healing Group Key Distribution with Revocation Capability*", in *10th ACM Conf. Comp. Commun. Security CCS,* Washington DC, USA, 2003.

[Liu'06] Y. Liu, Y. Li, and H. Man. "*A Distributed Cross-Layer Intrusion Detection System for Ad Hoc Networks*", in *Annales Des Telecommunications,*2006.

[Lunt'92] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, P. Neumann, and C. Jalali, "*A Real-time Intrusion Detection Expert System (ides)*", 1992, Computer Science Laboratory, SRI International, Menlo Park, California, Technical. Report. URL: http://www.csl.sri.com/papers/9sri/9sri.pdf.

[Madden'02] S.R. Madden, M.J. Franklin, J. Hellerstein, and W. Hong. "*Tag: A Tiny Aggregation Service for Ad-hoc Sensor Networks*", in *OSDI,*2002.

[Maki'00] S. Maki, T. Aura, and M. Hietalahti. "*Robust Membership Management for Ad-hoc Groups*", in *5th Nordic Wksp. Secure IT Systems (NORDSEC'2000),*2000.

[Marti'00] S. Marti, T.J. Giuli, K. Lai, and M. Baker. "*Mitigating routing misbehavior in mobile ad hoc networks*", in *Mobile Computing and Networking (MOBICOM),* Boston, Massachusetts, USA, 2000.

[Merkle'80] R.C. Merkle. "*Protocols for Public Key Cryptosystems",* in *IEEE Symp. Research in Security and Privacy,*1980.

[Michiardi'02a] P. Michiardi and R. Molva. "*CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks ",* in *Communication and Multimedia Security 2002 (CMS'2002),* Portoroz, Slovenia 2002a.

[Michiardi'02b] P. Michiardi and R. Molva, "*Preventing denial of service and selfishness in ad hoc networks ",* ACM Mobile Computing and Communications Review, 2002b, 7(1).

[Miranda'03] H. Miranda and L.ı. Rodrigues. "*Friends and Foes: Preventing Selfishness in Open Mobile Ad Hoc Networks",* in *The 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03),*2003.

[Mui'02] A.H.L. Mui and M. Mohtashemi. "*A computational model of trust and reputation",* in *the 35th Hawaii International Conference on System Science (HICSS),*2002.

[Murthy'96] S. Murthy and J.J. Garcia-Luna-Aceves, "*An Efficient Routing Protocol for Wireless Networks",* ACM Mobile Networks and Application J., special issue on Routing in Mobile Commun. Networks, 1996: p. 183–197, 1(2).

[Myerson'91] R.B. Myerson, "*Game Theory: Analysis of Conflict"*, 1991, Harvard University Press: Cambridge, Mass.

[Naor'02] D. Naor, M. Naor, and J. Lotspiech, "*Revocation and Tracing Schemes for Stateless Receivers",* Electronic Colloquium on Computational Complexity (ECCC), 2002, 9(43).

[Nguyen'06] L. Nguyen and U.T. Nguyen. "*Study of Different Types of Attacks on Multicast in Mobile Ad Hoc Networks*", in *the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)* 2006.

[NS2'07] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/, 2007.

[Nuevo'03] J. Nuevo, "*A comprehensible glomosim tutorial*", 2003: Available on: http: //www.cs.virginia.edu/~jx9n/courses/cs656/glomoman.pdf.

[OPNET'07] OPNET website, http://www.opnet.com/, 2007.

[Ostrovsky'91] R. Ostrovsky and M. Yung. "*How to Withstand Mobile Virus Attacks*", in *10th ACM Annual Symp. Principles of Distributed Computing (PODC9I),* Montreal, Quebec, Canada, 1991.

[Ozaki'99] T. Ozaki, J. Kim, and T. Suda. "*Bandwidth Efficient Multicast Routing Protocol for Ad Hoc Networks*", in *IEEE International Conference on Computer Communications and Networks ICCCN,* 1999.

[Papadimitratos'02] P. Papadimitratos and Z. Haas. "*Secure Routing for Mobile Ad Hoc Networks*", in *SCS Commun. Net. and Distributed Systems Modeling and Simulation Conf. CNDS,* San Antonio, Texas, 2002.

[Papadimitratos'03] P. Papadimitratos and Z.J. Haas. "*Secure Link State Routing for Mobile Ad Hoc Networks*", in *IEEE Symposium on Applications and the Internet Workshops,* Orlando, FL, USA, 2003.

[Park'97] V.D. Park and M.S. Corson. "*A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks*", in *IEEE Annual Conference on Computer Communications INFOCOM 97,*1997.

[PARSEC'07] PARSEC website,http://pcl.cs.ucla.edu/projects/PARSEC, 2007.

[Paul'02] K. Paul and D. Westhoff. "*Context Aware Detection of Selfish Nodes in DSR based Ad Hoc Networks*", in *IEEE GLOBECOM 2002,* Taipei, Taiwan, 2002.

[Perkins'03] C. Perkins, E.M. Belding-Royer, and S.R. Das, "*Ad hoc on-demand distance vector (AODV) routing. RFC 3561*", 2003, IETF.

[Perkins'94] C. Perkins and R. Bhagwat. "*Highly Dynamic Destinationsequenced Distance-vector Routing (DSDV) for Mobile Computer*", in *ACM SIGCOMM 94 Conf. Commun. Architectures, Protocols and Applications,*1994.

[Perkins'99] C. Perkins and E. Royer. "*Ad Hoc on Demand distance Vector (AODV) Algorithm*", in *2nd IEEE Wksp. Mobile Comp. Systems and Applications (WML'SA'99),*1999.

[Perrig'01a] A. Perrig, D.X. Song, and J.D. Tygar. "*ELK, A New Protocol for Efficient Large-group Key Distribution?*" in *IEEE Symp. Security and Privacy,*2001.

[Perrig'01b] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler. "*Spins: Security Protocols for Sensor Networks*", in *the Annual ACM Int'l. Conf. Mobile Computing and Networks (MobiCom 2001),* Rome, Italy, 2001.

[Pfleeger'02] C.P. Pfleeger and S.L. Pfleeger, "*Security in Computing*". Third Edition ed. 2002: Prentice Hall.

[Pietro'04] R.d. Pietro, L. Mancini, and A. Mci. "*Efficient and Resilient Key Discovery based on Pseudo-random Key Pre-deployment*", in *18th Int'l. Parallel and Distributed Processing Symp,*2004.

[Pietro'03] R.d. Pietro, L.V. Mancini, Y.W. Law, S. Etalle, and P. Havinga. "*A Directed Diffusion-based Secure Multicast Scheme for Wireless Sensor Networks*", in *1st Int'l. Wksp. Wireless Security and Privacy (WiSPr'03)*,2003.

[Przydatek'03] B. Przydatek, D. Song, and A. Perrig. "*SIA: Secure Information Aggregation in Sensor Networks*", in *SenSys03*,2003.

[Rabin'89] M. Rabin, "*Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance*", J. ACM, 1989, **36**(2): p. 335–348.

[Royer'99] E. Royer and C. Toh, "*A review of current routing protocols for ad-hoc mobile wireless networks. Mobile Wireless Networks*", IEEE Personal Communications, 1999, **6**(2): p. 46–55.

[Sanzgiri'02] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, Belding-Royer, and E.M. "*A Secure Routing Protocol for Ad Hoc Networks*", in *10th IEEE Int'l. Conf. Network Protocols (ICNP '02)*,2002.

[Sasson'02] Y. Sasson, D. Cavin, and A. Schiper. "*On the accuracy of manet simulators*", in *ACM Principles of Mobile Computing (POMC 2002)*, Toulouse, France, 2002.

[Shamir'79] A. Shamir, "*How to Share A Secret*", Commun. ACM, 1979, **22**(11): p. 612–613.

[Sinha'99] P. Sinha, S. Sivakumar, and V. Bharghavan. "*MCEDAR: Multicast Core Extraction Distributed Ad Hoc Routing*", in *IEEE WCNC*,1999.

[Srinivasan'03] V. Srinivasan, P. Nuggehalli, C. F.Chiasserini, and R. R.Rao. *"Cooperation in wireless ad hoc networks",* in *The 22th IEEE Annual Joint Conference on Computer Communications and Networking INFOCOM'03,* San Francisco, California, USA, 2003.

[Staddon'02] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean. *"Self-healing key Distribution with Revocation",* in *IEEE Symp. Security and Privacy,* The Claremont Resort Oakland, CA, 2002.

[Stallings'05] W. Stallings, *"Cryptography and Network Security, Principles and Practice"*. 4th Edition ed. 2005: Prentice Hall.

[Steiner'96] M. Steiner, G. Tsudik, and M. Waidner. *"Diffie Hellman Key Distribution Extended to Group Communication",* in *ACM Conf. Comp. and Commun. Security,*1996.

[Steiner'98] M. Steiner, G. Tsudik, and M. Waidner. *"A New Approach to Group Key Agreement",* in *Int'l. Conf. Distributed Computing Systems,*1998.

[Stinson'02] D. Stinson, *"Cryptography: Theory and Practice"*. Second Edition ed. 2002: CRC/C&H.

[Toh'96] C.K. Toh. *"A novel Distributed Routing Protocol to Support Ad Hoc Mobile Computing",* in *IEEE 15th Annual Int'l. Phoenix Conf. Comp. and Commun,*1996.

[Venugopalan'03] R. Venugopalan, P. Ganesan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu. *"Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis",* in *CASES '03: the 2003 Int'l. Conf. Compilers, Architecture and Synthesis for Embedded Systems,* San Jose, California, USA, 2003.

[Wang'06] W. Wang and X.-Y. Li, "*Low-cost routing in selfish and rational wireless ad hoc networks*", IEEE Transaction on Mobile Computing, 2006, **5**(5): p. 596–607.

[Wong'98] C.K. Wong, M.G. Gouda, and S.S. Lam. "*Secure Group Communications Using Key Graphs*", in *ACM SIGCOMM '98 Conf. Applications, Technologies, Architectures, and Protocols for Comp. Commun,* 1998.

[Wood'02] A.D. Wood and J.A. Stankovic, "*Denial of Service in Sensor Networks*", Computer, 2002, **35**(10): p. 54-62.

[Xie'02] G.G. Xie, C.E. Irvine, and T.E. Levin. "*Quantifying Effect of Network Latency and Clock Drift on Time-driven Key Sequencing*", in *ICDCSW '02: Proc. 22nd Int'l. Conf. Distributed Computing Systems,* Washington, DC, USA, 2002.

[Xu'02] N. Xu, "*A Survey of Sensor Network Applications*", 2002, University of Southern California, Technical Report. URL: http://courses.cs.tamu.edu/rabi/cpsc617/resources/sensor%20nw-survey.pdf.

[Yang'04] H. Yang, H.Y. Luo, F. Ye, S.W. Lu, and L. Zhang, "*Security in mobile ad hoc networks: Challenges and solutions.*" IEEE Wireless Communications, 2004, **11**(1): p. 38-47.

[Yang'02] H. Yang, X. Meng, and S. Lu. "*Self-organized Network Layer Security in Mobile Ad Hoc Networks*", in *ACM MOBICOM Wireless Security Wksp (WiSe'02),*2002.

[Yasinsac'02] A. Yasinsac and J. Davis. "*Modeling Protocols for Secure Group Communication in Ad Hoc Networks*", in *10th Int'l. Wksp Security Protocols,* Cambridge, UK, 2002.

[Yau'03] P. Yau and C.J. Mitchell. "*Reputation methods for routing security for mobile ad hoc networks*", in *SympoTIC '03, Joint IST Workshop on Mobile Future and Symposium on Trends in Communications*, Bratislava, Slovakia, 2003.

[Yi'03] S. Yi and R. Kravets. "*Moca : Mobile Certificate Authority for Wireless Ad Hoc Networks*", in *2nd Annual PKI Research Wksp (PKI '03)*, Gaithersburg, MD, 2003.

[Zapata'02] M.G. Zapata and N. Asokan. "*Securing Ad Hoc Routing Protocols*", in *ACM Wksp. Wireless Security (WiSe 2002)*,2002.

[Zeng'98] X. Zeng, R. Bagrodia, and M. Gerla. "*Glomosim: A library for parallel simulation of large-scale wireless networks*", in *Parallel and Distributed Simulation*, Canada, 1998.

[Zhang'03] Y. Zhang, W. Lee, and Y. Huang, "*Intrusion Detection Techniques for Mobile Wireless Networks*", ACM Wireless Networks, 2003, 9(5): p. 545–556.

[Zhong'03] S. Zhong, J. Chen, and Y.R. Yang. "*SPRITE: A simple, cheat-proof, credit-based system for mobile ad-hoc networks*", in *the 22th IEEE Annual Joint Conference on Computer Communications and Networking INFOCOM'03*, San Franciso, CA, USA, 2003.

[Zhou'99] L. Zhou and Z.J. Haas, "*Securing Ad Hoc Networks*", IEEE Network, 1999, **13**(6): p. 24-30.

[Zhu'03] S. Zhu, S. Xu2, S. Setia, and S. Jajodia. "*Establishing Pair-wise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach*", in *11th IEEE Int'l. Conf. Network Protocols*,2003.

[Zhu'04] S. Zhu, S.S.S. Xu, and S. Jajodia. "*Gkmpan: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks*", in *1st Annual Int'l. Conf. Mobile and Ubiquitous Systems: Net. and Services (MobiQuitous'04)*,2004.

[Zimmermann'95] P. Zimmermann, "*The Official PGP Users Guide*", 1995: MIT Press.