Analysing the Performance of MANET Trust-Based Routing Protocols in the Presence of Attacks

Ali Alzahrani

Supervisor: Nigel Thomas
Matthew Forshaw

School of Computing
Newcastle University

This dissertation is submitted for the degree of
Integrated PhD in Computer Science

August 2023

To my dearest mother, wife, and children,

I dedicate this thesis to you with heartfelt gratitude and appreciation for your unwavering support and understanding throughout my five-year journey. Your patience, encouragement, and love have been my pillars of strength, without which this accomplishment would not have been possible.

I also extend my sincerest thanks to my supervisors, Nigel Thomas and Matthew Forshaw, whose expertise, guidance, and encouragement have been invaluable in shaping my research and enabling me to achieve this milestone. Your mentorship and wisdom will always remain with me as I embark on new adventures.

Thank you all for believing in me, cheering me on, and being my constant source of inspiration. This achievement is as much yours as it is mine.

Declaration

I confirm that this dissertation is entirely original and has not been submitted previously for any other academic qualification, either in part or in full, at this university or any other institution. I declare that this dissertation is my own work and does not include any material that has been produced in collaboration with others, except where specific reference is made to their work in the text and Acknowledgements.

Ali Alzahrani
August 2023

Abstract

Mobile ad-hoc networks (MANETs) are wireless multi-hop networks that do not rely on any fixed infrastructure, unlike traditional networks. Nodes in MANETs are formed dynamically and are free to move in any direction at variable speeds. MANETs can be used efficiently in situations in which infrastructure has been destroyed, such as in earthquakes, or where there is no fixed infrastructure, such as in wild areas, or where the existing infrastructure is insufficient, e.g. for a festival or other event.

MANETs do not have a centralised administration and nodes join and leave the network freely without any validation or authentication. These special characteristics make MANETs vulnerable to many types of network attack that can negatively affect their performance. Moreover, because of their nature, it is not possible to employ the solutions designed for traditional networks. One possible solution that could help assure the performance of MANETs in the face of a network attack is to implement trust management. The principle of trust management in MANETs is that each node monitors the behaviour of its neighbouring nodes and tries to detect any malicious activities. Once a node identifies that a neighbouring node is malicious, it will categorise it as untrustworthy and avoid sending data to it in the future.

There is more than one way of designing trust management schemes in MANET. One is a direct trust scheme, in which each node calculates the trust values for each of its neighbouring nodes itself. Another is an indirect trust scheme, in which a node receives recommendations about its neighbouring node from other nodes in the network. This study proposed and implemented four direct trust schemes and evaluated their performance against black-hole, grey-hole, selfish, and flooding attacks. It also proposed and evaluated the performance of an indirect trust scheme against a black-hole attack.

The NS2 and NS3 network simulators were used to run the simulations. The mechanisms were implemented with the most common MANET routing protocol, ad-hoc on-demand distance vector (AODV). Having added the trust management schemes to the AODV protocol, the study compared the performance of the plain AODV and the trust-based AODV in the presence of the attacks.

The simulations showed that implementing a direct trust management mechanism within the AODV protocol improved the performance of MANETs in the presence of

attacks at the cost of a slight increase in the overhead. The indirect trust scheme resulted in greater improvements in performance than the direct trust schemes but with higher overhead.

Publications

During my PhD, I have published the following papers:

1. A. Alzahrani, H. Jari, and N. Thomas, "Analysing the effect of mobility on the performance of MANET routing protocols," *35th Annual UK Performance Engineering Workshop UKPEW*, pp. 1–11, 2019

2. A. Alzahrani, H. Jari, and N. Thomas, "Performance evaluation of MANET trust-based aodv protocol in the presence of blackhole attacks," *36th Annual UK Performance Engineering Workshop UKPEW*, pp. 30–40, 2020

3. A. Alzahrani, H. Jari, and N. Thomas, "A novel indirect trust mechanism for addressing black hole attacks in MANET," *DIVANet '21: In Proceedings of the 11th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 27–34, 2021

4. A. Alzahrani and N. Thomas, "Direct trust management mechanism to detect selfish attacks on MANET," *38th Annual UK Performance Engineering Workshop UKPEW*, pp. 12–19, 2022

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AODV Ad Hoc On-Demand Distance Vector

BM Behavioral Mobility

CI Confidence Interval

DoS Denial of Service

DSDV Destination Sequenced Distance Vector

DSN Destination Sequence Number

DSR Dynamic Source Routing

FFM Fluid Flow Mobility

GCM Geographic Constraint Mobility

GM Gravity Mobility

ITAODV Indirect Trust-based Ad Hoc On-Demand Distance Vector

MANET Mobile Ad-hoc Network

QoS Quality of Service

RERR Route Error Packet

RGM Random Gauss-Markov Mobility

RREP Route Reply Packet

RREQ Route Request Packet

RWP Random Waypoint Mobility

RW Random Walk Mobility

TAODV Trust-based Ad Hoc On-Demand Distance Vector

TTL Time To Live

Chapter 1

Introduction

A mobile ad-hoc network (MANET) is a decentralised, wireless, self-configured network that is formed dynamically by multiple mobile nodes without the use of any centralised administration or existing infrastructure [5, 6]. MANETs do not rely on a pre-existing infrastructure, such as access points, routers, or servers. Instead of using routers to forward data through the network, each node participates in routing, forwarding the data from the source node to the destination node. A route in MANET is created when it is needed, in other words when a node wants to send data to another node in the network [7]. There are many routing protocols for MANETs, each of which has its own mechanism for creating the routes through the network [8]. In MANETs nodes are completely free to move in any direction at varying speed. Also, the nodes are free to join and leave the network at any time without an authentication process or prior notification. As each node in a MANET is independent and free to move in any direction, the network's topology changes frequently [9]. One of the main challenges in establishing a MANET is ensuring that each node has sufficient capacity to work as a router and meet the demand of routing requests made by other nodes. The challenge becomes even harder when the scale of the network increases. Figure 1.1 shows how a MANET can be formed by different nodes with different capabilities.

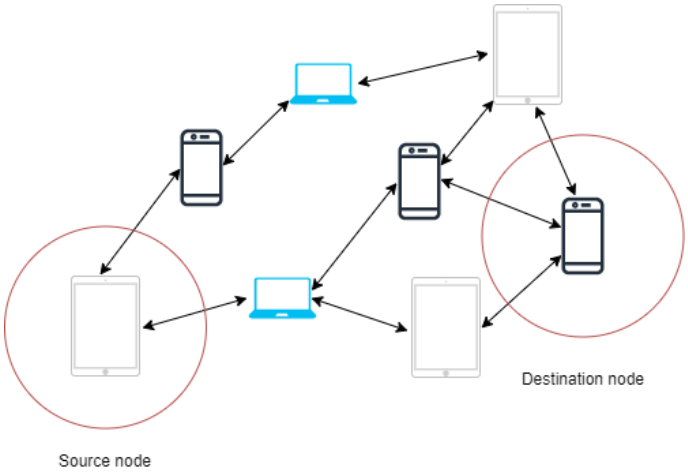

Fig. 1.1 Mobile ad-hoc network (MANET)

MANETs allow for flexible and quick deployment in areas without a fixed network infrastructure. This makes MANETs ideal for several applications, such as military operations, disaster relief efforts, and vehicular networks. However, the lack of a fixed infrastructure also presents major limitations in the network's scalability, stability, and performance. The network architecture, operational requirements, and routing capabilities of MANETs present the following characteristics [10, 11]:

1. Most of the time, packets need to be forwarded by many nodes to reach the destination node. Nodes in MANETs need to be cooperative and have the required capability to work as routers. This cannot always be guaranteed.

2. The topology in MANETs is dynamic because all nodes are free to join and leave the network at any time. This causes rapid route changes and can lead to the lose of packets.

3. The resources of the nodes such as battery and memory, are limited.

4. The processes of forwarding the data packets and determining the routes are distributed across the nodes.

5. The number of nodes and the topology are variable.

6. The links created between the nodes are unreliable and their capacity is variable.

7. MANETs are self-configuring and self-healing.

8. Because of the lack of a centralised administrator and management, each node in the network is autonomous.

These characteristics make MANETs different from traditional networks and so they need different solutions to fix issues and improve performance. MANET routing protocols face greater challenges in providing a decent level of quality of service (QoS) for users [12]. Thus, their characteristics need to be taken into account when designing any solution for MANETs.

The absence of a central administration leads to many vulnerabilities that can be exploited to harm the network performance [13]. For example, uncooperative (malicious) nodes can join the network at any time, as can cooperative nodes, without any authentication or validation. They can use the network to send and receive the data they need, but without helping the other nodes deliver their packets. Even worse, they can launch attacks to shut down the network or reduce its performance.

1.1 Research problem

MANET routing protocols have a major security issue that affects their performance in the presence of network attacks. Due to the special characteristics of MANETs, the routing protocols are vulnerable to many such attacks. A network attack is defined as any attempt to alter, disable, expose, steal, destroy, or gain unauthorised access to any data in the network [14]. The main issue with MANET routing protocols is that they trust all nodes, assuming that they will cooperate in forwarding the data and will not engage in any malicious activity. Thus, they do not provide any defence against attacks launched by malicious nodes when they occur [15].

One of the most popular MANET routing protocols is the ad hoc on-demand distance vector (AODV) protocol. This protocol creates a route between the source node and the destination node when it is needed, but because of the open nature of MANETs, malicious nodes can participate in the route. When a malicious node is involved in establishing a route, it can launch several types of attack against the network [16]. Various attacks can affect the performance of MANET routing protocols, such as black-hole, grey-hole, selfish, and flooding attacks. Networks attacks can lead, for example, to changing the direction of the packets sent or dropping them, which will lead to a reduction in the throughput of the network. Table 1.1 shows how each of the four attacks works and the effect on the performance of MANETs.

Table 1.1 Network attacks and their effect on MANETs

Attack	How it works	Consequences
Black-hole	A middle malicious node sends a fake routing reply packet to the source, saying "Send me your data, I have the most up-to-date route to the destination node".	The malicious node drops the data sent by the source node rather than forwarding them to the next node in the route. This will create a black-hole in the network and might lead to DoS.
Grey-hole	A node operates normally in a cooperative manner, but at some point turns malicious and drops data packets received rather than forwarding them.	The malicious node passes some data packets and drops others rather than forwarding all of them. This will reduce the performance of the network.
Selfish	To save resources, a selfish node does not forward data received from other nodes, but only sends and receives the data packets which are in its own interest.	Data arriving at a selfish node will not be forwarded to the next node in the route, affecting network performance.
Flooding	A malicious node sends a huge number of fake routing requests asking for routes to non-existent destinations.	The network is flooded by fake requests, which can consume its resources and result in DoS.

Note: DoS = denial of service

1.1.1 Why we have chosen these four attacks

This research focuses on these four attacks because of the following reasons:

1. Relevance to MANETs: Black-hole, grey-hole, selfish, and flooding attacks specifically target the unique characteristics of MANETs, such as decentralized control, limited resources, and dynamic topology. Researching these attacks provides insights into vulnerabilities and threats specific to mobile ad hoc networks.

2. Impact: These attacks can cause significant disruptions to the network's functioning, stability, and dependability. Comprehending their effects is essential to develop efficient mechanisms for improving the network performance.

3. Complexity: These attacks represent a range of complexities, from simple black-hole attacks to more sophisticated grey-hole behaviors. By studying attacks with varying levels of complexity, we can develop comprehensive defense strategies.

4. Baseline for Comparison: As these attacks are well-known and widely studied, they serve as a baseline for comparing the effectiveness of new security and performance solutions. We can use them to benchmark the performance of our proposed defense mechanisms.

5. Real-World Threats: In real-world scenarios, these attacks have been known to be employed by malicious entities. Understanding these threats helps us address actual security challenges faced by MANETs.

More examples of attack that can affect MANET performance are the wormhole, Sybil, and jamming attacks. In the wormhole attack, an attacker captures packets at one location in the network and tunnels them to another location, creating a shortcut for communication which can disrupt the routing process. In the Sybil attack, a malicious node impersonates multiple identities in the network, which can disrupt network services, cause resource exhaustion, and manipulate routing algorithms. In the jamming attack, an attacker emits radio signals at the same frequency as the wireless transmissions, disrupting communication between nodes and causing packet loss.

As MANETs are completely different from traditional networks with centralised routers, they need different solutions to improve their performance in the presence of attacks.

1.1.2 The difference between black-hole and grey-hole attacks

Black-hole attack

In a black hole attack, a malicious node advertises itself as having the shortest or most optimal path to a specific destination node. As a result, other legitimate nodes in the network divert their traffic towards this malicious node, believing it to be the most efficient route. However, the malicious node simply drops all the incoming packets instead of forwarding them further. This behavior effectively creates a "black hole" in the network, where data packets sent to that destination disappear and never reach their intended recipients.

The impact of a black hole attack is severe as it completely disrupts the communication between legitimate nodes, leading to data loss and hindering the overall network performance. These attacks are relatively straightforward to execute, and their detection can be challenging due to the decentralized nature of MANETs.

Grey-hole attack

In this type of attack, the malicious node selectively drops certain packets while forwarding others. The decision to drop or forward packets is based on specific criteria set by the attacker, such as the source, destination, or content of the packets.

The "grey" aspect of this attack comes from the fact that the malicious node behaves cooperatively at times, forwarding some packets to maintain the appearance of normal network behavior. However, it also selectively drops packets to disrupt the network's performance and routing efficiency.

The impact of a grey-hole attack is less than the black-hole but can still be significant. By strategically dropping specific packets, the attacker can disrupt critical communication links or target specific data flows, leading to uneven distribution of traffic and potential data loss. Detecting and mitigating grey hole attacks is more challenging compared to black hole attacks due to the deceptive behavior exhibited by the malicious node.

In summary, the main difference between a black-hole and a grey hole attack in MANETs lies in their behavior. A black hole attack drops all incoming packets, while a grey-hole attack selectively drops certain packets while forwarding others. Grey-hole attacks are harder to detect, making them a more sophisticated threat to the performance and security of MANETs.

1.2 Research aims and objectives

1.2.1 Aims

This thesis aims to propose schemes to improve the performacne of MANETs in the presence of attacks. MANETs are not a new technology; indeed, they have existed for more than two decades. However, they are still vulnerable to many types of basic attack [17]. MANETs are valuable as a low-cost technology that can be used in many situations and thus, it is essential to scrutinise their routing protocols and improve their algorithms.

The most harmful event that can happen to a network's performance is the launch of a successful attack. An attack will clearly affect the network's performance unless the malicious node is detected and isolated quickly. Since MANETs are more vulnerable than traditional networks, it is necessary to understand how each individual attack works and affects MANETs. This will make it possible to create a defence mechanism against each attack and later combine them into an approach that should work against many attacks.

Another aim of this thesis is to use the trust management principle to create solutions. These solutions should make MANETs more reliable when they are under attack. Many

mechanisms can be built upon the trust management principle. Since each attack has its unique way of attacking MANETs, each may require a different defence mechanism to work effectively against it and reduce its harm. This thesis seeks to understand in detail how black-hole, grey-hole, selfish and flooding attacks affect the performance of MANETs and develop trust management solutions to reduce their negative effects on the network. Each proposed solution should be able to hack into the attack mechanism, and then detect and isolate the malicious node. Thus, it will be possible to improve network performance in the presence of an attack.

As MANET routing protocols are open source, producing more direct trust management schemes and evaluating them is possible at little cost. When a malicious node launches a successful attack, the network will not necessarily shut down. The network may continue working, but its performance might be affected and this effect could be high or low. Based on this observation, the research questions are as follows:

1. What is the extent of impact that mobility and attacks (black-hole, grey-hole, selfish, and flooding) can have on MANET performance metrics?

2. Do direct trust management techniques have the potential to enhance MANET performance when facing attacks, and if they do, to what extent?

3. How much overhead can be added to the nodes to process the trust management algorithms in normal situations when there is no attack?

4. How do direct and indirect trust management techniques differ in terms of the amount of improvement in MANET performance in the presence of attacks, and the amount of overhead added to the nodes?

1.2.2 Objectives

The objectives of this thesis are as follows:

1. To analyse the performance of MANETs in normal situation when there are no attacks and all nodes are cooperative. This is necessary to identify the parameter that has the greatest effect on performance and can later be considered the main parameter used in evaluating the proposed solutions. (Related to research question 1).

2. To analyse the performance of MANETs in the presence of black-hole, grey-hole, selfish, and flooding attacks. (Related to research question 1).

3. To propose and implement a trust management scheme for each attack and evaluate its performance in the presence of the attack. (Related to research question 2).

4. To analyse the performance of the proposed trust management schemes when there are no attacks to determine the overhead they add to the routing protocol. This is very important to establish the trade-off between the improvement the schemes provide and the overhead they cause. (Related to research question 3).

5. To compare the performance of two types of trust management: direct and indirect. (Related to research question 4).

Each proposed scheme will be added individually to a MANET routing protocol. The protocol's performance will be analysed before and after the scheme is added in the presence of an attack. This will show the level of improvement in network performance after adding the scheme, which can be measured using various metrics, such as the throughput and packet delivery ratio (PDR).

1.3 Research motivation and contribution

1.3.1 Motivations

From the academic perspective, all MANET routing protocols are open source and free to use, which means there is no need for any licence to amend them or modify their algorithms. This is a major advantage for any researcher. The network simulators compatible with MANETs are also open source and free to use. This provides greater freedom in testing, evaluating, and implementing the solutions, as well as obtaining more results. This is a major motivation not only to undertake this research for the PhD but also to continue after that, as it would even be possible to work from home without the need for financial support or expensive equipment.

From the side of the technology itself, enhancing the performance of MANETs routing protocols holds considerable potential benefits. Firstly, it has the capacity to improve the efficiency and reliability of communication within MANETs, enabling the development of novel applications and services. For instance, dependable communication in MANETs has the potential to support remote healthcare, disaster management, and military operations.

Secondly, improving the performance of MANET protocols can help to reduce the impact of network attacks such as the four mentioned above. By designing robust and secure protocols, users can ensure that MANETs can operate seamlessly in hostile environments, particularly in situations where communication and coordination are critical.

Finally, improving the performance of MANET protocols can significantly contribute to the advancement of wireless networking research. Researchers who tackle the unique challenges posed by MANETs can develop new algorithms and schemes that are transfer-

able and applicable to other wireless network types. These advancements can lead to the discovery of innovative solutions that benefit society as a whole.

1.3.2 Contributions

The contributions presented in this thesis are as follows:

1. Analysis of Node Speed Impact: This study introduces a unique analysis of the effect of mobile node speed on MANET performance. It provides novel insights into the relationship between node mobility speed and important performance metrics, specifically throughput and Packet Delivery Ratio (PDR). (Related to research question 1).

2. Analysis of Attack Effects: The study conducts a thorough analysis of the impact of various attacks, including black-hole, grey-hole, selfish, and flooding attacks, on MANET performance. By comprehensively studying the effects of these attacks, the research advances the understanding of MANET security challenges and provides valuable insights into the vulnerabilities and potential disruptions caused by these attacks. (Related to research question 1).

3. Proposing Direct Trust Management Schemes: The thesis proposes four novel direct trust management schemes specifically designed to detect and mitigate the four types of attacks mentioned above. These schemes present approaches to enhance MANET performance by establishing trust relationships among nodes and effectively identifying and isolating malicious behaviors. (Related to research question 2).

4. Overhead Analysis of Trust Management Schemes: The research analyzes the overhead incurred by integrating the proposed direct trust management schemes into the existing protocol. This analysis is crucial as it helps in evaluating the practicality and feasibility of implementing the proposed security mechanisms in MANETs. (Related to research question 3).

5. Novel Indirect Trust Management Mechanism: In addition to direct trust management, the thesis introduces a novel indirect trust management mechanism. This mechanism is specifically tailored to enhance MANET performance under black-hole attack conditions. The comparison of its performance with the direct trust management mechanism in the presence of a black-hole attack highlights the unique benefits and trade-offs of these two approaches. (Related to research question 4).

1.4 Thesis structure

The thesis comprises seven chapters, presented in Figure 1.2.

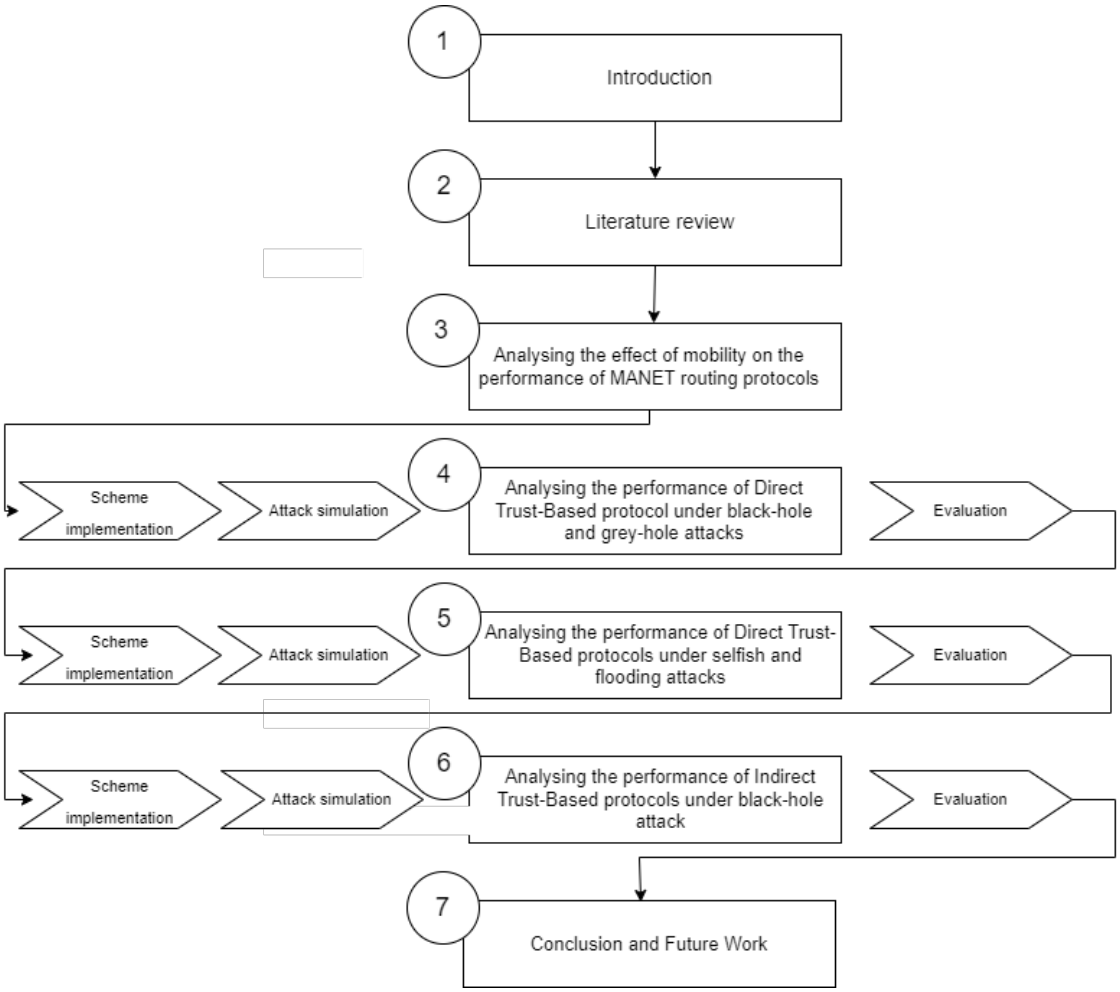

Fig. 1.2 Thesis structure

Chapter 1 describes the motivation for this work, the research problem, objectives, contributions, and the structure of the thesis.

Chapter 2 provides a solid background to MANETs, reviews previous work most related to the research scope, and explains the methodology used to conduct the research.

Chapter 3 analyses the effect of the speed of the nodes' mobility on the performance of various MANET routing protocols, namely ad hoc on-demand distance vector (AODV), dynamic source routing (DSR), and destination sequenced distance vector (DSDV). The outcomes of this chapter have been published in [1]

Chapter 4 analyses the effect of black-hole and grey-hole attacks on the performance of AODV. The chapter also implements and evaluates two direct trust schemes designed to counter the two attacks. The findings of this chapter have been published in [2]

Chapter 5 analyses the effect of selfish and flooding attacks on the performance of AODV. This chapter implements and evaluates two proposed direct trust schemes designed to counter the two attacks. The outcomes of this chapter have been published in [4]

Chapter 6 proposes a novel indirect trust management scheme to detect and isolate black-hole attacks. The contribution of this chapter has been published in [3]

Chapter 7 summarises the work done in the thesis, draws conclusions, illustrates its limitations and propose avenues for future work.

1.5 Summary

This chapter was an introduction for the thesis. It explained what MANETs are and illustrated their unique characteristics. It described the research problem and determined the research questions. The aims, objectives, motivations, structure and the potential contributions of the thesis were explained. In the next chapter we are going to dig in deep and start the literature review where we cover a solid background about the topic, the methodology we are going to use and many more.

Chapter 2

Literature Review

2.1 Introduction

The last chapter explained the research problem and the effect that attacks can have on the performance of MANETs. It mentioned briefly how each attack works and its effect on MANETs, then set out the aims and objectives of the research. The chapter also stated the motivations for undertaking this study and its expected contributions. Finally, it outlined the thesis structure and listed the chapters forming the thesis.

This chapter provides background information about MANETs, how they work and their vulnerabilities in some detail. It includes a broad explanation of the principle of trust management and how it might improve MANET performance. The different types of routing protocols in MANET and how they work are described and explained. The chapter also reviews related work, including many studies that have previously tackled this research problem. It also explains the proposed approaches to address the attacks and improve the performacne and how these differ from those used in the existing literature. Finally, the chapter sets out the methodology and tools used to conduct the research and explains them in some depth.

In this thesis, the parameter latency refers to the time it takes for a packet to be transmitted from the source node to the destination node. Latency is also sometimes referred to as "end-to-end delay". Since MANETs are characterized by their dynamic and self-organizing nature, the latency can be influenced by various factors such as routing protocols, network load, network topology, and node mobility.

2.2 Background

A mobile ad-hoc network MANET is a collection of independent nodes that interact and communicate with each other within their transmission ranges either directly one-to-one or

through intermediate nodes [5]. The nodes in MANET are self-organised, and they are connected using wireless links without any fixed infrastructure. The nodes can be mobile phones, laptops or tablets with different capabilities. The nodes can freely move at any time in random directions at random speeds [18]. Any node may exit the transmission range of its neighbouring nodes at any time, which leads to a break in the route between them. When that happens, a new route must be created between the source node and the destination node. Since there is no central administration in a MANET, administrative control is distributed among the nodes involved in the network. The nodes are expected by default to cooperate in performing specific functions, such as routing, sending necessary requests and replying to requests made by other nodes [5]. Each node in a MANET can function as a source, destination or router. Indeed, the node can do all three functions at the same time. It can communicate with another node by sending and receiving packets and simultaneously participate in a route to forward packets between other nodes.

MANET routing protocols rely on the nodes' cooperation [19]. The routing protocols assume that all participating nodes in the route that links the source node to the destination node are cooperative, but this is not always the case [20, 21]. Some nodes engage in malicious activities, which typically reduce the performance of the network. MANETs have many unique characteristics that make them different from traditional networks. MANET topology changes frequently, so many new nodes will join and many others will leave over each interval [22]. Usually, a node in a MANET communicates with new nodes that it has not dealt with before, so it has no background information on whether they are cooperative or malicious.

MANETs are open to access by any entity with the appropriate resources and equipment and access cannot be restricted or authenticated [19]. A malicious node can easily become engaged in any route. In traditional networks, preventing a malicious node from accessing the network is the first barrier in protecting the performance of the network. This advantage does not exist in MANETs.

Nodes in MANETs are heterogeneous and have different capabilities in terms of energy availability, quality of service requirements, mobility patterns and transmission power [5]. Some nodes may be cooperative, but they do not have the ability to help. For example, when a node has minimal free space in its memory, it may only be able to reply to a limited number of requests made by its neighbouring nodes. This is also the case with the transmission range. If the transmission range of a node is limited to 30 m, for example, it will not be able to create links with nodes at a greater distance from it.

Due to these unique characteristics of MANETs, they are highly vulnerable to network attacks [5]. The first step a hacker would take to attack a network is to connect to it. Attacks launched while connected to the targeted network are the most harmful and successful.

Unfortunately a hacker does not need to make much effort to connect to a MANET. There is no need to provide a password, key, session token, or user account information.

The second step a hacker would take to run a successful attack is to position itself as a node between the source and destination nodes to gain complete control of the data packets sent. This is a popular network hacking strategy called man-in-the-middle. Once a hacker has participated in the route that connects the source node to the destination node, it is possible to run many attacks and harm the network's performance easily and successfully. Unfortunately, this is a straightforward step to take in MANETs. There are no restrictions on any node participating in creating a route to link a source to a destination [19]. Thus, any node can be man-in-the-middle regardless of whether it is cooperative or malicious. Because of this, MANETs are vulnerable to various types of network attack such as black-hole, grey-hole, selfish and flooding attacks, and many more.

The routing mechanism in MANETs has two main phases: the discovery phase and the maintenance phase [5]. The routing phase starts when a node wants to send some data to another node in the network. In this phase, the source node searches for a route to the destination by broadcasting a route request (RREQ) packet asking the neighbouring nodes for a fresh route to the destination. If any neighbouring node has a route to the destination, it will reply to the source by sending a route reply (RREP) packet. Most network attacks exploit this phase because of the lack of any validation or authentication. The maintenance phase is when the route between the source and destination breaks for any reason; the source needs to maintain the transfer and find another route. The node that has lost the connection on the route to the destination node will send a route error (RERR) packet back to the source node to report that the route is no longer valid. When the source node receives the RERR packet, it will stop streaming the data packets and trigger the discovery phase again to find a new route. The diagram in Figure 2.1 shows how MANET routing protocols switch between the discovery phase and the maintenance phase.

The solutions that help improve the performance of traditional networks under such attacks will not work for MANETs [18]. For example, in traditional networks, it is possible to install software on the server to monitor the behaviour of the nodes and detect malicious activities and then block the malicious nodes. In MANETs, this solution is impossible because there is no server or centralised administration. One of the promising ideas for improving MANET performance in the presence of network attacks is a principle called trust management. This is discussed in detail in Section 2.5.

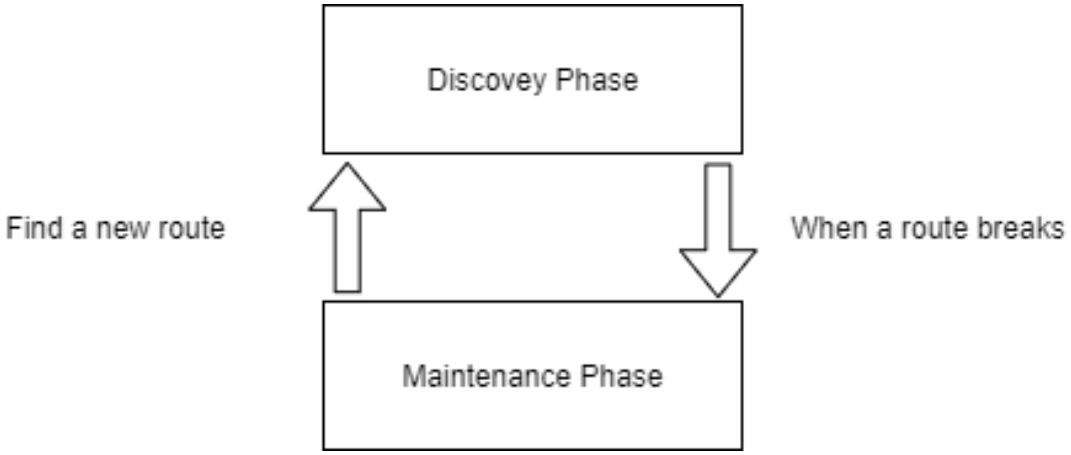

Fig. 2.1 Discovery and maintenance phases in MANET routing protocols

2.3 Mobile ad-hoc network (MANET) routing protocols

Routing protocols define how routers communicate with each other and how to select routes to connect a source node to a destination node in the network [5]. A routing protocol can be defined as a set of rules and procedures used by nodes to determine the best route for forwarding data packets between nodes. When a data packet is sent from one node to another, it passes through a series of nodes that form the path to the destination.

Designing routing protocols for MANETs is much more challenging than for wired networks for several reasons [5]:

1. Dynamic topology: Node mobility in ad-hoc networks causes frequent changes in topology.

2. Variable capacity: The capacity of wireless links is variable and unpredictable.

3. Limited resources: In MANETs, the nodes have limited computing, power and bandwidth resources.

4. Lack of centralised administration: In a traditional network, a central node such as the server acts as a control point for routing decisions. In a MANET, this does not exist and all nodes need to participate in the routing processes. A routing protocol in a MANET needs to be distributed across the network.

5. Security issues: Due to the lack of a fixed infrastructure, MANETs are more vulnerable to network attacks, such as those mentioned before.

Thus, designing routing protocols for MANETs is a difficult task in terms of confidentiality, integrity and availability.

Routing protocols are classified into three different groups: reactive, proactive and hybrid [23]. In the first group, routes to specific destinations are determined when they are required by source nodes using route discovery processes. In the second group, the routes to all destination nodes are determined when the network starts and are maintained using update processes. The last group combines the main properties of the proactive and reactive classes.

2.3.1 Reactive Routing Protocol (On-Demand)

Reactive routing protocols are also known as on-demand protocols. In this type of protocol, routes are created whenever a source needs to send data to a destination node [5]. Reactive routing protocols have two phases: the route discovery phase and the route maintenance phase. In the first phase, route discovery, the source node tries to discover a fresh route to access the destination node. It does this by asking its neighbouring nodes if any of them has a fresh route to the desired destination node. Once the source node receives a reply from a neighbouring node confirming that it has a fresh route to the destination node, it starts sending the data packets through this node, which should forward them to the next node in the route [5].

The second phase, route maintenance, triggers if the route between the source and destination nodes breaks for any reason. In this phase, the source node is informed that the link is broken, so it stops sending the data packets and triggers the discovery phase again to find a new route. This phase is used because the dynamic topology of the network causes the route failure between nodes to increase [11].

One of the advantages of reactive routing protocols is that they can reduce routing overhead because they do not need to search and maintain routes that do not have data traffic [24]. There are various well known reactive routing protocols, such as the ad hoc on-demand distance vector (AODV) protocol and the dynamic source routing (DSR) protocol.

Ad hoc on-demand distance vector (AODV)

AODV is a reactive routing protocol and thus defines the route between the source and the destination when required. Such protocols consist of two mechanisms: route discovery and route maintenance [9, 14].

Route discovery. Route discovery is a route-finding process in which a valid route between the source and the destination needs to be discovered [11]. In the AODV routing discovery process, the steps are as follows:

1. When a source node wishes to send data to a destination node, it checks its routing table. If a valid route exists, data transmission is started; otherwise, it generates an RREQ packet to find a valid route to the destination node. The RREQ packet contains the following information: source IP, destination IP, source sequence number, destination sequence number and broadcast ID number.

2. When an intermediate node receives this packet, it checks its routing table. If it has a valid route to the destination, it sends an RREP packet to the source to report that it has a fresh route to the destination node. The RREP packet contains the following information: destination IP, destination sequence number, hop count, lifetime and source IP. If there is no valid route to the destination, it broadcasts the RREQ packet within its transmission range.

3. Once the source node has received the RREP packet, it starts transferring data using that path.

Route maintenance. If a break in the link occurs:

1. The upstream node notifies the source by sending an RERR packet. This contains the following information: source IP, unreachable destination IP and destination sequence number.

2. All the intermediate nodes mark the route as invalid.

3. Once the source receives the RERR packet, it starts a new route discovery process.

Dynamic source routing (DSR)

DSR is another reactive routing protocol for MANETs. DSR employs a technique called source routing, in which the path sequence is predetermined by the source node [25]. The list of nodes that form the route to the destination is called the route record, which is included in the header of each packet sent by the source node.

Route discovery. The route discovery phase in DSR can be divided into four main steps:

1. When a node wishes to send data to a destination, it first checks its routing table for a valid route. If it has a valid route, data transfer will start. When an intermediate node receives the packet from the source node, it first checks whether it is the intended destination node. If it is, it will process the packet; otherwise, it will forward the packet to the next node according to the route record stored in the packet header. This process happens if the source node already has a route to the destination stored

in its routing table. If the source node does not have a path to the destination, it will broadcast an RREQ packet to the network.

2. When an intermediate node receives this packet, it checks whether it has already received it and if so, discards the duplicate; otherwise, it adds its own address to the packet and forwards it to another node. Also, it learns a new route in the network and stores it in its cache for possible use in the future.

3. When the destination receives the RREQ packet, it copies the route from the RREQ packet to an RREP packet and sends it back to the source.

4. Once the source node receives the RREP packet, it begins data transfer and caches the route for further data transmission.

Since the packet header contains the list of the nodes that form the route, the packet size increases as the route gets longer. Because of this, DSR is only suitable for small networks [26].

Route maintenance. Route maintenance is needed when there is a break in the link, which can occur due to the mobility or failure of intermediate nodes [27]. A link break can disrupt ongoing transmission and lead to data loss. Once a link break has occurred:

1. The upstream node notifies the source by sending an RERR packet. All intermediate nodes will mark this route as invalid.

2. When the source node receives the RERR packet, it again broadcasts an RREQ packet to find a new route.

2.3.2 Proactive Routing Protocols (Table-Driven)

Proactive routing protocols are called table-driven protocols since each node maintains one or more tables containing routing information to every other node in the network and when there is a change in the network topology, updating must be undertaken throughout the network [28]. This feature is useful for reducing the network latency, which is the time the request takes to go from the source node to the destination node. However, it causes extensive traffic and high power consumption [29]. In addition, proactive routing protocols are not suitable for large networks since they need to maintain node entries for every node in the routing table of each node. In table-driven protocols, the control messages increase as the network grows, which can rapidly increase the overhead. There are several well-known proactive routing protocols. One of the most common is the destination-sequenced distance vector (DSDV) protocol.

Destination-sequenced distance vector (DSDV)

DSDV is a well-known proactive routing protocol. It is a table-driven routing protocol that uses an improved Bellman Ford algorithm that has been employed successfully in many dynamic packet-switched networks to calculate the path from a source node to a destination node [30]. Because DSDV is table-driven, each node in the network knows a route to each other node before any transmission happens. These routes are stored in the routing table of each node [31]. Each entry in the routing table contains the following fields: the destination address, the next hop, the number of hops and the sequence number. The sequence numbers are used to distinguish invalid paths from fresh ones and to avoid a route loop. Even and odd sequence numbers are used to check whether a link is alive or broken respectively [30]. The nodes build their routing tables by broadcasting HELLO packets to obtain information about neighbouring nodes and available routes. The routing tables are updated regularly and exchanged between the nodes.

To send a data packet to a destination node, the source node will try to send it using the route with the minimum number of hops to save energy consumption. DSDV is efficient in terms of route discovery and there is less delay when sending a data packet through the network than in reactive protocols. However, the overhead can be higher and as a result DSDV does not provide QoS in large networks [32].

2.4 Hybrid trust management schemes

The hybrid trust management scheme combines both direct and indirect trust management techniques. It leverages both the direct observations of node behavior and the feedback received from other nodes to calculate trust values. The idea is to enhance the trust evaluation process by using multiple sources of information, leading to more robust trust decisions.

This thesis did not talk about this approach for the following reasons:

1. Complexity: Hybrid trust management can be more complex to implement and analyze than direct and indirect schemes. Combining both approaches requires careful consideration of how to weigh and combine the different sources of trust information effectively.

2. Overhead: Hybrid trust management may introduce additional communication and computational overhead in the network due to the need to exchange feedback and compute combined trust values.

3. Research Focus: We have limited resources and time to explore various aspects of trust management in MANETs. We prioritized studying direct and indirect schemes

because these two approaches already cover a wide range of use cases and provide valuable insights into trust-related challenges in MANETs.

2.5 MANET routing protocol vulnerabilities

MANET routing protocols are vulnerable to many types of routing attack due to the lack of security features. Any node can connect to the network without going through any authentication or validation process, as MANET routing protocols mechanisms assume that all nodes are cooperative. A malicious node can disrupt the routing process through the following actions [17]:

1. Modifying RREQ and RREP packets.
2. Spoofing the data in the packets
3. Generating fake RERR packets to cause delay in the network by asking the source node to search for unneeded routes.
4. Sending a huge number of fake RREP packets to the source node, which can cause denial of service.
5. Creating fake RREQ packets to keep the nodes busy and consume their resources.
6. Generating loops to resend old packets and thus take up the network bandwidth.

2.6 Trust management

One of the essential goals for MANETs is to provide routing security in the network and ensure confidentiality, integrity, availability and anonymity [19]. The existence and implementation of MANETs depend on the nodes' cooperative and trusting nature. There is a common assumption in existing MANET routing protocols that each node participating in the network is trustworthy [21]. This default assumption needs to change with the development of a mechanism that can distinguish a trustworthy node from a malicious node.

Trust can broadly be defined as a measure of subjective belief that an entity will perform an action and another will perform the promised work without the need to examine whether or not the work is done [33]. This definition contains two inspiring terms: measure and performance. This is how networks work; it is all about performance and the ability to measure it.

A trustworthy entity will always perform the action expected by the trustor. Trustworthiness can be applied to many components in any system, such as devices, nodes and software [15]. The trustworthiness principle has long been used in computer systems and networks. For example, in web services, there are many trust mechanisms that enable a browser to decide whether a particular web page is trustworthy or not. If the website is trustworthy, the browser will access it and show its content; if not, it will show a warning message and block it. In the case of uncertainty, the browser may ask users if they still wish to visit the website and ignore the warning. Theoretically, the same principle can be implemented in MANETs.

Since the relationship between nodes in MANETs starts with a communication and once this communication has ended the relationship ends, any trustworthiness mechanisms will depend on this communication [34]. In using the communication between the nodes, the trustworthiness mechanism should collect data, evaluate the behaviour of the target node and detect malicious activities. MANETS contain no components to be used in creating any trust schema except communications. This is due to the complete lack of a fixed infrastructure. Any designer of trust mechanisms for MANETs will need to analyse the communication between the nodes and determine what appropriate pieces of evidence can be collected from this communication to generate trust values and make decisions. The main idea of using trust management in MANETs is to create a distributed framework in which the nodes do not depend on a third party, such as a server, to detect specific attacks.

In MANETs, trust management is a reputational mechanism such that every node in the network observes and evaluates its neighbouring nodes' activities and tries to detect and isolate any malicious node [21]. It is a method of collecting the information about an entity needed to make a trust relationship decision about it [20]. Trust management can go further and obtain experiences from other entities. Thus, the trust decision can be made using information collected directly or using other nodes' experiences and recommendations.

Trust management has become crucial to solving many performance issues in MANETS. The principle of trust is derived from the social sciences and is used in many fields, such as economics, communications, business and computing and networking. In computer networks in general and in MANETs especially, trust entails believing that a node is cooperating with other nodes to forward the data through the network as expected without any disruption [15].

Ullah et al.[15] defined that the objectives of trust management in MANETs as follows:

1. To distinguish between trustworthy and malicious nodes.
2. To allow trustworthy nodes to participate in establishing routes between sources and destinations.
3. To isolate malicious nodes and prevent them from participating in the network

Due to the unique characteristics of MANETs, a trust management scheme must be distributive and self-organised. Also, it should consume fewer CPU cycles and battery, memory and bandwidth resources.

2.6.1 Trust management motivations

Applying trust management systems in MANETs is based on two main motivations [21]. First, the trust management system helps to detect and isolate malicious nodes to reduce the effect of misbehaving or faulty nodes. Second, the trust management system proposes a probability of the node's future behaviour and exploits the potential for improving the network's quality of service. Because MANETs do not have a centralised control unit to monitor the nodes' operations, a component node should be cautious when communicating with other nodes. The behaviour of the nodes can turn from cooperative to malicious over time and under certain environmental conditions. Therefore, identifying and quantifying the behaviour of nodes from a trusted and reliable perspective is essential to ensure proper operation in MANETs.

The fundamental idea of trust management in MANETs is to observe the behaviour of a neighbouring node and assign a trust value to it [15]. The trust value is determined by how the node behaves. A low value indicates that the node is misbehaving. Only if the node is considered trustworthy based on a high trust value will it be given access to the route that connects the source node to the destination node.

2.6.2 Trust management types

There are three types of trust management in MANETs: direct, indirect and global [34, 21]. Each of these is a general principle that can be implemented through an unlimited number of ideas.

Direct trust

In direct trust management, each node in the network builds its own trust list, which contains a list of nodes and their trust values. It monitors and evaluates the behaviours of neighbouring nodes and assigns trust values to them. It will then only deal with trustworthy nodes and ignore malicious ones [35]. The direct observation strategy works as follows:

1. The trust value lies between 0 and 1.

2. Each node will have an initial trust value of 0.5 (neutral) by default and this value can increase or decrease later depending on how the node behaves.

3. A malicious node will be given a trust value of 0.
4. Each node has a trust list in which it stores the trust values of its neighbouring nodes.
5. Nodes that are considered malicious will be isolated from any communication in the future.

Indirect trust

The indirect trust management principle is the same as the direct principle, but the nodes will also be sharing the trust values they have calculated with each other [36]. Once a node detects a malicious node, it will share this information with the other nodes in the network so they do not need to evaluate the behaviour of the malicious node again.

Global trust

In global trust management, there is a node in the network responsible for storing the trust values and sharing them with the other nodes in the network. Hence, every node in the network needs to talk to this node before it sends any data through the network to establish whether the neighbouring node is trustworthy or malicious. In this thesis, the focus is on direct and indirect trust management.

2.6.3 Trust management steps

Trust management in MANETs comprises four steps: trust initialisation, information collection, trust calculation and decision making [15]. Each step has its own challenges and difficulties in MANETs.

Trust initialisation

Trust initialisation happens when a node starts sending packets to its neighbouring node. In this step, a node in MANET has no previous interaction with its neighbour and has no background about its behaviour. This is a risky situation and most of the direct trust management algorithms proposed consider the default trust value to be neutral. For example, if the range of the trust value is between 0 and 1, the default value for the unknown node is 0.5.

Information collection

In direct trust management, a node directly monitors the following node's behaviour and gathers information based on this observation. Direct observation is a powerful technique

that can provide authentic information about a neighbouring node's behaviour. In indirect trust management, information is collected in two ways: through direct observation and by receiving recommendations from other nodes in the network.

Trust calculating

The information collected about the behaviour of the neighbouring node is used to calculate a trust value for that node. The trust value can be updated over time to arrive at a more accurate judgment. Calculating the trust values can be as simple as counting the packets dropped by the node in question, or it can be more complicated and use maths functions.

Decision making

Once the trust value of a specific node has been calculated, it can be used to decide whether the node is trustworthy or not. Based on this decision, the node in question must either be isolated or allowed to participate in creating routes. There are different levels of trust in more complex models. Figure 2.2 shows the sequence of the trust steps.

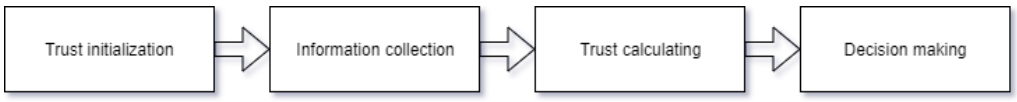

Fig. 2.2 Trust management steps of MANETs

Developing trust management solutions for MANETs is a very challenging task, not only because of the unique characteristics previously mentioned but also because adding any functions to the routing protocols will increase the overhead on the network [16]. Increasing the overhead usually reduces the network's level of performance. For example, more overhead can increase the end-to-end delay. End-to-end delay is the time a packet takes to travel through the network from the source node to the destination node [37]. Adding more functions and tasks to the routing protocols to check the trustworthiness of the neighbouring nodes might increase this time. When the end-to-end delay increases, the network's performance decreases. Thus, when developing a trust management solution, it is necessary to keep an eye on the overload and establish the trade-off between the improvement attained and the overload. If the overload is so high that it is very detrimental to the network's performance, it is not worth implementing the solution, even if it enhances some metrics.

2.7 Related work

Since the AODV routing protocol was introduced, many schemes and solutions have been proposed to improve its performance in the presence of network attacks. Some of these schemes use techniques that require high computational power, such as cryptography. Although many of these solutions improve the performance of MANETs by detecting and isolating malicious nodes, the cooperative nodes suffer from the high computational load required. This may not fit the special characteristics of MANETs. Other solutions rely on the assumption that the first RREP packet received and the RREP packet with the highest sequence number are sent by a malicious node. This assumption is true in the case of black-hole and grey-hole attacks, but not in the case of selfish attack. Various solutions have been suggested to improve the performance of MANETs black-hole, grey-hole and selfish attacks and these are reviewed in the following subsections.

Several solutions and algorithms have been suggested to deal with flooding attacks. The flooding attack model is completely different from the black-hole, grey-hole and selfish attacks. In flooding, the attackers use the RREQ packet to launch the attack, unlike black-hole, grey-hole and selfish attackers, who use the RREP packet. They flood the network with RREQ packets to non-existing destinations to keep the nodes processing these packets for the longest possible time. The solutions previously outlined concentrated on the RREP packets to detect black-hole, grey-hole and selfish attacks. In contrast, flooding attack solutions need to focus on and analyse RREQ packets.

2.7.1 Schemes that focus on only one or two attacks

Jhaveri et al.[38] suggested an observation mechanism to calculate a trust value for neighbouring nodes. This mechanism aims to detect malicious nodes and isolate them. The trust value is calculated dynamically after every time interval using three parameters: (i) the RREP packet sequence number; (ii) the routing table sequence number; (iii) the number of replies received over the time interval. This algorithm is designed to detect and isolate malicious nodes in the route discovery phase. When sending the RREQ packet in the discovery phase, the source node broadcasts a list of malicious nodes. When the intermediate nodes receive this list, they update their routing tables. This trust-based mechanism was found to improve the packet delivery ratio (PDR) under black-hole and grey-hole attacks.

Sharma and Chauhan [39] implemented a trust-based distributed algorithm in the AODV protocol to detect and separate grey-hole attackers from other nodes. Each node in the network monitors the behaviour of its neighbours; when it senses the existence of a grey-hole node, it stores it in a grey-hole attacker list. To confirm the decision, it sends RREQ packets to the other neighbouring nodes to see if the suspected node will still

behave maliciously. Based on the responses received from the neighbours, it can decide whether the suspected node is a grey-hole attacker or not. When it is confirmed that the suspected node is malicious, the source will avoid any future communication with it. This algorithm shows a better throughput in the presence of a grey-hole attack.

Vasantha et al.[40] proposed a trust-based mechanism to improve MANET performance under black-hole and grey-hole attacks. The mechanism detects malicious nodes by filtering their RREP packets at the source and intermediate nodes. The mechanism ignores the RREP packets with very high sequence numbers and also ignores the first RREP packet received. The malicious nodes will be added to a blacklist to prevent any communication with them in the future. The mechanism consists of two stages: prevention and detection. Both stages run while transmitting data between the source and destination nodes.

Jhaveri [41] proposed an algorithm to detect and separate multiple black-hole and grey-hole attackers during the routing discovery phase. They suggested modifying the functionality of receiving RREP packets. Each node in a MANET monitors the behaviour of its neighbouring nodes by analysing the RREP packet received. If an intermediate node detects a malicious node after receiving an RREP packet, it labels the RREP packet DO NOT CONSIDER and marks the node as a MALICIOUS NODE in the routing table. The intermediate node will reverse the path, sending the RREP packet back towards the source node, which will update the routing tables of all nodes with the malicious node entry. The new route towards the destination node is selected by unmarked RREP packets.

Bindra et al.[42] proposed an algorithm to detect black-hole and grey-hole attacks by adding extended data routing information to the routing table in the AODV protocol. The routing table exists in every node in the network and fields are added to detect malicious nodes and save a history of their previous malicious activities. Creating a historical record for each node's malicious activities is a way of addressing grey behaviour. This mechanism is built on the AODV protocol.

Marti et al.[43] proposed an idea called "watchdog", which is a direct trust management mechanism and a path rater. Each node in the network listens to the transmissions of the next node along the route to detect malicious behaviours. The path rater holds the trust values of the nodes, which range from 0 to 0.8, with neutral taking the value of 0.5. These values given to the nodes are updated continuously by 0.1 each 200 ms. The source node should be able to detect selfish nodes and avoid sending packets to them. This scheme has shown better performance in the presence of selfish attacks. However, it also increases the memory overhead as the watchdog mechanism needs to maintain the information collected from the packets.

Buchegger and Boudec [44] proposed an amended routing protocol called Cooperation Of Nodes – Fairness In Dynamic Ad-hoc NeTworks (CONFIDANT). This improved protocol contains a trust management and reputation system. While monitoring the

behaviour of the neighbouring node, the trust manager evaluates the node's activities and sends alarms to neighbouring nodes telling them about a malicious node. The malicious node is then isolated by all nodes that receive the alarm. This mechanism combines direct monitoring and a reputation system. The mechanism has shown better performance in the presence of black-hole attacks.

Michiarde and Molva [45] suggested a collaborative reputation scheme based on collaborative monitoring. This schema has a more complicated reputation system in which it is not only one node monitoring the behaviour of the next node, but many nodes monitoring the behaviour of the same nodes. The decision is made using the data collected by all nodes involved in the network. The decision here should be more accurate than using other methods because it is not made by an individual node. Each node has a reputation table and a watchdog mechanism. Each node monitors the behaviour of its neighbours, assigns trust values to them and saves those values in its trust table. The nodes share their trust values with each other. This mechanism has shown better performance in the presence of selfish attacks.

Pissinou et al.[46] proposed a secure MANET routing protocol based upon a direct trust mechanism establishing the trust level of the nodes along the route. This protocol stores trust values in the RREQ packet. When the source node wishes to initialise a route to the destination, it sends the RREQ packet to its neighbouring node. The intermediate node updates the trust values in the RREQ packet and forwards it to the next node. When the RREQ packet reaches the destination node, the destination node replies to the source by sending an RREP packet containing the level of trust of the route. Then the source can choose the route with the highest trust level rather than choosing the shortest route.

Balakrishnan et al.[47] proposed a reputation-based trust model called secure MANET routing with trust intrigue. The evidence of trustworthiness in this model is collected efficiently from direct interactions with the neighbouring nodes and through recommendations from other nodes. This mechanism contains two components: detection and reaction. In the detection phase, evidence of the malicious behaviour of neighbouring nodes is collected and in the reaction phase, the source node uses this evidence to accept or reject a newly discovered route. Also, in the reaction stage, the source node can predict the future behaviour of a cretin node by using the evidence collected.

Wang et al.[48] proposed an improved trust-based version of the DSR routing protocol. Each node in the MANET has a trust table that stores the trust values of the neighbouring nodes. The amended DSR selects the routing path with the highest trust values and least delay, unlike the standard DSR, which chooses the routing path that has the lowest hop count. The trust values are calculated and updated individually by the nodes. A trust value ranges between -1 and +1, where -1 means distrust and +1 absolute trust. Implementing this mechanism in the DSR protocol showed a better PDR under selfish attacks.

Xin Li et al.[49] proposed a trust algorithm based on packet forwarding ratio (PFR). PFR is calculated using the ratio of the number of packets forwarded to the number of packets received. Trust values are assigned based on the PFR results. The trust value of any node can increase or decrease based on its behaviour in forwarding packets. When the PFR increases, the trust value increases, and when it decreases, the trust value decreases. The trust value in this model is between 0 and 1, where 0 means distrust and 1 means absolute trust. A trust value between 0 and 0.5 indicates the node is malicious, 0.5 to 0.75 indicates the node is suspect, 0.75 to 0.9 suggests the node might be not entirely trustworthy and 0.9 to 1 means the node is trustworthy. Nodes with low trust values are isolated and not allowed to participate in new routes.

Yi et al.[50] suggested a scheme called Flooding Attack Prevention (FAP), which aims to monitor the RREQ packets received from neighbouring nodes. The FAP algorithm assumes that there is less of a risk of a node that sends fewer RREQ packets being a malicious node running a flooding attack. Thus, if a node sends fewer RREQ packets, it is assigned a higher priority for selection. When a node broadcasts a high rate of RREQ packets, the neighbouring nodes will observe this and reduce the responding priority for this node.

Ping et al.[51] added a fixed threshold to the FAP. The modified algorithm assumes that if the RREQ packets received by a neighbouring node exceeds the threshold, it is launching a flooding attack. Any RREQ packets from this node will be ignored in the future. The disadvantage of this modification is that if the attacker guesses the fixed threshold, it is possible simply to make the number of RREQ packets less than the threshold. This way a flooding attack will not be detected.

Song et al.[52] proposed a filtering mechanism that has two thresholds: the rate limit and the blacklist limit. The source node only accepts RREQ packets from nodes that have not reached the rate limit. After receiving the RREQ packet, the source node compares the sequence number of the packet with the one stored in the blacklist. If it is greater than the sequence numbers in the blacklist, the RREQ packet is discarded and the node will be added to the blacklist. The disadvantage of this mechanism is that the attacker can run the attack by reducing the flooding rate and sequence number if he/she discovers the values of the thresholds.

Balakrishnan et al.[47] designed a solution that consists of three thresholds: the blacklist threshold, the whitelist threshold and the transmission threshold. The barrier that prevents a flooding attack in this mechanism is that the RREQ packet received from a neighbouring node is only processed if the transmission rate of the node is less than the transmission threshold. If the transmission range of the node is greater than the transmission threshold, the RREQ packet will be discarded and the node will be added to the blacklist. The nodes that do not exceed the threshold for a given number of times are

added to the whitelist, so there is no need to check the transmission range next time when receiving RREQ packets from them. This technique is useful for keeping the overhead low.

Venkataraman et al.[53] proposed a trust management solution to mitigate the impact of flooding attacks on MANETs. The algorithm of the solution classifies neighbours based on their given trust values under three categories: friend, stranger and acquaintance. A friend node is trusted, a stranger is not trusted and an acquaintance is neutral. There are three transmission rate threshold values and each category has its own threshold value. When the source node receives an RREQ packet from a neighbouring node, it first checks its type and compares its transmission rate to the threshold value of its category. If the transmission rate is below the threshold value for the category, the RREP packet is processed; otherwise, the packet will be discarded and the node will be added to the blacklist.

Bandyopadhyay et al.[54] proposed a scheme that allows a limited number of RREQ packets to be sent by any source node in the network. The limit was set at 10 RREQ packets per second. This low rate helped prevent flooding the network with RREQ packets. The disadvantage of this approach is that the malicious node can change the value of the allowed rate parameter because it has full access to it.

2.7.2 Schemes that require high computational power

Khattak and Nizamuddin [55] provided a hybrid approach for preventing black-hole and grey-hole attacks. The proposed algorithm uses a hash function and timestamp base solution. The source node selects the second shortest route to the destination instead of the first. The probability of the existence of a black-hole attacker decreases for the second shortest route. This is because black-hole and grey-hole attackers send RREP packets to the source stating that they have the shortest route to the destination. Choosing the second shortest path is thus safer in the AODV protocol. The point of monitoring the RREP packets coming from the neighbouring nodes and ignoring the one with the shortest path is a direct trust mechanism for improving MANET performance.

Velloso et al.[20] suggested a recommendation exchange protocol (REP), which enables a node to exchange trust values with its neighbouring node. In this model, the trust values are calculated based on direct observation. The decision to isolate a specific node is made using the evidence collected individually and by recommendations received from neighbouring nodes. Each node in the network assigns a trust value to its neighbours of between 0 and 1. This model consists of two phases: the learning phase and the trust phase. The learning phase is responsible for collecting data and converting it into knowledge. The trust phase is responsible for using that knowledge to detect and isolate malicious nodes.

Yu et al.[56] defined a trust management system that provides a degree of assurance for the future behaviour of nodes based on the services previously received from the

nodes. Yu et al. classified the trust and reputation management system into two major categories, namely individual-level trust and system-level trust. The individual-level trust mechanism allows the source node to initialise communication with the subject node, aggregate declarations from other nodes about prospective communications, evaluate the trustworthiness of potential interaction based on the past recorded data and make a trustworthiness decision on whether to interact with the subject node. The system-level trust mechanism concentrates on applying punishment based on the node's trustworthiness and reputation to improve the utility for nodes that are highly trustworthy.

Lee et al.[57] suggested adding two new packets to the AODV protocol: a route confirmation reply (CREP) packet and a route confirmation request (CREQ) packet. When an intermediate node receives an RREQ packet from the source node, it checks its routing table and sends an RREP packet back to the source and sends a CREQ packet to its next-hop node towards the destination node. When the next-hop node receives the CREQ packet, it sends a CREP packet back to the source node if it has a fresh route to the destination node. When the source node receives the CREP and RREP packets, it validates the path to the destination by comparing the paths in the two packets. If both agree, the path is valid.

Al-Shurman et al.[58] proposed a solution requiring additional computation which results in an increased overhead. The source node stores the sequence numbers of the last packets sent and the last packets received in two separate tables. When the source node receives an RREP packet, it compares the sequence number in this packet with the stored numbers. If there is a match, the source node starts data transmission; otherwise, it will classify the RREP packet as malicious. The source node will send an alarm message to the other nodes to block the malicious node.

Raj and Swadas [59] designed a solution that adds an additional task for the nodes: checking the RREP packet before accepting it. When a node receives an RREP packet, it checks if the sequence number is higher than a threshold value. If the sequence number of the RREP packet is higher than the threshold, it is classified as malicious and the node that sent it is added to a blacklist. When a node has detected a malicious node, it sends an alarm packet to inform all neighbouring nodes. This solution has been found to increase the overhead.

Mistry et al.[60] suggested a solution that entails analysing the RREP packets received. When a source node receives the first RREP packet, it does not react to it immediately but waits for some time to receive multiple RREP packets and then saves them in a table. The source node then analyses the stored RREP packets in the table and rejects those with very high sequence numbers. The nodes that sent these packets are marked as malicious. The source node will arrange the remaining RREP packets according to the sequence numbers of their destination nodes, and the one with the highest number will be selected. This

mechanism increases the end-to-end delay as the source node has to wait some time to receive multiple RREP packets.

Research	Scheme	Lack
Jhaveri et al.[38]	An observation-based trust calculation approach for neighboring nodes, with the goal of detecting and isolating malicious nodes	Only tackles black-hole and grey-hole attacks
Sharma and Chauhan [39]	Each node in the network observes its neighbors' behavior and identifies the presence of a grey-hole node, which it then adds to a grey-hole attacker list.	Only tackles grey-hole attacks
Vasantha et al.[40]	The mechanism detects malicious nodes by filtering RREP packets at source and intermediate nodes, disregarding those with very high sequence numbers and the first received RREP packet.	Only tackles black-hole and grey-hole attacks
Jhaveri [41]	They proposed altering the way RREP packets are processed, wherein each node in a MANET examines the behavior of its neighboring nodes by analyzing the received RREP packets.	Only tackles black-hole and grey-hole attacks
Bindra et al.[42]	An algorithm is introduced to identify black-hole and grey-hole attacks by incorporating extended data routing information into the routing table within the AODV protocol.	Only tackles black-hole and grey-hole attacks
Marti et al.[43]	The idea of "watchdog" is a direct trust management mechanism and path rater where each node in the network monitors the next node's transmissions along the route to detect malicious behaviors.	Only tackles selfish attacks

Buchegger and Boudec [44]	The trust manager monitors the neighboring node's behavior, evaluates its activities, and alerts neighboring nodes about malicious nodes, leading to the isolation of the identified malicious node by all nodes in the network.	Only tackles black-hole attacks
Michiarde and Molva [45]	A reputation scheme that relies on collaborative monitoring for evaluation.	Only tackles selfish attacks
Pissinou et al.[46]	An secure MANET routing protocol that utilizes a direct trust mechanism to establish trust levels of nodes along the route.	Only tackles black-hole attacks
Balakrishnan et al.[47]	The model efficiently collects evidence of trustworthiness through direct interactions with neighboring nodes and recommendations from other nodes, comprising two components: detection and reaction.	Only tackles black-hole attacks
Wang et al.[48]	Unlike the standard DSR, the modified version prioritizes the routing path with the highest trust values and minimum delay, rather than the one with the lowest hop count.	Only tackles selfish attacks
Xin Li et al.[49]	The trust algorithm utilizes packet forwarding ratio (PFR) to calculate trust values, which are determined by the ratio of forwarded packets to received packets.	Only tackles black-hole attacks
Yi et al.[50]	The FAP algorithm operates on the assumption that nodes sending fewer RREQ packets pose a lower risk of being malicious nodes running a flooding attack.	Only tackles flooding attacks

Ping et al.[51]	The modified algorithm considers a neighboring node to be launching a flooding attack if it receives RREQ packets exceeding a threshold, leading to the subsequent ignoring of any RREQ packets from that node.	Only tackles flooding attacks
Song et al.[52]	The filtering mechanism employs two thresholds, the rate limit and the blacklist limit, allowing the source node to accept RREQ packets only from nodes that haven't reached the rate limit.	Only tackles flooding attacks
Balakrishnan et al.[47]	The proposed solution comprises three thresholds: the blacklist threshold, the whitelist threshold, and the transmission threshold.	Only tackles flooding attacks
Venkataraman et al.[53]	The solution's algorithm categorizes neighbors into three groups: friend, stranger, and acquaintance, based on their assigned trust values.	Only tackles flooding attacks
Bandyopadhyay et al.[54]	The scheme restricts each source node in the network to send a maximum of 10 RREQ packets per second, ensuring a limited number of transmissions.	Only tackles flooding attacks
Khattak and Nizamuddin [55]	The proposed algorithm employs a hash function and timestamp-based approach, with the source node choosing the second shortest route to the destination instead of the first.	High computational power
Velloso et al.[20]	The model involves two phases: learning (collecting and converting data into knowledge) and trust (using the knowledge to detect and isolate malicious nodes).	High computational power

Yu et al.[56]	The trust and reputation management system was divided into two main categories: individual-level trust and system-level trust.	High computational power
Lee et al.[57]	The AODV protocol was extended by introducing two additional packets: a route confirmation reply (CREP) packet and a route confirmation request (CREQ) packet.	High computational power
Al-Shurman et al.[58]	The source node maintains two separate tables for the last sent and received packets' sequence numbers, comparing the sequence number in the received RREP packet with the stored values.	High computational power
Raj and Swadas [59]	The solution involves introducing an extra responsibility for the nodes, which includes inspecting the RREP packet before accepting it.	High computational power
Mistry et al.[60]	Upon receiving the initial RREP packet, the source node refrains from immediate response and instead waits for a certain period to collect multiple RREP packets, which are then stored in a table.	High computational power

2.8 Proposed approach to detect black-hole, grey-hole and selfish attacks

In the scheme proposed in this research, the source node monitors the behaviour of the neighbouring node by sending several packets and watching to see if the neighbouring node forwards them or not. If the neighbouring node forwards them, we assume that this node is cooperative. If not, we assume the node is malicious. This scheme should be able to detect black-hole, grey-hole and selfish attacks. The three attacks have one common behaviour, namely dropping the data packets instead of forwarding them. However, each attack has its own method of dropping packets. The black-hole attack drops all packets

received, the grey-hole attack drops some packets and passes others on, and the selfish attack drops all packets received but passes on the packets that are in its interests. Thus, it is necessary to modify the method used to calculate the trust value in the scheme slightly to suit each type of attack. This scheme is implemented by adding a trust table and a packet list to each node to analyse the packets sent and store the trust values. The scheme is explained in detail in Chapter 4.

2.9 Proposed approach to detect flooding attacks

The proposed approach used to detect flooding attack addresses two issues. The first is the number of RREQ packets that a source node can send and the second is the length of time the RREQ packet can live in the network. For a flooding attack to be successful and achieve its goal of affecting the network's performance, it needs to flood the network with the highest possible number of RREQ packets and ensure they live in the network for the longest possible time. By default, MANET routing protocols have a variable called time to live (TTL) which determines the length of time an RREQ packet can live in the network to search for a route before it gets discarded. Also, they do not allow unlimited RREQ packets to be sent by any node in the network to prevent flooding. However, these two values are accessible by any node in the network, including malicious nodes. A malicious node can modify the values of these variables and replace the default values with high values. In this way, the malicious node can send as many RREQ packets as it wishes and have them live in the network for whatever length of time it specifies. The approach in this thesis will have the intermediate nodes calculate these two values rather than relying on the values received form the source. The values will not be accessible by the source node to prevent any malicious amendments. The scheme is explained in detail in Chapter 5.

2.10 Novelty of the proposed scheme

The works previously cited improved MANET performance in the presence of attacks, but with some limitations:

1. Most require high computational resources, such as encryption, which are not always available or possible in MANETs.

2. Some of these schemes lead to high overhead. For example in the scheme suggested in [58], the end-to-end delay reached 1.5 seconds when the pause time was zero. In [59] the proposed scheme has raised the end-to-end delay to 0.27 seconds. In the proposed scheme in this thesis that tackles the black-hole and grey-hole attacks, the highest end-to-end delay was 0.017 seconds which is shown in section 4.7.

3. They assume that the RREP packet with the highest sequence number and the first RREQ packet that arrives are the only malicious ones, which is not always the case.

4. Each solution was evaluated against only one or two attacks. For example, many studies address the problem of black-hole attacks and assume that if the solution works against this type of attack, it will work against grey-hole and selfish attacks. This is a false assumption as grey-hole and selfish attacks are slightly different in the way they work and the effects they have.

5. In the case of flooding attacks, most of the solutions have used a fixed threshold to limit the number of RREQ packets sent and thus reduce the potential for attack. However, if an attacker finds out the threshold, he/she can set the flooding rate to be less than the threshold. Thus, the flooding will not be detected.

The schemes in this thesis consider the principle of trust management and does not use any techniques that require high computational loads, such as cryptography, which can lead to high overload. Also, it does not rely on the assumption that the first RREP packet received and the RREP packet with the highest sequence number are malicious and others are not. This assumption can lead to categorising a normal cooperative node as malicious. Also, the evaluation of the proposed trust management solution is extended to encompass a variety of attacks: black-hole, grey-hole, selfish and flooding. In terms of the flooding attack, the threshold will be dynamic rather than fixed to make it harder for the attacker to estimate the threshold value.

2.11 Methodology

The research methodology is a set of methods, principles and practices used to conduct a study. In a PhD, the methodology provides an organised approach to analyse data and evaluate outputs systematically, thus answering the research questions [61]. Typically, a PhD research project aims to conduct an investigation of particular subject matter. Selecting the appropriate methodology is essential to accomplish a the research project successfully. The methodology in this thesis is simulation.

Simulations are used to conduct research examining the performance and security of network routing protocols. Most of the popular network protocols, such as the Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Transmission Control Protocol (TCP), have been developed and improved over time using simulations as a main part of conducting the experiments to test them. Using simulations allows the researcher to create a controlled environment in which the outputs can be established with greater accuracy than in open environments. The researcher can create a virtual network with specific

configurations, hardware and settings for the protocols. This can then be simulated, run, observed and studied. Network simulations can reflect real-world scenarios in terms of mobility patterns, hardware specifications, node mobility speed and all other parameters.

In the network performance field, simulations allow the researcher to add loads to the virtual network and vary parameters, such as the number of nodes, area dimensions, pause time, node mobility speed and requests. This is very useful to evaluate the performance of the network under different scenarios.

In the context of measuring the performance of mobile ad hoc networks (MANETs), three primary research methods are available:

1. Mathematical Modeling: This method involves developing mathematical equations and analytical models to describe the behavior and performance of MANETs.

2. Direct Experiments: This method involves setting up physical MANETs and conducting real-world experiments to observe their performance.

3. Simulation: Simulation involves using specialized software or tools to create a virtual environment that emulates the behavior of a MANET.

Simulation is an appropriate method for measuring the performance of mobile ad hoc networks due to its cost-effectiveness, reproducibility, control, and scalability. It provides a safe and controlled virtual environment to explore various network scenarios and parameters, enabling researchers to gain valuable insights without the expense and limitations of direct experiments or complexities of mathematical modeling.

Validation in the context of simulation refers to the process of ensuring that the simulated model accurately represents the real-world system it is intended to mimic. It is a critical step in simulation research to establish the credibility and reliability of the simulation results. The goal of validation is to demonstrate that the simulation outcomes closely match the actual behavior and performance of the target system. The validation process involves comparing the output of the proposed schemes with the plain routing protocol AODV.

In this thesis the simulations were run using the network simulators NS-2 and NS-3 and some other helpful tools for generating data and formatting outputs.

2.11.1 NS-2.35 network simulator

There are several simulation tools available for studying mobile ad hoc networks and other communication systems such as NS2, NS3, OPNET Modeler, OMNeT++, and QualNet. Here are some of these simulators features:

NS2

1. Language: NS2 uses Tcl (Tool Command Language) for scripting.
2. Community: It has a strong user community and extensive documentation.
3. Flexibility: Provides a high degree of flexibility for protocol and scenario customization.
4. Protocols: Supports a wide range of network protocols and communication models.
5. Visualization: Offers basic visualization capabilities for network behavior.

NS3

1. Language: NS3 uses C++ and Python for simulation scripting.
2. Modularity: Designed for modularity and extensibility, making it easier to incorporate new protocols and features.
3. Realism: Focuses on more realistic models and simulations, which can lead to more accurate results.
4. Community: Has an active community, but documentation might not be as extensive as NS2's.

OPNET Modeler

1. Graphical Interface: Offers a graphical modeling environment.
2. Commercial Tool: Originally a commercial tool, it provides a visual way to design and simulate networks.
3. Protocols: Supports a variety of protocols and technologies.
4. Cost: Licensing costs may be a consideration for academic users.

OMNeT++

1. Modularity: Similar to NS3, OMNeT++ emphasizes modularity and component-based simulation.
2. Framework: Provides a flexible framework for building simulations using C++.
3. Community: Active community with a focus on discrete event simulation.
4. Visualization: Offers visualization tools for analyzing simulation results.

QualNet

1. Commercial Tool: A commercial simulation tool known for its support for large-scale network simulations.
2. Features: Offers a range of features for wireless network simulation, including MANETs.
3. Ease of Use: Known for being user-friendly, with a graphical interface and script-based options.
4. Protocols: Supports various communication protocols and technologies.
5. Cost: As a commercial tool, it comes with licensing costs.

NS3 is a powerful and versatile open-source network simulator that offers realism, accuracy, extensive protocol support, and active community engagement. These factors justify its use for measuring the performance of mobile ad hoc networks. This thesis started with using NS2 as it is still reliable and used by researchers and university around the world and then later we upgraded to NS3.

NS-2 is an open-source tool designed to help researchers explore networks, measure the performance of routing protocols and evaluate proposed solutions [11]. This tool helps in designing and enhancing routing protocols for both wired and wireless networks. NS-2 is widely used by universities and researchers for studying the performance of routing protocols and evaluating new proposed solutions.

NS-2.35 is an objective-oriented event simulator. It was introduced in 1995 by the Virtual InterNetwork Testbed (VINT) project at the University of California and was improved some years later by the Monarch Project at the University of Carnegie Mellon to enable it to support node mobility [62]. It enables simulation of the services and protocols used by wired and wireless networks and has the capability to analyse the behaviour of both existing and new protocols. The ability of NS-2.35 to generate many scenarios rapidly with random movement and traffic patterns is one of its key advantages. All the necessary features, including abstraction, visualisation, emulation, traffic production and scenario production are presented in NS-2.35. A wide range of MANET routing protocols, including AODV, DSR, DSDV, and many others, are supported by NS-2.35.

Two programming languages are used in NS-2.35: the Object-oriented Tool Command Language (OTcl), which is used to design and configure networks, and C++, which describes the internal functioning of the simulation. OTcl is quick to consider changes, making it ideal for configuring the network, while C++ is fast to run, making it suited for running the simulation. NS-2.35 was designed to combine the benefits of these two languages.

2.11.2 NS-3.33 network simulator

The Network Simulator 3 (NS-3.33) is an open-source and discrete event simulation platform designed for simulating and modelling many types of computer networks. It is used to study how routing protocols behave in different network scenarios. Unlike the previous version, NS2, NS-3.33 is designed to be more adaptable and flexible, with a greater selection of components for different types of networks, including MANETS. NS-3.33 is developed in C++ and it also supports Python scripting language. By supporting a modern and powerful scripting language like Python, NS-3.33 provides enhanced capabilities compared to NS-2.35.

2.11.3 Network simulator generator:

NSG is a scenario generator for NS-2.35 that automatically generates Tool Command Language (TCL) scripts [63]. It is a Java-based tool that can generate TCL scripts for wired and wireless networks. NSG has a friendly graphical user interface that gives the user the ability to create any desired network scenarios without coding. This tool has been used widely by researchers and engineers to create network scenarios.

2.11.4 Bash scripting

Bash is an interpreter that processes shell commands. A command that requests the use of operating system services is issued in the form of plain text. A Bash script is a text file containing a series of commands. A Bash script can contain any command that can be run at the terminal. Any sequence of terminal commands can be written as a Bash script in a text file.

Bash scripting is used to automate the execution of tasks so that researchers do not need to run them individually and repetitively. It is an excellent way to automate many different tasks in a computer system. In NS-2.35, using scripts, it is possible to automate the creation of trace files, the execution of simulations and the analysis of the outcomes. This can be helpful for quickly and effectively testing a range of various scenarios.

A bash script consists of a series of commands and functions that are used to initialise and configure a simulation's parameters, run the simulation and analyse the results. For example, a basic bash script can be used to generate a trace file for simple node-to-node communication and then execute multiple simulations with different parameters, such as node mobility speed and pause time. The most useful feature of bash scripts is that they can be reused for a variety of different simulations. The parameters and variables in the script can be changed simply to generate different scenarios.

2.11.5 Confidence interval

A confidence interval (CI) is a range of values within which a result is estimated to lie with a certain probability [64]. When measuring system performance, it is essential to calculate the CI to establish the accuracy of results as it shows the reliability of the evaluation. At a certain level, usually 95%, we can be relatively confident that the results are accurate. This helps reduce uncertainty when making decisions based on the results. Each single experiment in this thesis was run multiple times and the uncertainty surrounding each value was calculated using the 95% CI.

2.12 Summary

This chapter has shown a solid understanding of the research field, problem and the potential solution. It has proposed two approaches to handle network performance in the presence of attacks. It has set out the methodology to be used and the tools employed to assist in the experiments.

The next chapter explores the effects of node mobility on MANET performance. As previously mentioned, the mobility of nodes is the most unique characteristic of MANETs and must be considered in the design of any solution. The chapter includes an analysis of the effects of varying the node mobility speed on the performance of three MANET routing protocols, i.e. AODV, DSR and DSDV. This addresses several questions: Does increasing the speed of the nodes' mobility affect the performance of the network? Which metrics are most affected? Do all three protocols present the same or different effects and why? This analysis is undertaken in the ideal situation when all nodes are cooperative and in the absence of any malicious activities. Understanding the effect of node mobility on MANET performance will help measure the success of the proposed solutions later.

Chapter 3

Analysing the Effect of Mobility on the Performance of MANET Routing Protocols

3.1 Introduction

When conducting experiments to measure the performance of a system, it is very important to look closely and carefully at the metrics and parameters that will be used. Since many metrics are used to measure the performance of MANETs [5], we need to choose those that can help measure the success of the proposed solutions. Also, we need to select the parameters that have a direct impact on the metrics both when the network is under attack and in the ideal situation when there is no attack. The first step is to explore MANET performance under normal conditions, when there is no malicious activity, and try to understand which metrics are affected by which parameters. This approach makes it more feasible to then attain an accurate evaluation offering potential direct trust management schemes.

A node in a MANET acts as a source, destination or a router when participating in a route [65]. Because routes in MANETs are multi-hop based, the possibility of route failure is very high. In multi-hop routes, if a single node moves outside the transmission range of either of its neighbours, the whole route fails. Route failures are considered a key factor that impacts the performance of all MANET routing protocols [22]. Furthermore, there is an effect on establishing a new route when nodes remain out of transmission range for a long time. This effect, termed pause time, refers to when a node ceases moving. Nodes moving and stopping are the first factor to be taken into account in any routing protocol algorithm in MANETs.

Since one node moving outside the transmission range of its neighbouring nodes in the route leads to a route break, the number of broken routes will increase as the speed of the nodes increases. There is a direct relationship between the speed of node mobility and the number of broken links [66]. For the transmission of a data packet from a source node to a destination node to be successful, a multi-hop route must not undergo any route failure [22]. If a route does break, a fresh one must be established as quickly as possible. However, this is delayed in the case of high pause times and failure to establish a fresh route rapidly will clearly reduce the network performance.

The impact of mobility has been reported in many studies. These have demonstrated that the speed of node mobility and pause time parameters have direct effects on metrics such as throughput and PDR, as well as some others. Qin et al.[10] analysed the effect of the speed of node mobility on the PDR, the number of link breaks and link duration. They found that varying the speed of the nodes' mobility led to different levels of performance. Ananad et al.[67] showed the direct impact of varying node mobility speeds on throughput and end-to-end delay in three routing protocols, AODV, DSR and DSDV. They concluded that there were differences in the effects of node mobility speed depending on the protocol. Gujral et al.[68] analysed the effect of node mobility speed, transmission range and the number of nodes on the performance of MANETs in terms of throughput, end-to-end delay and battery consumption. They used three routing protocols, AODV, DSR and DSDV. The results of their simulations showed that the performance of all protocols was affected, with some differences.

The effect of node speed may vary from one routing protocol to another as each protocol uses a different algorithm to establish and maintain routes. Thus, node mobility and pause time should have different effects on the performance of reactive and proactive protocols.

This chapter analyses the effect of node mobility speed and pause time on the performance of MANETs. For the analysis, we vary the values of the node mobility speed and pause time to observe the effects on two metrics: throughput and PDR. All simulations in this chapter were run when all nodes were cooperating ideally in forwarding packets, without any malicious activity.

3.2 Mobility models

Node mobility is the main attribute of MANETs and their performance must be studied under the condition of mobility [69]. Real-life mobility patterns are varied and can be very complex to model depending on the mission and objectives of the participating nodes. The more complex the mobility patterns, the more difficult they are to model and simulate as

more details need to be added. The mobility of nodes in MANETs causes the topology of the network to change frequently and MANETs are expected to provide a decent level of performance during such dynamic changes.

The performance of MANET routing protocols is greatly influenced by dynamic changes in the network topology. Moreover, the same mobility model can yield different results for different routing protocols. After selecting the good solution according to the experimental setup model to reflect the real-life scenario, the mobility parameters should be varied.

There are various mobility models, such as random walk (RW), random waypoint (RWP), geographic constraint mobility (GCM), random Gauss–Markov (RGM), fluid-flow mobility (FFM), gravity mobility (GM) and behavioural mobility (BM) [7]. In the case of MANETs, RWP mobility is the most relevant. This pattern can reflect a range of real-life scenarios, such as a group of people communicating inside or outside a building, a number of soldiers communicating in a rough area, or a group of volunteers helping people in a disaster situation, such as an earthquake. In RWP mobility, a node moves freely in random directions at random speeds.

3.2.1 Random waypoint (RWP) mobility

The RWP model was first defined by Johnson and Maltz in 1996 to study and evaluate the performance of MANET routing protocols. This model is considered the first attempt to create a mobility model simulating human movement [70] and it is the most common mobility pattern used in MANETs [7]. In this mobility model, a mobile node moves from its position to a new position by randomly selecting the destination coordinates, its speed and its pause time when it arrives at the new location. When the node reaches the destination, it pauses for a random amount of time - the pause time - then moves to a new destination and so on. When the pause time expires, the mobile node randomly chooses another destination, speed and pause time.

The RWP model has three variables: (V_{max}, T, and V_i). The node speed varies randomly from 0 to V_{max}, T is the pause time and V_i is the direction that the node will take [70]. In RWP, the node chooses a random destination, called the waypoint, and travels to it at a constant speed chosen randomly from the range (0, V_{max}). After the node reaches its destination, it pauses for a period of time equal to T, then it moves to the new destination. Figure 3.1 shows a sample pattern of a node's movement in the RWP model.

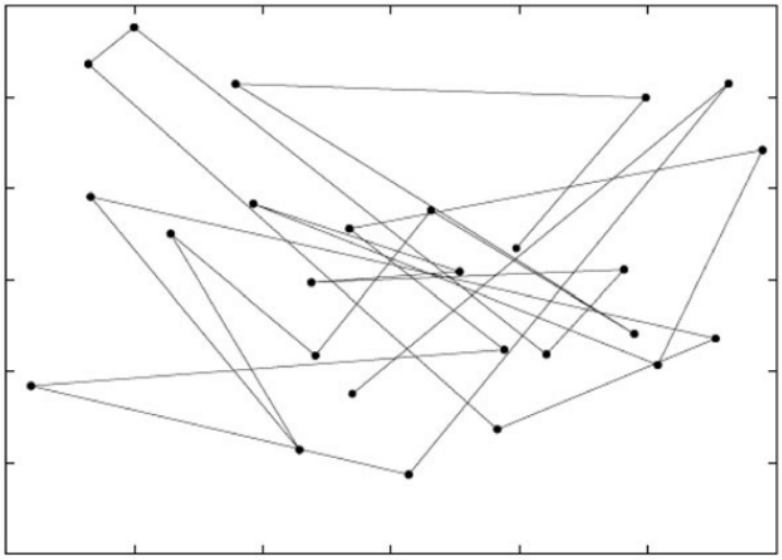

Fig. 3.1 Travel pattern of the random waypoint (RWP) mobility model [70]

3.3 MANET performance

In MANETs, mobility has a significant impact on the throughput and PDR. MANETs are composed of wireless nodes that interact with one another in an ad hoc way and therefore the nodes have the potential to change position constantly. Thus, the availability of a communication link or route between two nodes may alter. If a node moves out of an area where another node is located, they will no longer be able to interact, resulting in a reduction in throughput and PDR. In addition, when a node moves, its signal can get weaker, leading to fragmentation, which makes it more difficult to send and receive packets in a timely manner. Moreover, an interruption in the connection between nodes can result in an increase in collisions and lost packets, which can further impair the throughput and PDR. Therefore, mobility plays a crucial role in the performance of MANETs and must be handled appropriately for the network to run at a high level of performance.

In addition to throughput and packet delivery ratio (PDR), other metrics used to measure the performance of Mobile Ad hoc Networks (MANETs) include end-to-end delay, network coverage, energy consumption, jitter, network connectivity and packet loss rate. These metrics provide valuable insights into various aspects of the network's behavior, such as energy efficiency, connectivity, and stability.

Network coverage is the extent to which the communication range of nodes covers the entire area of interest. It indicates the portion of the network that is within the reach of one or more nodes for effective communication. Energy consumption is the amount of energy expended by individual nodes in the network while performing communication, processing, and other tasks. It can also refer to the overall energy usage of the entire network. Jitter

represents the variation in packet arrival times at the destination node. It is the difference in delay between consecutive packets and indicates the variability in transmission and queuing times. Network connectivity refers to the degree to which nodes in the network can communicate with each other through direct or indirect paths. Packet loss rate is the percentage of transmitted packets that do not reach their intended destination successfully.

Using throughput and PDR is the most sufficient in terms of measuring the success of proposed trust schemes due to their simplicity and direct impact on the network's efficiency and data delivery. They offer quick assessments of performance and allow for relative comparisons between different scenarios and protocols.

3.3.1 Throughput

Throughput refers to the volume of data packets moved successfully from the source node to the destination node over a specific period of time, measured typically in bits per second (bps) [23]. A higher throughput value denotes better routing protocol performance. This metric is crucial and sensitive and may well be the most important in many applications since it indicates how effectively the network can serve an application. A network with low throughput can support fewer applications and services and data transmission rates will be slower. For MANETs to run reliably and effectively, it is essential that they have solid throughput and thus this parameter must be always taken into account when measuring MANET performance.

This thesis uses the throughput only, not the goodput. Goodput focuses on the useful data that reaches the destination application, ignoring any extra data that might be necessary for communication purposes such as the RREQs, RREPs, and RERRs. Goodput is also measured in bits per second (bps) or kilobits per second (Kbps).

Throughput measures the rate at which data is successfully transferred between source and destination nodes in a network. It represents the total volume of data that passes through the network in a given time period. In terms of data, Throughput includes all data, including the useful data payload and any overhead introduced by headers and control packets. In terms of packets, It considers all packets sent and received, regardless of their size or purpose (data, control, acknowledgments, etc.).

Goodput measures the rate of useful data that successfully reaches the destination application and is delivered to the end user or application. It focuses on the actual application data without considering any overhead due to control packets or headers. In terms of data, it excludes any overhead introduced by headers and control packets. It only considers the data that carries meaningful information for the application. In terms of packets, goodput is generally not measured in terms of packets, as it's more concerned with the actual data content rather than the number of packets.

3.3.2 Packet delivery ratio (PDR)

The PDR is the number of packets received by the destination node divided by the number of packets sent from the source nodes [5]. The PDR is typically associated with the throughput as they tend to increase and decrease in tandem.

In MANETs, the PDR is an important metric for measuring network performance, indicating the frequency with which data packets sent by a source node are received by the destination node. A high PDR shows good network performance, with low packet loss due to efficient routing, whereas a low PDR denotes poor network performance, with high packet loss due to congestion, interference, or a dynamic topology. A low PDR may be the result of an inefficient routing protocol, incorrect or out-of-date routing tables, or poor signal quality. With a decreasing PDR, it is likely that other factors, such as an increase in packet delay, will affect the usability of the network application.

Throughput and Packet Delivery Ratio (PDR) are correlated metrics in MANETs. Higher PDR, indicating successful packet delivery, generally leads to higher throughput as it implies more efficient resource utilization, reduced retransmissions, and good routing, contributing to an improved overall data transmission rate in the network.

3.4 Tools used in the experiments

In this set of experiments, we used the NS-2.35 network simulator, a network simulator generator (NSG) tool and a mobility generator tool called setdest. Also, we developed a bash script code to automate the simulation and to read the results from the trace files and present them in a readable format. The programming languages used to conduct the experiments were TCL, C++ and bash scripting. The simulations took different amounts of time, from 10 seconds up to one hour, depending on the parameter values.

3.5 Simulation and evaluation

Network simulation is the process of establishing a virtual representation of a mechanism or a network to analyse its behaviour and evaluate a given system [71]. Simulations help researchers and engineers understand the performance characteristics of target networks and identify potential problems and solutions. Simulation is typically used as a resource in the design of new schemas and routing ptotocols.

Our experiments here aimed to study the effect of mobility on the performance of three routing protocols, AODV, SDR and DSDV. The simulation model had two variable parameters: speed of mobility (metre/second [m/s]) and pause time (seconds [s]). The mobility speed was set to increase from 10 to 50 m/s. The pause time started at 5 s and

increased by 5 s each time until it reached 50 s. The simulation ran for 3600 s. It was repeated 10 times and the mean was taken as the result for every metric. The metrics used in the experiment were throughput and PDR. For all simulations, the confidence interval (CI) was set at 95%.

3.5.1 Change in throughput varying the node mobility speed

Figure 3.2 shows the change in throughput when varying the range of the node mobility speed from 10-50 m/s in steps of 10 m/s, with fixed simulation parameters as shown in Table 3.1.

Table 3.1 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	NS-2.35	-
MAC protocol	IEEE 802.11	-
Antenna	omni antenna	-
Mobility model	random waypoint	-
Simulation time	3600	second
Simulation area	800 * 800	metre
Number of nodes	10	-
Packet size	512	byte
Pause time	5	second
Traffic type	UDP	-
Transmission range	250	metre
Routing protocols	AODV, DSR and DSDV	-

The parameter settings chosen for measuring the performance of MANET routing protocols while varying the mobility speed of the nodes are appropriate as following:

1. Mobility Model - Random Waypoint: Random Waypoint is a commonly used mobility model for simulating realistic movement patterns of nodes in MANETs. It allows nodes to move randomly within the simulation area, making it suitable for evaluating the adaptability of routing protocols under different node movement scenarios.

2. Simulation Time - 3600 seconds: A simulation time of 3600 seconds (1 hour) provides sufficient duration to observe the network behavior over an extended period. This duration allows the network to reach a steady-state, and the protocols can demonstrate their effectiveness under varying mobility speeds.

3. Simulation Area - 800 * 800: The chosen simulation area size of 800 * 800 meters provides a reasonable spatial range for evaluating the protocols' performance. It

ensures that nodes have ample space to move and interact within the network, simulating realistic MANET scenarios.

4. Number of Nodes - 10: Using 10 nodes is appropriate for small-scale simulations. It allows for manageable computation times while still providing insights into the impact of mobility speed on the network when considering a limited number of nodes.

5. Packet Size - 512 bytes: A packet size of 512 bytes represents a common data packet size used in various real-world applications. Evaluating routing protocols with different packet sizes helps understand how they perform with various traffic patterns.

6. Pause Time - 5 seconds: The chosen pause time of 5 seconds between node movements introduces realistic behavior as nodes may pause during their movement. This variation in movement patterns can significantly impact the efficiency of routing protocols.

7. Traffic Type - UDP: Using UDP (User Datagram Protocol) as the traffic type helps assess the protocols' performance under connectionless and unreliable traffic conditions. It is common for multimedia and real-time applications in MANETs.

8. Transmission Range - 250 meters: The chosen transmission range of 250 meters is relevant for MANETs, as it defines the communication range between neighboring nodes. This range setting allows researchers to analyze the impact of node mobility on connectivity and packet forwarding within the transmission range.

These parameters provide a realistic simulation environment and allow for meaningful observations regarding the effectiveness and adaptability of routing protocols in dynamic mobile scenarios throughout the thesis.

The parameter "Antenna = omni antenna" refers to the type of antenna used by the nodes in the network. An omni antenna is an omnidirectional antenna that radiates and receives signals in all directions equally. In other words, it has a 360-degree coverage pattern, providing a spherical radiation pattern around the antenna's axis.

Topology

The initial network topology refers to the arrangement of nodes and their connectivity at the beginning of the simulation before any node movements occur. At this point, the nodes are typically distributed randomly within the simulation area, and their communication links are established based on the transmission range of their omni antennas.

As the simulation progresses and the mobility model (random waypoint) comes into play, the individual nodes start to move following their predefined mobility patterns. The mobility model determines the speed, direction, and pause times of each node, leading to dynamic changes in the network topology over time but all simulations start with the same topology at the beginning.

As nodes move, their positions change, affecting their connectivity with neighboring nodes. The network topology becomes dynamic, with nodes forming and breaking communication links as they move closer or farther away from each other. This is how the initial topology started in this simulation and in all simulations in the thesis.

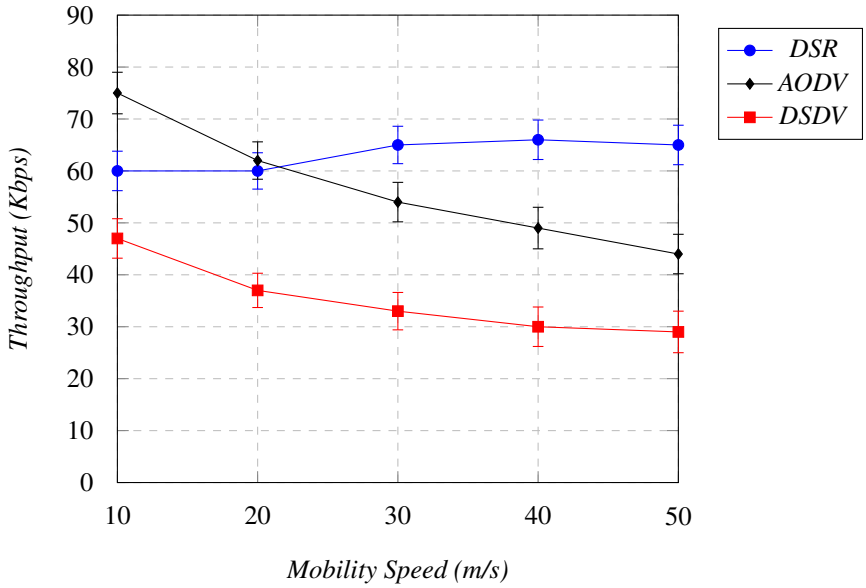

Fig. 3.2 Throughput vs. mobility speed (95% CI)

Figure 3.2 shows the impact of increasing the mobility speed on the performance of the three protocols, AODV, DSDV and DSR, in terms of throughput. We can see that as the mobility speed increases, the throughput for both the AODV and DSDV protocols decreases, whereas the throughput for DSR is stable and does not seem to be affected by the increase in node mobility speed.

Initially, the performance of the DSR protocol is lower than that of AODV, but as the speed of node mobility increases, the throughput also increases. DSDV presents the worst performance of the three. This is because proactive protocols require the routing tables to be updated and maintained and an increase in the node mobility speed results in more routing maintenance and greater frequency of updates.

Overall, the network dimension here is 800 * 800 m and thus we do not expect to see a major drop in the throughput level with an increase in speed. Although both the reactive AODV and DSR protocols and the proactive DSDV protocol react similarly in

the simulation of low to high node mobility speed, the DSDV protocol exhibits poorer performance compared to the AODV and DSR protocols, with the performance of the DSR protocol being best overall.

3.5.2 Change in packet delivery ratio (PDR) varying the node mobility speed

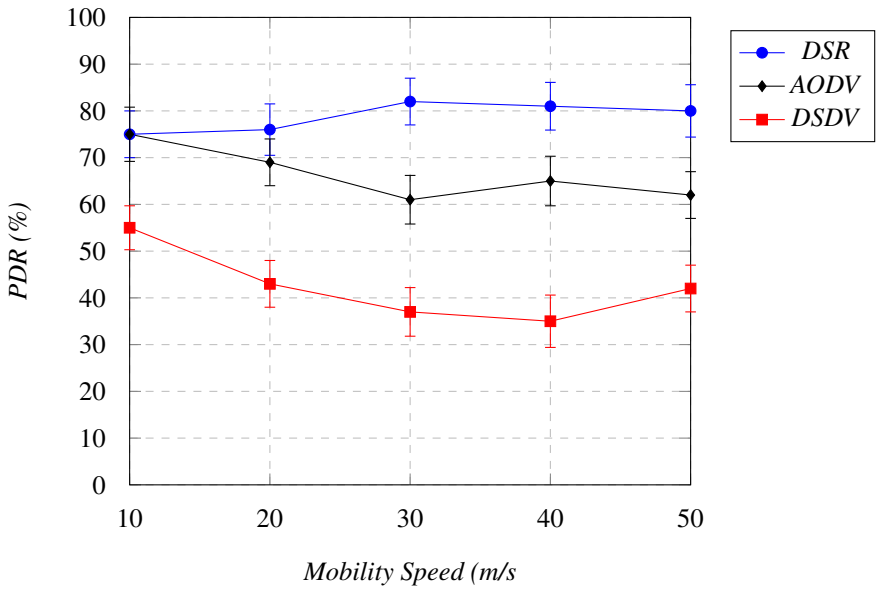

Fig. 3.3 PDR vs. mobility speed (95% CI)

In Figure 3.3, DSR shows better results than AODV and DSDV, with stable performance during the simulation. All three protocols present a smooth profile in terms of the PDR in a manner that is predictable for all speeds. The PDRs of AODV and DSDV decrease slightly with an increase in speed. One of the main reasons for this is that the greater the mobility speed of the nodes, the more link breakages occur due to the rapid pace. The reactive DSR and AODV protocols present better performance than the proactive protocol DSDV when the node mobility speed increases. The performance of DSR is better than AODV overall, except at the beginning of the simulation.

3.5.3 Change in throughput varying the pause time

Figure 3.4 shows how increasing the pause time can affect the throughput for the AODV, DSR and DSDV protocols. The pause time increases from 5 to 50 s, with fixed simulation parameters as shown in Table 3.2.

Table 3.2 Simulation parameters

Parameter	Value	Unit
Simulator	NS-2.35	-
MAC protocol	IEEE 802.11	-
Antenna	omni antenna	-
Mobility model	random waypoint	-
Simulation time	3600	second
Simulation area	800 * 800	metre
Number of nodes	10	-
Packet size	512	byte
Node speed	5 - 50	m/s
Traffic type	UDP	-
Transmission range	250	metre
Routing protocols	AODV, DSR and DSDV	-

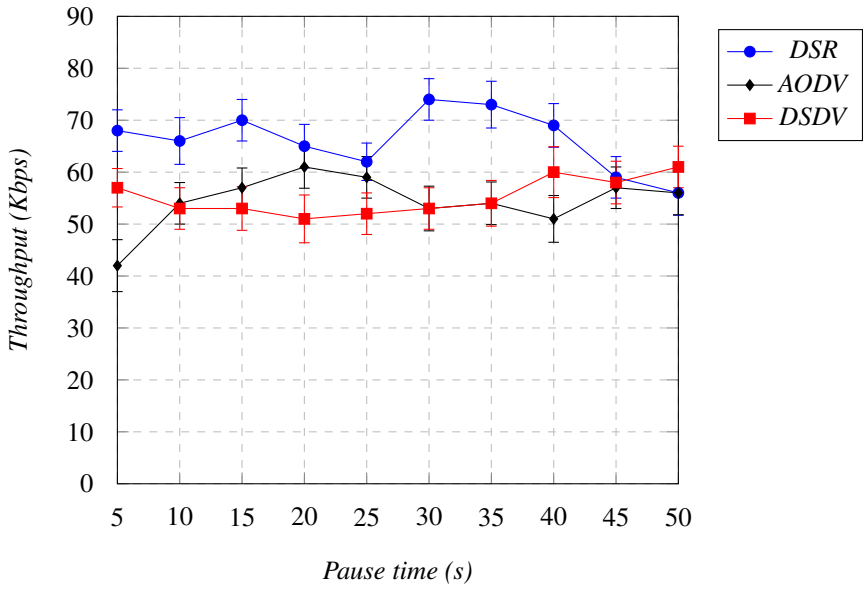

Fig. 3.4 Throughput vs. pause time (95% CI)

Figure 3.4 compares the throughput of the three protocols with an increase in pause time. Having a higher pause time means that the nodes have lower mobility [8]. DSR again shows the best throughput from the beginning to the end of the simulation. We can see that changing the pause time can sometimes increase the throughput and sometimes reduce it. If the nodes pause longer when they are within transmission range of each other, it can aid performance. However, if they pause longer when they are outside each other's transmission range, the effect on throughput will be detrimental. The performance values for AODV and DSDV were close, but DSDV presented better stability.

3.5.4 Change in packet delivery ratio (PDR) varying the pause time

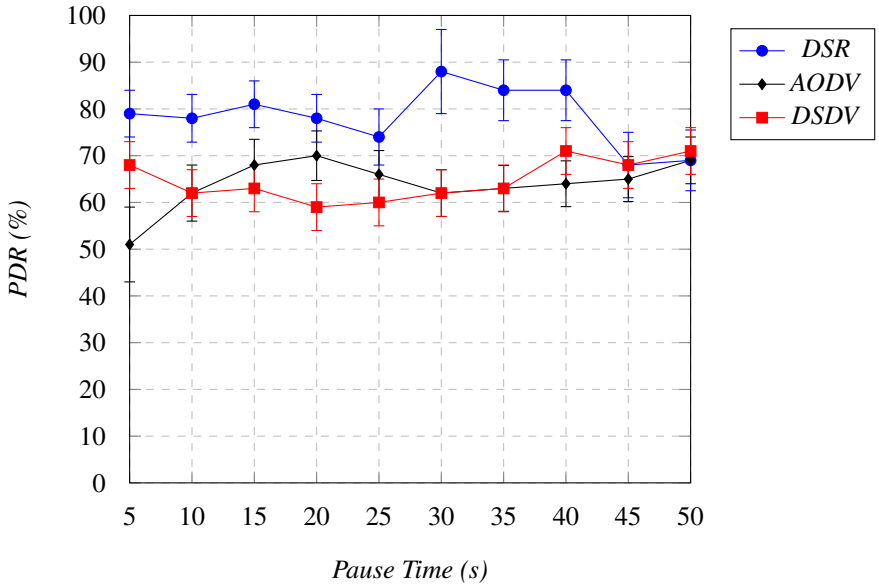

Fig. 3.5 PDR vs. pause time (95% CI)

At the beginning of the simulation, with a pause time of 5 s, there are clear differences between the three protocols; however, at the end of the simulation, with a pause time of 50 s, the PDRs are fairly similar. In general, DSR presents better PDR performance. This could be because it has lower routing overheads and entails less frequent route discovery processes than AODV and DSDV [8].

The simulation was run 10 times and the mean value was taken for each pause time. As the nodes near each other and their movement pauses for longer, the PDR will be high; in contrast, when the source node and destination node are far from each other, there is no available route and movement is paused for a long time, resulting in a very poor PDR.

AODV and DSR present better overall performance than DSDV as the routes are created on demand and this increases the possibility of data delivery as the pause time increases. However, it can also be observed that the PDR sometimes decreases due to the increase in delay and routing load. In AODV and DSR, routes are created on demand, but if nodes are not available to create a route, the route setting time increases, thus increasing the delay in the network. However, DSDV, which is table-driven, presents more stable PDR performance with an increase in pause time as there is less need to update the routing tables. We can conclude that the overall performance of the DSR protocol is better than both the AODV and DSDV protocols; of the latter, DSDV has better stability.

3.5.5 Comparison of the overall performance of AODV, DSR and DSDV protocols

Table 3.3 Comparison of overall performance of AODV, DSR and DSDV

Network Scenario	Metrics	AODV	DSR	DSDV
Increase in node mobility speed	Throughput	Better than DSDV	Best	Worst
	PDR	Better than DSDV	Best	Worst
Increase in pause time	Throughput	Similar to DSDV	Best	Similar to AODV
	PDR	Similar to DSDV	Best	Similar to AODV

Based on the simulation results, the AODV, DSR and DSDV protocols are all accessible and dependable, providing decent levels of performance; however, there are some contrasts in performance in certain cases. With an increase in the speed of nodes, DSR showed better performance over the two metrics, throughput and PDR. With an increase in the pause time, DSR was again better than AODV and DSDV with regard to throughput and PDR.

The reactive AODV and DSR protocols exhibited better performance than the proactive DSDV protocol in all the scenarios. One of the main factors driving this is that AODV and DSR are on-demand protocols in which routes are created whenever a source needs to send data to a destination node, whereas the proactive DSDV protocol is table-driven and each node maintains one or more tables containing routing information to every other node in the network and when there is a change in the network topology, updating throughout the network is necessary.

DSR outperformed AODV and DSDV because it is a reactive routing protocol that requires less overhead and is better suited for high-speed mobility. DSR is designed to operate over a smaller bandwidth and has low latency. In addition, it offers improved reliability and scalability due to its ability to use multiple routing paths [5]. In highly dynamic networks, this makes it less susceptible to packet loss and congestion.

3.6 Summary

This chapter has analysed the effects of mobility on the performance of three routing protocols: AODV, DSR and DSDV. Using simulations, we have evaluated the performance

of the three protocols under different conditions, varying the node mobility speed and pause time parameters. We have measured the effect on the protocols using the throughput and PDR metrics. The simulations demonstrate that node mobility speed has a real effect on MANET performance. Higher speeds can cause more link breakages between the nodes and lead to lower performance. The coming chapters adopt node mobility speed as the main parameter to be varied in evaluating the performance of the proposed schemes against attacks.

Chapter 4

Analysing the Performance of a Direct Trust-Based Protocol Against Black-Hole and Grey-Hole Attacks

4.1 Introduction

The last chapter analysed the effect of mobility on the performance of two reactive protocols, AODV and DSR, and one proactive protocol, DSDV. This provided a clear picture of the effect of node mobility on MANET performance in ideal scenarios, when all nodes are cooperative and behave as expected by the routing protocols in forwarding the packets through the network. Based on those results, the analysis in this chapter adopts node mobility speed as the prime parameter in evaluating the proposed schemes protecting against different types of attack.

This chapter proposes and evaluates two direct trust management schemes that should enhance the performance of MANETs in the presence of black-hole and grey-hole attacks. In the first scheme, which is designed to tackle a black-hole attack, each node counts the number of packets dropped by its neighbouring node. If the number of dropped packets reaches a certain threshold, the node in question will be given a low trust value and isolated. In the second scheme, which is designed to tackle a grey-hole attack, each node calculates the ratio of the dropped packets instead of the number. If the ratio reaches a certain percentage, a low trust value will be given to the malicious node and it will be isolated.

First, the effects of black-hole and grey-hole attacks on the performance of the AODV protocol are evaluated. Then we implement the direct trust management schemes with the AODV protocol, calling the new protocol TAODV. We then compare the performance of the TAODV and AODV protocols under black-hole and grey-hole attacks independently.

The outcome will determine the extent to which the principle of direct trust management can reduce the harm of black-hole and grey-hole attacks on MANETs.

While DSR have shown better performance than AODV in chapter 3 under regular conditions, there are several reasons why we have chosen AODV for measuring the performance of MANETs in the presence of attacks:

1. Real-world relevance: AODV (Ad-hoc On-demand Distance Vector) is one of the more widely used and standardized routing protocols for MANETs.

2. Vulnerability assessment: AODV, like any other routing protocol, has its vulnerabilities, and studying its performance under attack conditions can help identify its weaknesses and potential security flaws.

4.2 Black-hole attack

A black-hole attack is a type of DoS attack that has a detrimental effect on the performance of MANET routing protocols [40, 72]. Such attacks have a serious impact on the PDR and the throughput of the network. In this attack, the malicious node attracts all packets by falsely claiming to have the shortest and most up-to-date route to the destination node and then drops the packets instead of forwarding them, creating a black hole in the network [72]. The black-hole attacker continuously observes the network traffic, replies to any RREQ packets and places itself between the source node and the destination node [73]. According to the AODV mechanism, the source node selects the RREP packet that has the highest destination sequence number (DSN).

The DSN is stored in each entry in the routing table of a node. When an intermediate node receives an RREQ, it checks its routing table to see if it has route to the required destination. If it does, the entry for this route should have a DSN field. The DSN increases incrementally each time the intermediate node receives information about the destination node through a RREQ, RREP or RERR. Thus, the higher the DSN, the more up-to-date the route. Once the intermediate node finds a route to the required destination in its routing table, it replies to the source node with an RREP, which contains information about the route, including the DSN. If the source node receives several RREPs, it will choose the one with the highest DSN. Each entry in the routing table of a node consists of the following fields [74];

- Destination IP Address
- Destination sequence number (DSN)
- Valid DSN flag

- Other routing flags (valid, invalid, repairable, under repair)
- Network interface
- Hop count (number of hops needed to reach the destination)
- Next hop (next hop in the route)
- Lifetime (time the route will be valid until expiration or deletion)

In AODV, the route with the highest sequence number is considered the most up-to-date path to the destination node and is thus selected as the most efficient route [75]. Comparing the sequence numbers of neighbouring nodes determines the most reliable route, i.e. the RREP with the highest sequence number. The malicious node exploits this rule and replies to the source node with an RREP that has a high fake sequence number [40].

Figure 4.1 shows how a black-hole attack works in the network. Node A is the source, Node B is the destination and Node C is the malicious node. In reactive protocols such as AODV, a process of routing discovery will initialise to find a fresh and ready route. Node C replies to Node A as soon as it receives the RREQ packet, claiming it has an existing path through it to Node B. Node A will accept the response from Node C without any validation. Node C will start dropping the packets coming from Node A rather than forwarding them.

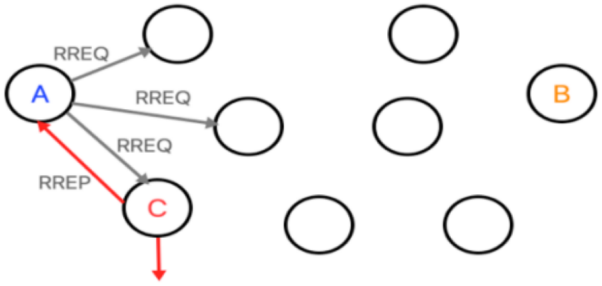

Fig. 4.1 Black-hole attack in MANET

The malicious node in the black-hole attack exploits the vulnerability in the routing discovery phase of reactive protocols such as AODV. The malicious node replies to an RREQ with an RREP that contains a high DSN before any reply from the destination reaches the source. The source will choose the route that goes through the malicious node. The black-hole attacker keeps repeating the same process with other nodes, attracting them to select its route and creating a black hole in the network. The malicious node can then discard the packets or misuse the network traffic.

4.3 AODV vulnerabilities

In terms of black-hole attacks, the malicious node exploits the following vulnerabilities in AODV:

1. The malicious node can modify the data in its routing table. The routing table in each node can be accessed, read and modified by the node itself. The routing tables in AODV are not protected by any mechanisms, such as authentication or encryption. As a result, it is possible for any node to insert fake data in its routing table and send them through RREPs to its neighbours. The fake data here will be a fake route with a very high sequence number.

2. The source node does not have a method for validating the RREPs received. By default, all incoming RREPs are considered trustworthy in AODV and all the data they carry is deemed true.

3. After accepting the RREP and starting to send the data packets to the malicious node, the source node has no mechanism to establish whether the packets are forwarded to the destination node or not.

Thus, the AODV protocol has three major vulnerabilities to black-hole attacks, one related to the malicious node and two related to the victim node. In terms of the malicious node, it can make up fake routes and sequence numbers. For the victim node, there is no means of validating the control packets received or of tracking the packets sent.

4.4 Suggested direct trust management scheme

Since the goal of a black-hole attack is to attract packets from the source nodes and drop them, the key point in designing a defense mechanism is to monitor this specific behaviour. Once we understand the behaviour of the attack and how it works, we can create a scheme to detect it. To detect a black-hole attack using the concept of direct trust management, each node needs to watch the behaviour of the next node in the route to see if it passes the data packets or drops them.

4.4.1 Non-malicious dropping of packets

It would appear that dropping data packets in MANETs is always malicious. However, a cooperative node can drop packets for many reasons. It is crucial to distinguish such cooperative nodes from malicious nodes that drop packets deliberately. A cooperative node can drop packets temporarily due to system overload or interference from other wireless

signals [76]. A node should thus not be judged based on an initial small number of dropped packets; it should have more than one opportunity to forward the packet to the next node. If it fails to do so consistently, its behaviour is likely to be malicious. There is no specific number of opportunities an intermediate node should be given to demonstrate that it is cooperating.

In the trust initialisation step, which is the process by which the source node starts to trust its neighbouring nodes, the source node requires initial evidence enabling it to establish whether or not it should trust its potential neighbouring node. This can be done by sending a specific number of test packets to evaluate the trustworthiness of the neighbouring node. If the neighbouring node drops all these test packets in sequence, it can be judged to be malicious.

Test packets are special packets used in Mobile Ad hoc Networks (MANETs) to assess the performance and trustworthiness of neighboring nodes. These lightweight packets contain minimal data, such as source and destination node IP addresses. By sending and receiving test packets, nodes can evaluate link quality, measure metrics like packet loss rate and signal strength, and discover and verify neighboring nodes. Test packets play a crucial role in dynamic routing, network maintenance, and security evaluation, helping to maintain updated routing tables, optimize path selection, and identify potential misbehavior or malicious activity.

4.4.2 Threshold for determining malicious intent

The number of test packets that should be sent to decide the trustworthiness of a neighbouring node is based on the requirements and mechanism of the protocol. Each routing protocol should have a different mechanism and rules for establishing trust. Also, the exact number of test packets to be sent will likely depend on the application to be run on the network and its performance requirements.

For our analysis, we proposed three fixed thresholds for test packets: 25, 50 and 100. However, setting a fixed threshold may not deliver accurate decisions in all network scenarios. A fixed threshold may work well and lead to the detection of malicious nodes in a network scenario with certain parameters, but result in an inaccurate decision in a different scenario. In our case, using the parameters shown in Table 4.1, setting a fixed threshold of 25 packets blocked some cooperative nodes as it is very possible for a node in a MANET to drop 25 packets in sequence due to a shortage of resources, not due to malicious activity. Some cooperative nodes were thus labelled malicious as a result of inaccurate decisions. With a fixed threshold of 100 packets, we were able to detect malicious nodes accurately, but the overload increased sharply. For each source node to send 100 test packets to its neighbouring node and observe its behaviour before it starts

streaming data to it takes a long time and increases the delay in the network. The middle option, a fixed threshold of 50 test packets, resulted in an acceptable level of overhead while also being able to detect malicious nodes. Figure 4.2 shows the overhead (end-to-end delay) in the three cases.

Table 4.1 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	Ns-2.35	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1000 * 1000	metre
Number of nodes	15	-
Node speed	15-50	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total of simulation runs	10	-
Malicious nodes	1	-
Attack type	black-hole	-
Routing protocol	AODV, TAODV	-

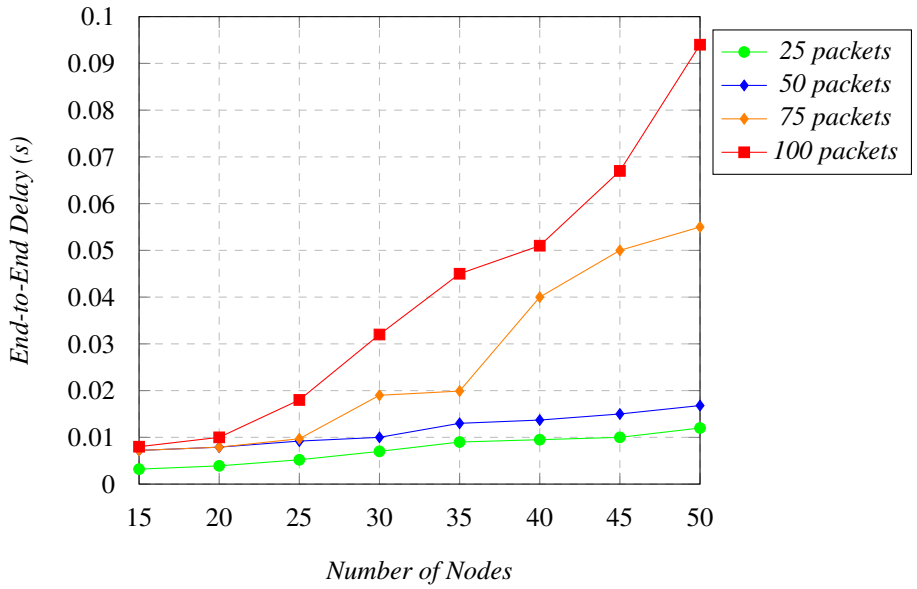

Fig. 4.2 End-to-end delay for thresholds of 25, 50 and 100 packets (95% CI)

Based on these results, we selected 50 test packets as the threshold. This offers the best trade-off between improving the throughput and PDR by isolating a malicious node while keeping the overhead as low as possible.

4.4.3 Implementation of direct trust management scheme

To implement this scheme, each node needs to have a list storing the packets sent to the neighbouring node and a trust table storing the trust values for the neighbouring nodes. In Figure 4.3, Node 1 is the source node, Node 2 is the intermediate node and Node 3 is the destination node.

When Node 1 sends the first packet to Node 2, it stores the same packet in its packet list and increases the number of stored packets by one. The number of stored packets in the packets list is assigned to a variable n. When Node 1 establishes a connection with Node 2 for the first time, the packets list is empty and the trust value of Node 2 is T=*0.5* by default.

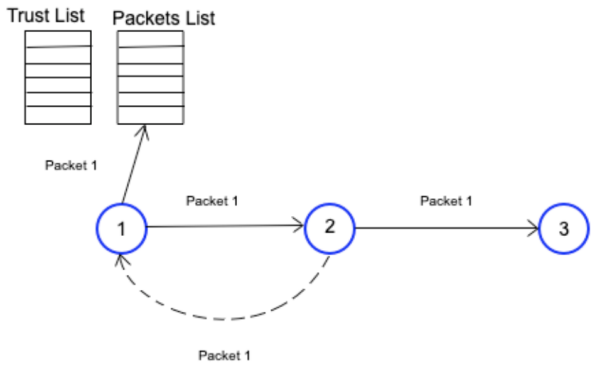

Fig. 4.3 Direct trust management implementation

When Node 2 forwards the packet to Node 3, it needs to send a copy of the packet back to Node 1 to confirm that it has been forwarded to the next node in the route. In this case, Node 1 will delete the packet from the packets list and change the trust value of Node 2 from 0.5 to 1.

However, if Node 2 drops the packet, Node 1 node will not receive a copy of the packet from Node 2. Node 1 will wait for `NET_TRAVERSAL_TIME` then send the next test packet, giving Node 2 another opportunity to show cooperative behaviour. Node 1 will again store the new packet in its packets list and the number of stored packets will be n=*2*. `NET_TRAVERSAL_TIME` is the length of time that a source node will wait to receive an RREP from its neighbouring node before it sends a new RREQ in the AODV protocol [77].

`NET_TRAVERSAL_TIME` in AODV and the TCP timer are similar in context of packet acknowledgments. The similarity here lies in their usage as timeouts for managing network communications in different scenarios.

Node 1 will continue to send packets until the number of stored packets n reaches the maximum value of 50, then it will change the trust value of Node 2 from 0.5 to 0. If Node

2 forwards one of the 50 packets, all the previously dropped packets will be deleted from the packets list stored by Node 1 and the counter *n* will start again from 0.

4.4.4 Flowchart

Figure 4.4. presents a flowchart describing the implementation of the proposed direct trust management scheme for detecting black-hole attacks in AODV. *T* represents the trust value given to the node in question and n represents the number of packets sent.

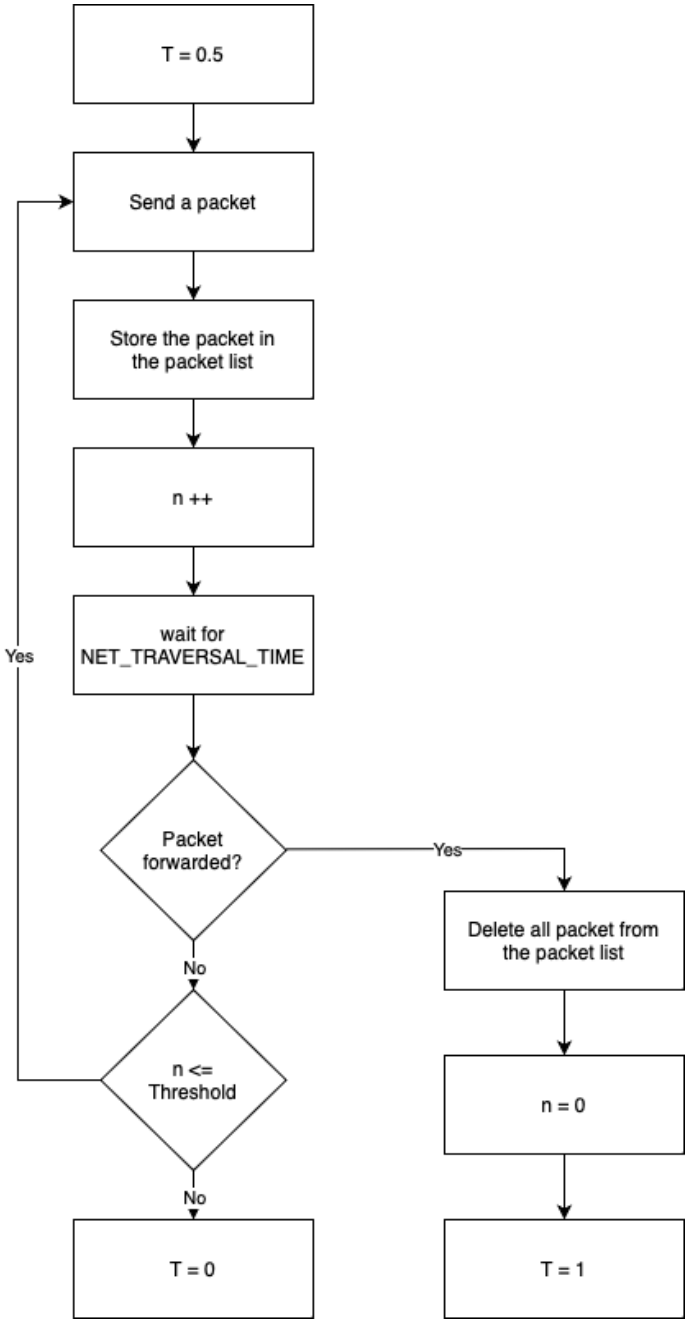

Fig. 4.4 Algorithm for the direct trust management scheme

4.4.5 Trust management scheme algorithm

Algorithm 1 shows the logic of the proposed trust management scheme for detecting black-hole attacks in AODV.

Algorithm 1 Algorithm of proposed approach

n: Number of sent packets

T: Trust value

Th: Threshold

```
1: n = 0
2: while n <= Th do
3:     Send packet to the neighbouring node
4:     n++
5:     if neighbouring node forwards the packet then
6:         n = 0
7:         T = 1
8:         Exit
9:     end if
10: end while
11: T = 0
```

4.4.6 How the malicious node is isolated

1. Trust List Check: Before sending data to any neighbouring node, the source node first checks its Trust List table to determine the trust value of the destination node. If the destination node is not present in the Trust List, the source node will assume that the trust value of this node is 0.5.

2. Avoiding Data Transmission: If the neighbouring node's trust value is zero, the source node avoids sending data to that node directly. Instead, it looks for an alternative route to deliver the data.

3. Selecting an Alternative Route: When the source node discovers that the neighbouring node is malicious, it triggers the route discovery process in AODV to find a new route to the destination.

4.5 Simulation and evaluation

AODV is an open-source routing protocol, which means that its source code can freely be accessed and modified. We modified the existing AODV protocol by adding the proposed

direct trust management scheme and called the new version of the protocol TAODV. We used the NS-2.35 network simulator to develop the TAODV protocol and evaluate its performance against black-hole attack.

4.5.1 Added scheme and functions

The trust scheme added to the AODV protocol contained six functions: (i) insert, which inserts a new trust value; (ii) lookup, which searches for the trust value of a target node; (iii) delete, which deletes a trust value; (iv) update, which updates a trust value; (v) flush, which deletes all stored packets from the packets list; (vi) count, which returns the total number of stored packets in the packets list. We also added a timer variable to store the time that the source node should wait until sending the next test packet. Rather than using a fixed amount of time, we used `NET_TRAVERSAL_TIME`, which is already calculated in the AODV protocol. If this specific time passes without the source node receiving a copy of the packet back from the intermediate node indicating that it has been forwarded, the packet has been dropped by the intermediate node.

4.5.2 Metrics and parameters

To evaluate the performance of TAODV, we here compare its performance to that of the plain AODV protocol in the presence of a black-hole attack. We use the throughput and PDR as metrics to measure performance. The parameters used in the simulation are shown in Table 4.2.

Table 4.2 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	Ns-2.35	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1000 * 1000	metre
Number of nodes	15	-
Node speed	10-50	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total no. of simulation runs	10	-
Malicious nodes	1	-
Attack type	black-hole	-
Routing protocol	AODV, TAODV	-

4.5.3 Throughput performance varying the speed of node mobility

Figure. 4.5 shows the throughput of AODV and TAODV when both are under black-hole attack and with increasing node mobility speed.

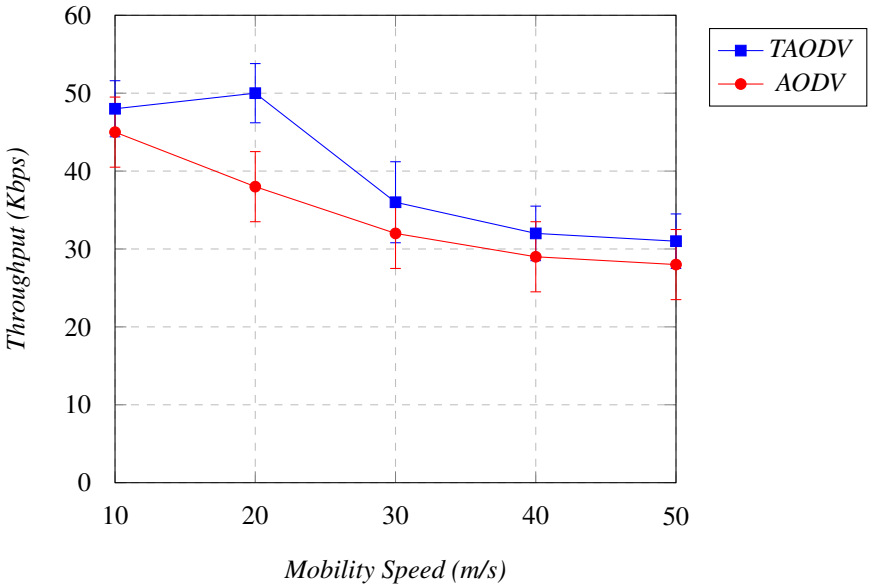

Fig. 4.5 Throughput of TAODV vs. AODV protocols under black-hole attack (95% CI)

Figure 4.5 shows that the performance of TAODV in terms of throughput started higher than AODV and it maintained better performance until the end of the simulation. This means the proposed direct trust management scheme detected and isolated the malicious node. Hence, TAODV was the best protocol in this scenario.

Figure 4.5 further shows that increasing the node mobility speed leads to reduced network throughput for both the TAODV and AODV protocols. This is due to the fact that higher speeds lead to more link breakages. When a link breaks, the reactive routing protocol will trigger maintenance phase processes. It will then take some time until a new route is established and some packets will be lost [5].

4.5.4 PDR performance varying the speed of node mobility

As for throughput, Figure 4.6 shows better performance for TAODV than AODV in terms of the PDR during the simulation at all node mobility speeds.

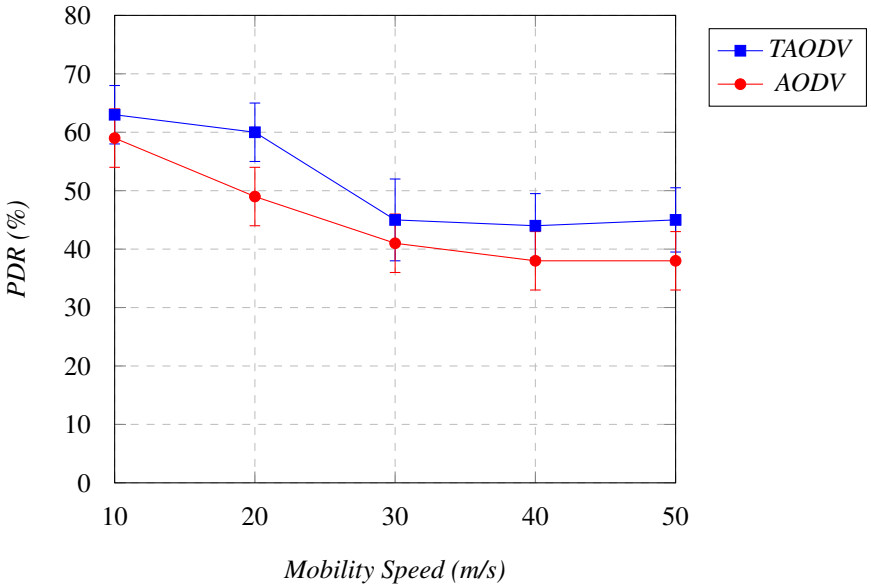

Fig. 4.6 PDR of TAODV vs. AODV protocols under black-hole attack (95% CI)

4.5.5 Throughput performance varying the number of malicious nodes

Figure 4.7 shows the results of evaluating the TAODV and AODV protocols with an increase in the number of malicious nodes while maintaining a constant node mobility speed of 30 m/s.

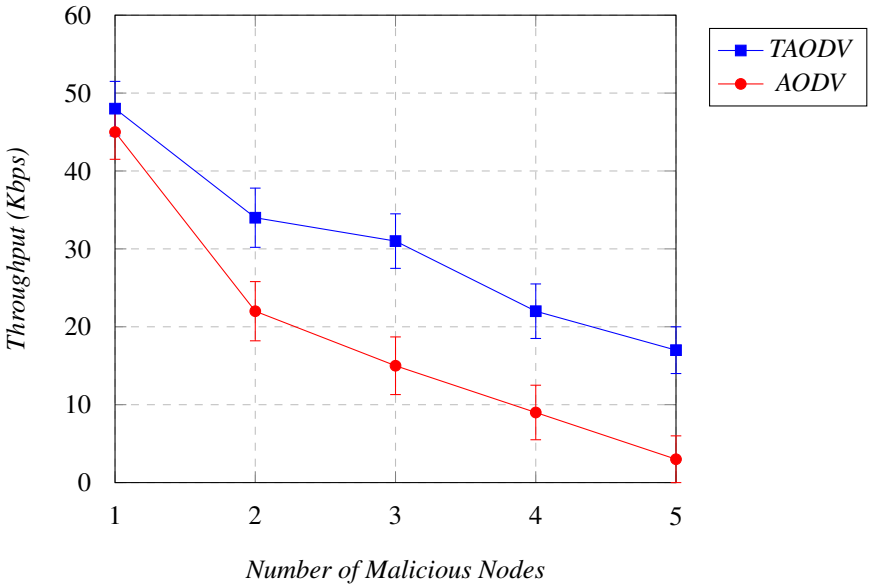

Fig. 4.7 Throughput of TAODV vs. AODV protocols under black-hole attack (95% CI)

Figure 4.7 shows that increasing the number of malicious nodes launching black-hole attacks at the same time affects the throughput of both the TAODV and AODV protocols. However, TAODV performs better than AODV from the beginning to the end of the simulation.

At the end of the simulation, when there were 5 malicious nodes, the throughput of AODV was very close to zero, almost reaching denial of service. In TAODV, with 5 malicious nodes the throughout was around 18 Kbps, indicating the network was affected but still running.

4.5.6 PDR performance varying the number of malicious nodes

Figure 4.8 shows that the PDR for both the TAODV and AODV protocols decreases as the number of malicious nodes increases. The PDR of TAODV decreased from 63% to 28% when the number of malicious nodes increased from 1 to 5 compared to a decrease in AODV from 59% to 12%.

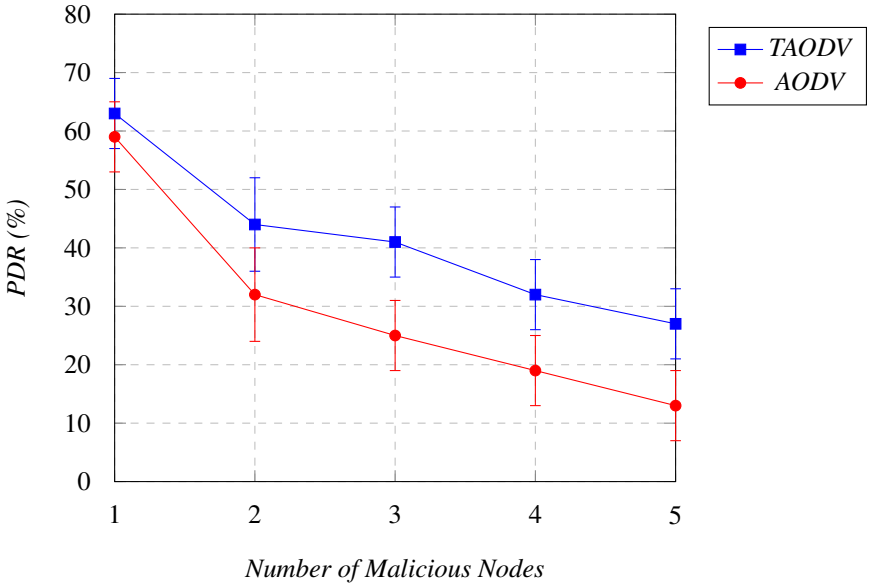

Fig. 4.8 PDR of TAODV vs. AODV protocols under black-hole attack (95% CI)

4.5.7 End-to-end delay performance varying the number of nodes

Figure 4.9 illustrates the end-to-end delay observed in both the AODV and TAODV protocols under conditions devoid of any attack. In the context of TAODV, a slight elevation in the end-to-end delay becomes evident as the number of nodes increases. This phenomenon can be attributed to the mechanism's inherent design, whereby the source node promptly identifies the trustworthiness of its neighbor upon successful forwarding

of the initial test packet. Consequently, subsequent test packets are preemptively skipped, negating the necessity for the source node to complete the transmission of the entire set of 50 test packets before making a decision. This strategic approach serves to maintain a reasonable level of end-to-end delay, aligning with the scheme's intent.

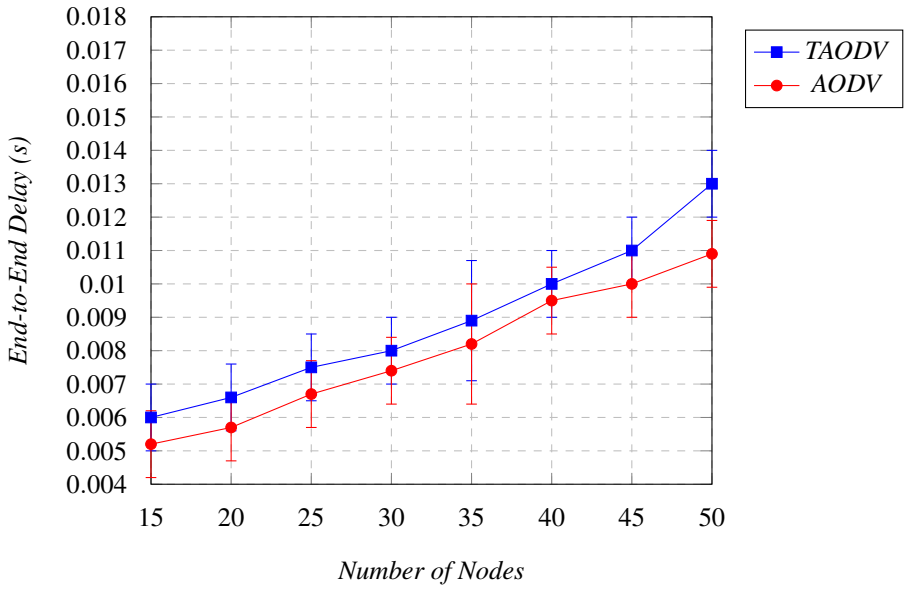

Fig. 4.9 End-to-End Delay in TAODV vs. AODV (95% CI)

4.6 Grey-hole attack

A grey-hole attack is an enhanced version of the black-hole attack in which the malicious node's activities are not predictable [38]. The node that is launching a grey-hole attack functions as a cooperative node for some time during the route discovery process in forwarding the packets, but then at some point turns malicious [73]. Its ability to change state makes it especially difficult to detect [38].

In MANETS, a grey-hole attack is a model of malicious attack in which an attacker selectively forwards traffic to harm the network performance [78]. Grey-hole attacks differ from black-hole attacks in that they selectively forward and drop packets. Black-hole attacks are much easier to detect and isolate. Typically, more complex mechanisms are required for routing protocols to detect grey-hole attacks. The characteristics of grey-hole attacks include selectively forwarding or dropping packets, reducing network services and manipulating network packets [78].

Figure 4.10 shows how a grey-hole attack works, where Node A is the source, Node B is the destination and Node C is the malicious node.

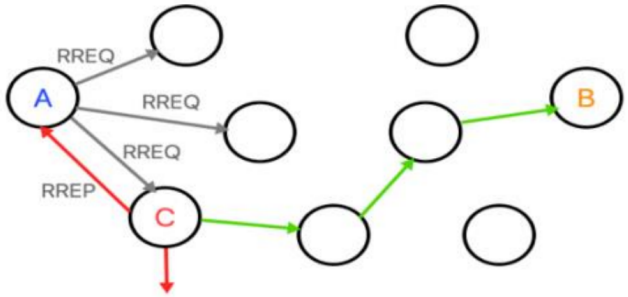

Fig. 4.10 Grey-hole attack in MANET

In a grey-hole attack, the first step the malicious node takes is to capture the route by exploiting the vulnerabilities of the route discovery phase in the AODV protocol. When the malicious node has accessed the route successfully without engaging in any malicious activities, it starts dropping the incoming packets with a certain probability [79]. In grey-hole attacks, the malicious node may cooperate for a certain period of time and drop packets the rest of the time. Also, it may drop packets received from a specific node and forward all packets from another node. Because of this unpredictable deceptive behaviour, detecting a grey-hole attack is more difficult than detecting a black-hole attack.

In this analysis, we apply the same trust management scheme as for black-hole attacks, detecting grey-hole attacks and seeing how they work. Table 4.3 shows the parameters used in the simulation.

Table 4.3 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	Ns-2.35	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1000 * 1000	metre
Number of nodes	15	-
Node speed	10-50	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total no. of simulation runs	10	-
Malicious nodes	1	-
Attack type	grey-hole	-
Routing protocol	AODV, TAODV	-

4.6.1 Throughput performance varying the speed of node mobility

We ran AODV and TAODV in the presence of a grey-hole attack, increasing the node mobility speed from 10 m/s to 50 m/s. Figure 4.11 shows that the TAODV and AODV throughput are almost identical under a grey-hole attack. This indicates that the suggested scheme is not very effective in detecting a grey-hole attack and isolating the malicious node.

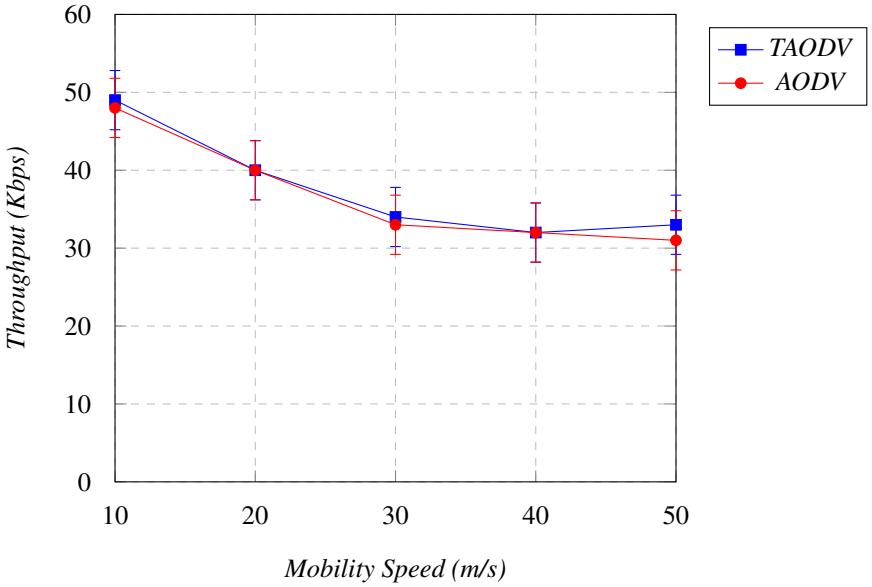

Fig. 4.11 Throughput of TAODV vs. AODV protocols under grey-hole attack (95% CI)

4.6.2 PDR performance varying the speed of node mobility

In general, the PDR and throughput are connected, so they increase and decrease in tandem. They are interdependent because if the throughput, i.e. the data rate, decreases, the number of packets being transmitted successfully decreases, and thus the PDR also decreases. Likewise, any increase in throughput will lead to an increase in the PDR.

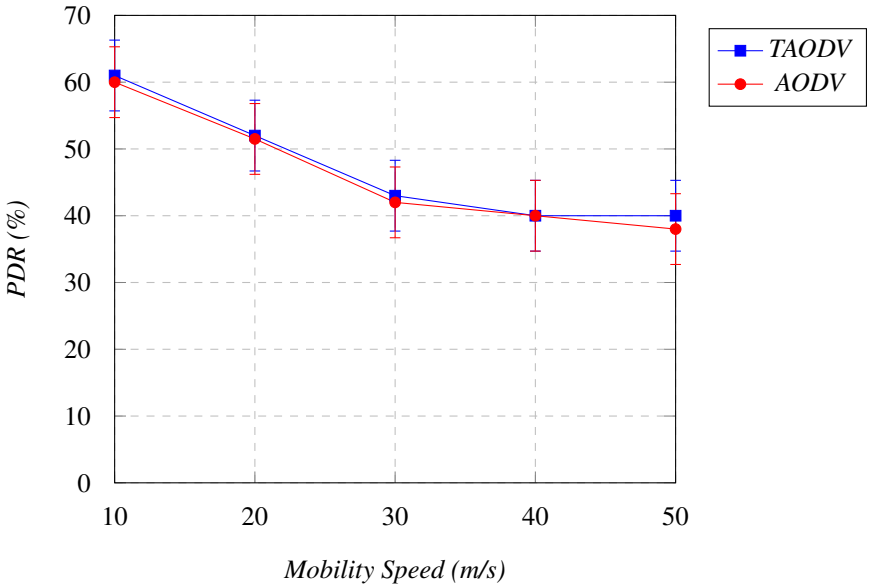

Fig. 4.12 PDR of TAODV vs. AODV protocols under grey-hole attack (95% CI)

Figure 4.11 shows that the PDR of the TAODV and AODV protocols are almost identical under grey-hole attack for the same reason as in the case of throughput outlined above.

4.6.3 Why a grey-hole attack is distinctive

The direct trust management method proposed calculates the trust value by counting dropped packets. If 50 packets are dropped by the neighbouring node, it will be considered malicious activity and the neighbouring node will be isolated. This method showed good results in detecting black-hole attacks because the black-hole attacker drops all incoming packets and does not forward any. The situation is different in the case of a grey-hole attack as the grey-hole attacker passes on some packets and drops others, which makes this method of establishing a trust value less effective.

In the proposed scheme, the counter n starts counting the dropped packets and if a single packet is passed on, the counter resets and starts from zero. This means the grey-hole attacker can drop 49 packets without being detected and if it forwards packet number 50, the counter will reset. The malicious node can then drop another 49 packets without being detected. This loop could continue for very long time, depending on how many packets the grey-hole attacker intends to drop each time.

The grey-hole attack model in this simulation has the malicious node dropping a random number of packets from 1 to 100, then forwarding a random number and so on. This is why we observe a very small improvement for the TAODV protocol in this simulation. If the malicious node drops 50 packets or more sequentially, it will be detected

and isolated and thus the throughput will increase. There is a possibility that the malicious node could drop more than 50 packets and be detected, but this chance is quite low. In this case, the malicious node will likely not be detected or isolated under the proposed mechanism and it will keep launching the attack and harming the network performance.

4.6.4 Calculating trust values using the dropped packets ratio

From the results of the last two simulations, it is apparent that detecting a grey-hole attack based on the number of dropped packets does not always work. It works only if the malicious node drops 50 packets or more in sequence. A possible solution would be to reduce the number of test packets, for example to 25 instead of 50. This would increase the possibility of detecting a grey-hole attack, but it would still be possible for the malicious node to drop 24 or fewer packets without being detected. Were we to go further and reduce the number to 10 packets, for example, there would be a risk of considering cooperative nodes malicious and isolating them. This is due to the reduction in the tolerance of an intermediate node dropping some packets for reasons that are not malicious, such as a shortage of resources. Such a strict mechanism might not be suitable for MANET environments, in which nodes with different resources and abilities are supposed to cooperate.

Another possible solution would be to consider the ratio of dropped packets rather than the number. We can amend the mechanism to calculate the ratio of dropped packets and base the decision of whether to consider a neighbouring node malicious or not on this metric. After sending 50 test packets, the ratio of dropped packets will be calculated. If it is greater than 20%, it will be considered a grey-hole attack.

Throughput performance varying the speed of node mobility

Figure 4.13 shows that the TAODV protocol performs better based on the ratio of dropped packets when the network is under a grey-hole attack. However, comparing the throughput of TOADV with that of AODV, the difference is marginal. This means it is still possible for the grey-hole attack to remain undetected when dropping less than 20% of the 50 test packets each time.

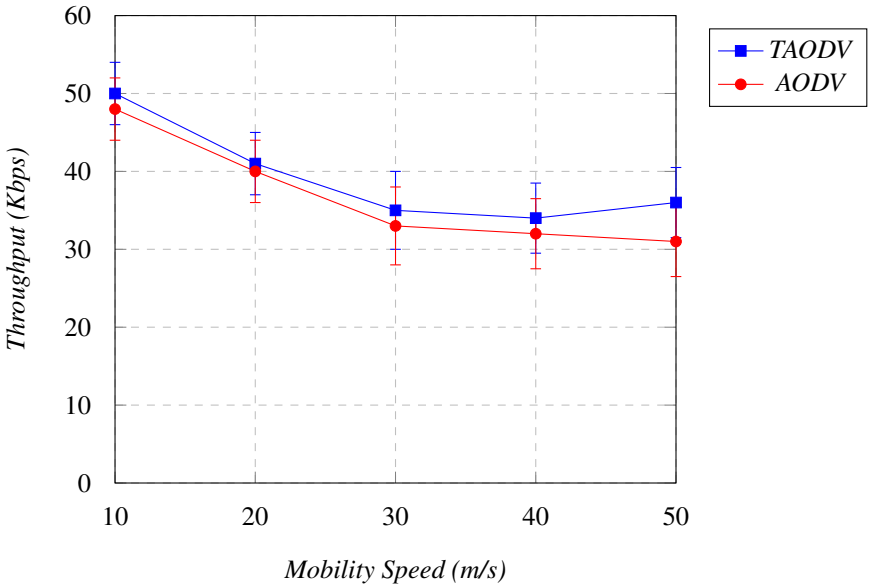

Fig. 4.13 Throughput of TAODV vs. AODV under grey-hole attack (95% CI)

PDR performance varying the speed of node mobility

As for throughput, the PDR of the TAODV protocol increased slightly with the new amendment to the mechanism.

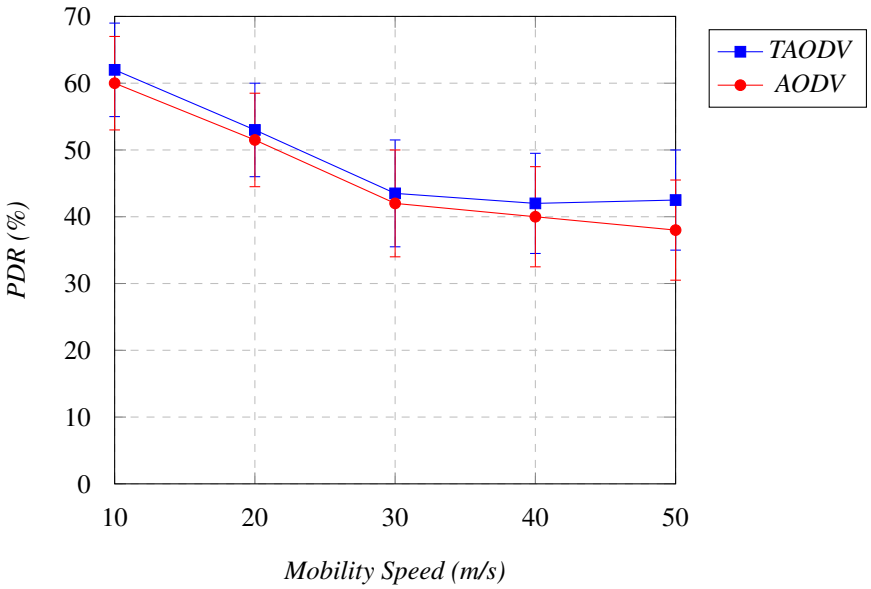

Fig. 4.14 PDR of TAODV vs. AODV under grey-hole attack (95% CI)

Throughput performance varying the number of malicious nodes

We ran both the AODV and TAODV protocols in the presence of a grey-hole attack, increasing the number of malicous nodes from 1 to 5 while keeping node mobility constant at 10 m/s. Table 4.4 shows the parameters used in the simulation.

Table 4.4 Simulation parameters

Parameter	Value	Unit
Simulator	Ns-2.35	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1000 * 1000	metre
Number of nodes	15	-
Node speed	10	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total no. of simulation runs	10	-
Malicious nodes	1 - 5	-
Attack type	grey-hole	-
Routing protocol	AODV, TAODV	-

Figure 4.15 shows that the throughput of both TAODV and AODV decreased with an increase in the number of malicious nodes from 1 to 5. The TAODV protocol performed better than the AODV protocol from the beginning to the end of the simulation, but was very close to the AODV performance.

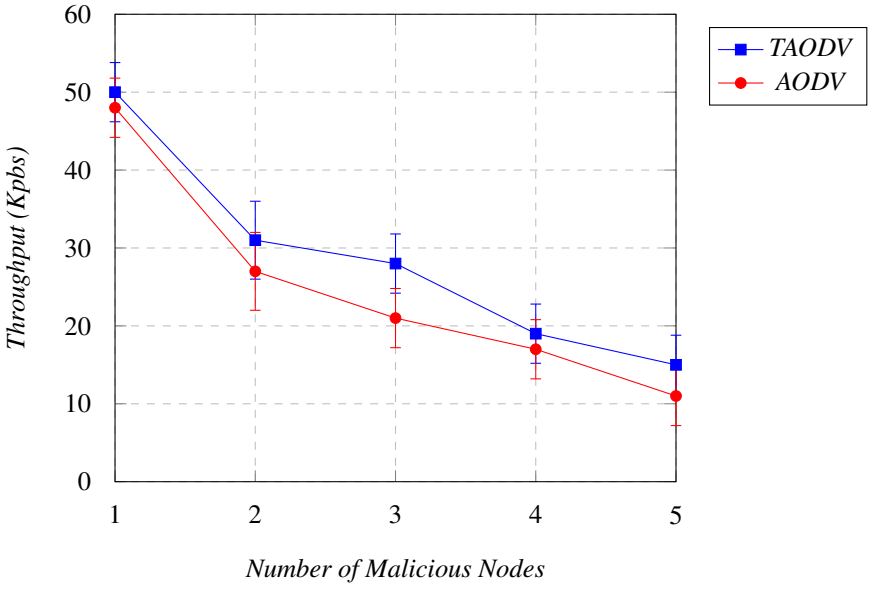

Fig. 4.15 Throughput of TAODV vs. AODV under grey-hole attack (95% CI)

PDR performance varying the number of malicious nodes

As for throughput, the PDR decreased with an increase in the number of malicious nodes, as shown in Figure 4.16. At the end of the simulation, the PDRs were 18% for AODV and 22% for TAODV. Thus, TAODV showed better PDR than AODV in the presence of a grey-hole attack, but both experienced harmful effects.

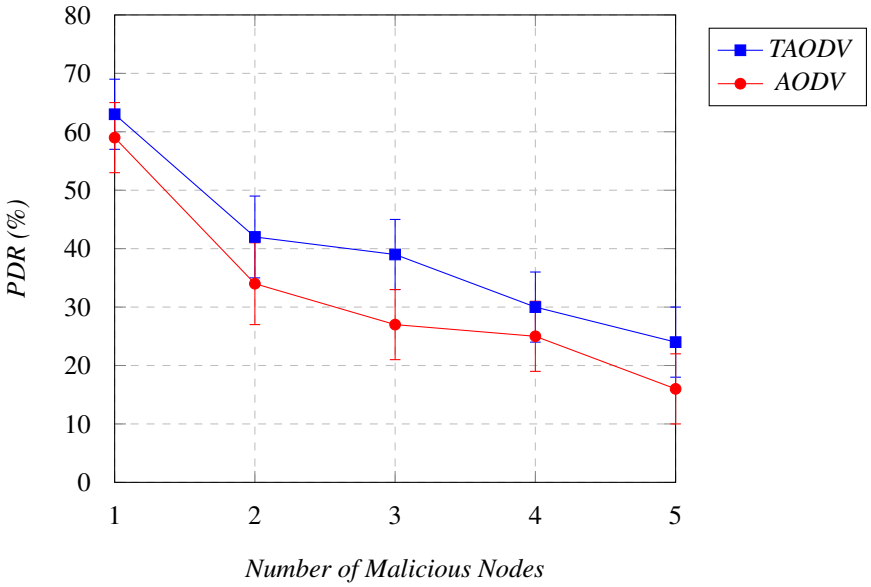

Fig. 4.16 PDR of TAODV vs. AODV under grey-hole attack (95% CI)

4.7 Effect of trust management scheme on overhead in AODV

When designing or implementing a network solution, it is very important to take the overhead into account. Overhead in networking is defined as the amount of transmitting time or processing used by additional mechanisms to manage the transfer of data over the network [5]. Overhead can also be defined as the additional resources, such as bandwidth, delay time and memory, needed to implement the additional scheme in the routing protocol [80]. These extra resources are mainly used to maintain routing between the nodes in MANETS to ensure they receive and send packets efficiently. Consuming more resources can reduce the performance of networks.

The mechanism added to the AODV protocol to reduce the effects of black-hole and grey-hole attacks results in extra work for the protocol. Although this extra work is not needed when there is no attack, it must still be undertaken in all cases. Hence, this overhead needs to be measured to determine the cost of the improvement to the network performance when under attack.

4.7.1 Overhead metrics

Overhead can be measured using many metrics, such as the number of hops, the PDR or end-to-end delay. Selecting the metric to be used for measuring overhead in MANETs depends on the application that will run and use the routing protocol. There is no one metric that can be used for all applications because different applications have different objectives. For example, if a network is used to run a streaming video application, the throughput is likely to be the primary metric used to measure the overhead. If the network is used to run a voice call application, the primary metric of interest might be the end-to-end delay. Thus, different applications need to apply different metrics to measure the overhead accurately.

As the analysis in this thesis does not focus on a specific application, any of these metrics can be used to measure the overhead. We use the end-to-end delay, varying the number of nodes. It is known that increasing the number of nodes tends to increase the number of hops and thus increases the delay time. In AODV, or in any MANET routing protocol, the delay time should increase when the number of nodes increases. First, we determine the end-to-end delay in AODV and TAODV under normal conditions with increasing numbers of nodes and compare the results. This will indicate how much load is added to the AODV protocol with the implementation of the direct trust management scheme. Table 4.5 shows the parameters used in the simulation to measure the overhead.

Table 4.5 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	Ns-2.35	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1000 * 1000	metre
Number of nodes	15 - 50	-
Node speed	10	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total no. of simulation runs	10	-
Routing protocol	AODV, TAODV	-

Figure 4.17 shows that a packet takes longer to travel from the source node to the destination node in TAODV than in AODV. This extra end-to-end delay is the overhead caused by adding the direct trust management scheme to the AODV protocol.

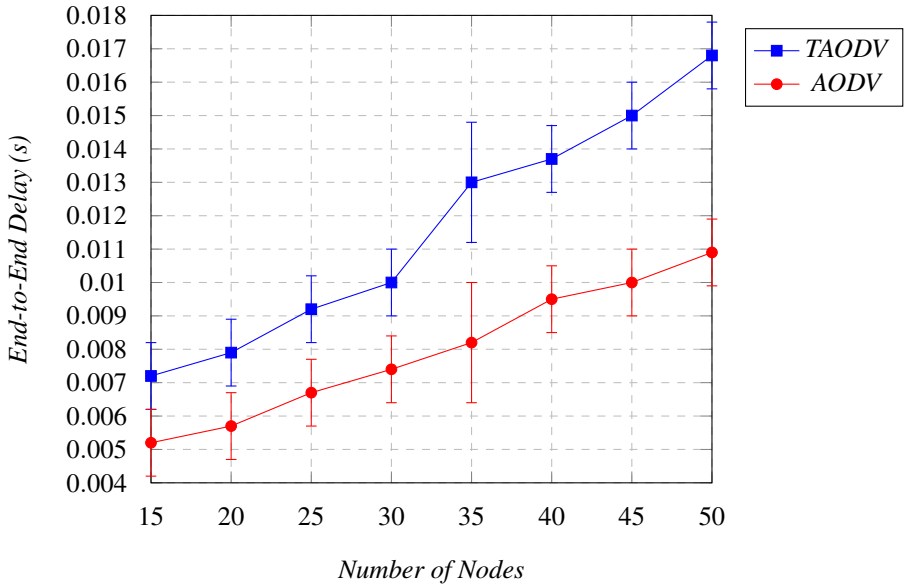

Fig. 4.17 End-to-End Delay in TAODV vs. AODV (95% CI)

At the beginning of the simulation, with 15 nodes, the end-to-end delay in TAODV was 0.002 s more than in AODV. Increasing the number of nodes, the difference in the end-to-end delay value between TAODV and AODV began to rise rapidly. At the end of the simulation, when the nodes numbered 50, the additional end-to-end delay was around 0.006 s. It is clear that any increase in the end-to-end delay will reduce the network performance, but the additional delay in TAODV is still very low and should be acceptable for many applications.

4.8 Summary

This chapter has reported on the implementation of the proposed direct trust management scheme in the AODV routing protocol, resulting in a trust-based protocol called TAODV. We evaluated the performance of the TAODV protocol in the presence of black-hole and grey-hole attacks. The simulations showed that TAODV performed better than AODV in the presence of a black-hole attack. However, in the case of grey-hole attacks, the scheme required slight modification as the calculation of the trust value in the presence of a black-hole attack did not work for a grey-hole attack. We had to calculate the trust value using the ratio of dropped packets instead of the number of dropped packets to be able to detect grey-hole attacks.

From the simulations we can say that the black-hole attack is more harmful but much easier to detect, whereas the grey-hole is less harmful but harder to detect. The grey-hole

attack becomes more dangerous when the malicious node continues launching the attack for long time.

The results of the simulations demonstrate that the direct trust management scheme was able to enhance the throughput and PDR of MANET to some extent in the presence of black-hole and grey-hole attacks. However, the scheme also increased the overhead on AODV slightly. We measured the overhead using the end-to-end delay metric.

In the next chapter, we use the same principle of direct trust management to improve MANET performance in the presence of selfish and flooding attacks.

Chapter 5

Analysing the Performance of Direct Trust-Based Protocols Under Selfish and Flooding Attacks

5.1 Introduction

In Chapter 4, we proposed, implemented and evaluated a direct trust management scheme against black-hole and grey-hole attacks. The scheme showed considerable improvement in network performance against black-hole attacks. However, it was unable to detect the grey-hole attacks effectively. With a small amendment to the calculation of the trust value, its performance improved slightly and it showed better results in the presence of grey-hole attacks.

In this chapter, we evaluate the same scheme used to detect black-hole attacks against selfish attacks and propose a new direct trust management scheme designed to enhance performance in the presence of flooding attacks. Flooding attacks differ entirely from the other three types of attack, so require a different trust management scheme.

In the evaluation, we use the same technique as in Chapter 4, namely comparing the performance of the plain AODV protocol with that of the TAODV protocol under attack.

5.2 Selfish attack

A selfish node is one that does not forward data packets sent by other nodes but instead sends its own traffic through the network [81]. It behaves in this way to maintain its own resources (e.g. CPU cycles, memory and battery) while exploiting those of other networks [82]. It participates in the routing discovery phase, but drops data packets instead of forwarding them to the next node in the route.

The strategy of this attack is similar to the grey-hole attack. The only difference is that a grey-hole attack is intended as an attack and its goal is to harm network performance, whereas the selfish node's goal is to conserve resources [82]. Technically, in a grey-hole attack, the malicious node drops some data packets and forwards others to make it harder to detect the malicious activity, while the selfish node drops all data packets, which makes it more detectable using the direct trust management principle.

The selfish node participates normally in the route discovery and route maintenance phases to deliver its own data, as does any cooperative node. However, if the data packets belong to another node, it drops them instead of passing them to the next node. The selfish node is motivated not to forward packets to conserve its own resources [83]. Unlike black-hole and grey-hole attacks, it does not use any misleading techniques, such as generating RREPs with a very high fake DSN to grab data traffic in the network. It behaves normally and cooperates well during the data transmission phase if the packets are in its own interests, but turns malicious when the data packets are not in its own interests.

Sankareswary et al.[84] detailed the potential activities of a selfish node as follows:

1. Turning off its battery when it does not have a communication of its own to share with another node.
2. Failing to broadcast an RREQ to all its neighbouring nodes on receipt.
3. Failing to pass the RREP to the source node.
4. Broadcasting the RREQ and passing the RREP back to the source node, but dropping the data packets instead of forwarding them.
5. Failing to broadcast an RERR packet when the link to the destination breaks.
6. Turning into a grey-hole node, which passes some packets and drops others to fight against detection mechanisms.

Figure 6.1 shows how a selfish node may participate in creating routes in MANETs but drop the data packets instead of passing them to the next node along the route.

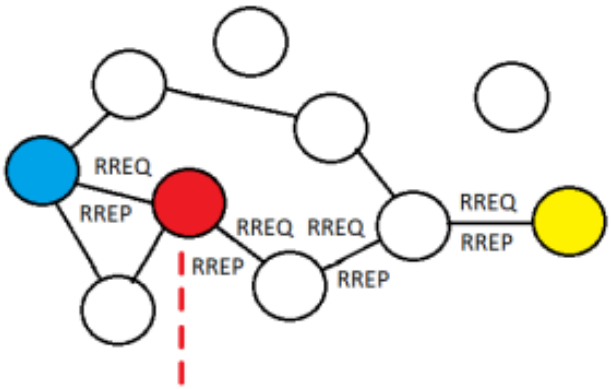

Fig. 5.1 Selfish attack in MANET

5.2.1 AODV vulnerability to selfish attack

When a selfish node is involved in a route, there will be no data streaming from the source node to the destination node [16]. The route becomes meaningless, but the AODV protocol will not consider this route broken because the selfish node will not tell the source node that the route is not valid by sending an RERR. The source will keep sending the data packets and the selfish node will keep dropping them. In contrast to when a route breaks because a node has moved outside the transmission range, AODV will trigger the maintenance phase once the source node receives an RERR packet. The AODV protocol does not trigger the routing maintenance phase in the case of a selfish attack as it keeps the route valid, but causes very low performance in the network. The selfish attack is not detectable in either the route discovery phase or the route maintenance phase in AODV. We can illustrate the vulnerabilities of AODV to selfish attacks as follows:

1. In AODV, when a source node or an intermediate node forwards a packet to a neighbouring node, it is not possible to establish whether that packet has been passed on or dropped. This is the main issue that malicious nodes exploit in AODV to behave in a selfish manner.

2. The default AODV protocol has no mechanism for validating a node's cooperation, for example by counting how many data packets the neighbouring node has forwarded or by setting a minimum number of data packets a node must forward.

Thus, a selfish node can participate in a route and stay there, consuming the network's resources and providing very low cooperation without being detected. A selfish node in both AODV phases will look like any cooperative node and will be allowed to participate in creating routes.

5.2.2 Proposed approach to enhance MANET performance under selfish attack

Since a selfish node drops all incoming data packets, one way of detecting it would be to send test packets and observe the behaviour of the neighbouring node. There is no need to calculate the percentage of dropped packets as in the case of grey-hole attacks. Here, we use the same scheme as for black-hole attacks, i.e. sending a number of test packets to the target node. If all are dropped, the target node is malicious.

After receiving the RREP packet, the source node will send the test packets and check if the intermediate node that has sent the RREP forwards them or not. If none of them are forwarded, a zero trust value will be assigned to this node in the trust table of the source node. Any RREP received from this node in the future will be ignored.

The scheme is implemented by modifying the existing AODV protocol. A trust table is added to each node, which contains a list of the neighbouring nodes and their trust values. The values are based on monitoring the behaviour of the neighbouring nodes.

5.2.3 Flowchart

Figure 5.2. presents a flowchart describing the implementation of the proposed direct trust management scheme for detecting selfish attacks in AODV. T represents the trust value given to the node in question and n represents the number of packets sent.

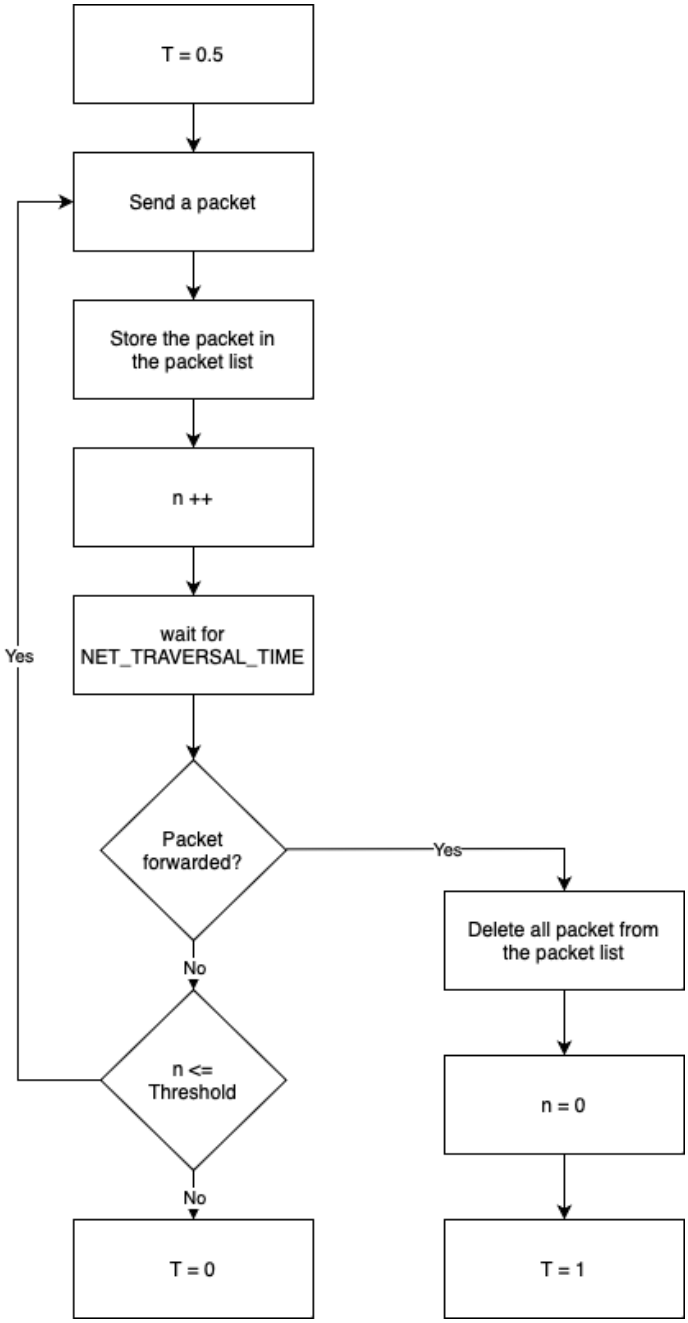

Fig. 5.2 Algorithm for the direct trust management scheme

5.2.4 Trust management scheme algorithm

Algorithm 2 shows the logic of the proposed trust management scheme for detecting selfish attacks in AODV.

Algorithm 2 Algorithm of proposed approach

n: Number of sent packets

T: Trust value

Th: Threshold

```
1: n = 0
2: while n <= Th do
3:     Send packet to the neighbouring node
4:     n++
5:     if neighbouring node forwards the packet then
6:         n = 0
7:         T = 1
8:         Exit
9:     end if
10: end while
11: T = 0
```

5.2.5 Simulation and evaluation

As previously described, we implemented the scheme by modifying the AODV protocol using the NS3.33 network simulator and called the new protocol TAODV. The performance of TAODV is here measured using three metrics: throughput, PDR and end-to-end delay. We compare the performance of the AODV and TAODV protocols under selfish attack. The parameters used to run the simulation are shown in Table 5.1.

Table 5.1 Parameters used to evaluate the performance of TAODV under selfish attack

Parameter	**Value**	**Unit**
Simulator	NS-3.33	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1200 * 1200	metre
Number of nodes	50, 60, ..., 200	-
Node speed	1	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total no. of simulation runs	10	-
Malicious nodes	5	-
Attack type	selfish	-
Routing protocol	AODV, TAODV	-

5.2.6 Throughput performance varying the number of nodes

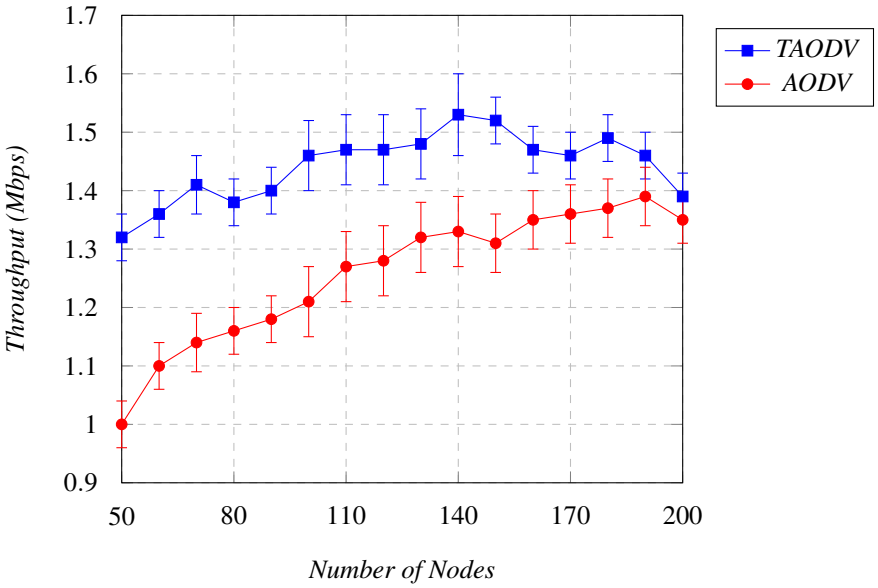

Fig. 5.3 Throughput of TAODV vs. AODV under selfish attack (95% CI)

Figure 5.2 shows that with an increase in the number of nodes in the network from 50 to 200, TAODV provides better throughput than AODV. TAODV performed better from the beginning of the simulation, with 1.32 Mbps of throughput compared to around 1 Mbps for AODV. TAODV maintained better throughput throughout the simulation.

5.2.7 PDR performance varying the number of nodes

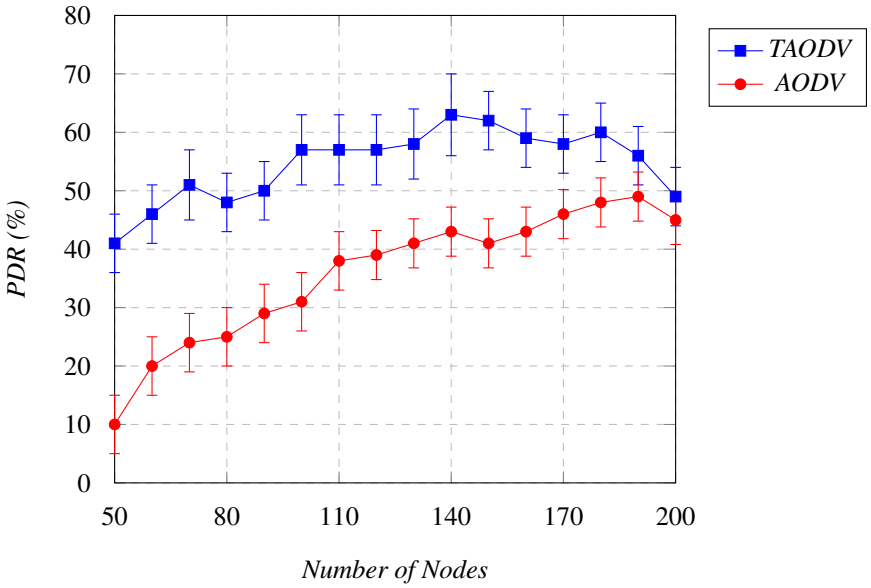

Fig. 5.4 PDR of TAODV vs. AODV under selfish attack (95% CI)

Figure 5.3 shows that TAODV provides better performance than AODV in terms of PDR under selfish attack. Thus, the direct trust management scheme was able to detect the selfish nodes and isolate them. Increasing the number of nodes increases the throughput and PDR in both the AODV and TAODV protocols because the source nodes are likely to have more routes from which to select to reach the destination nodes. Increasing the number of nodes will reduce the probability of a selfish node being selected by the source node.

5.2.8 Throughput performance varying the node mobility speed

In the following two simulations, we compare the throughput and PDR of TAODV and AODV, varying the node mobility speed while maintaining a constant number of nodes equal to 100. Figure 5.4 shows that both the TAODV and AODV protocles were affected by increases in the node mobililty speed, but TAODV performed better throughout the simulation.

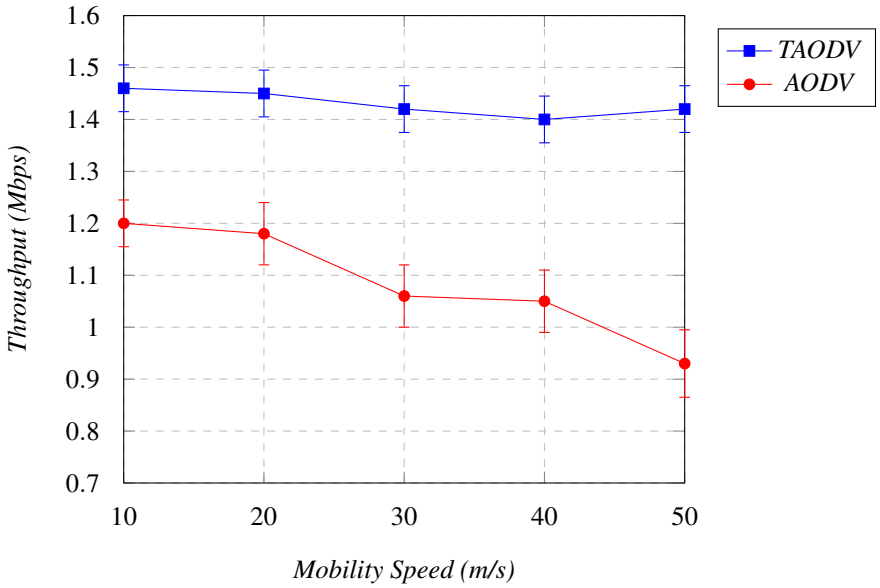

Fig. 5.5 Throughput of TAODV vs. AODV under selfish attack (95% CI)

For TAODV, the throughput started at 1.46 Mbps when the node mobility speed was 10 m/s and decreased slightly to 1.42 Mbps at a node mobility speed of 50 m/s. For AODV, the throughput started at 1.2 Mbps and dropped to 0.93 Mbps over the simulation.

5.2.9 PDR performance varying the node mobility speed

Figure 5.5 shows that TAODV performed better than AODV in terms of PDR from the beginning to the end of the simulation when under selfish attack. However, the PDR of both decreased with an increase in node mobility speed.

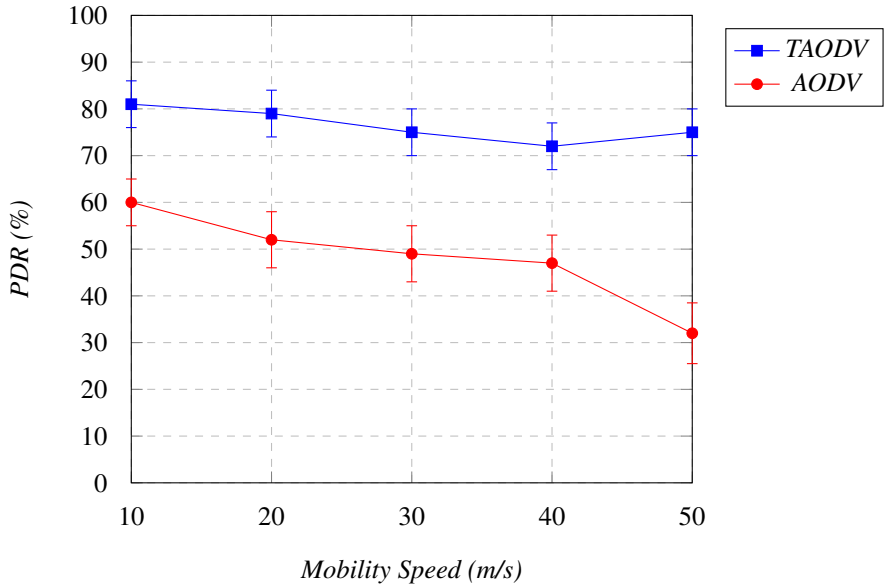

Fig. 5.6 PDR of TAODV vs. AODV under selfish attack (95% CI)

5.2.10 Throughput performance varying the number of malicious nodes

In the following simulation, we compare the throughput of TAODV and AODV, varying the number of malicious nodes while maintaining a constant number of nodes at 100. Figure 5.6 shows that both the TAODV and AODV protocols were affected by an increase in the number of selfish nodes. Throughput decreased with an increase in the number of selfish nodes, but TAODV performed better and was more stable to the end of the simulation.

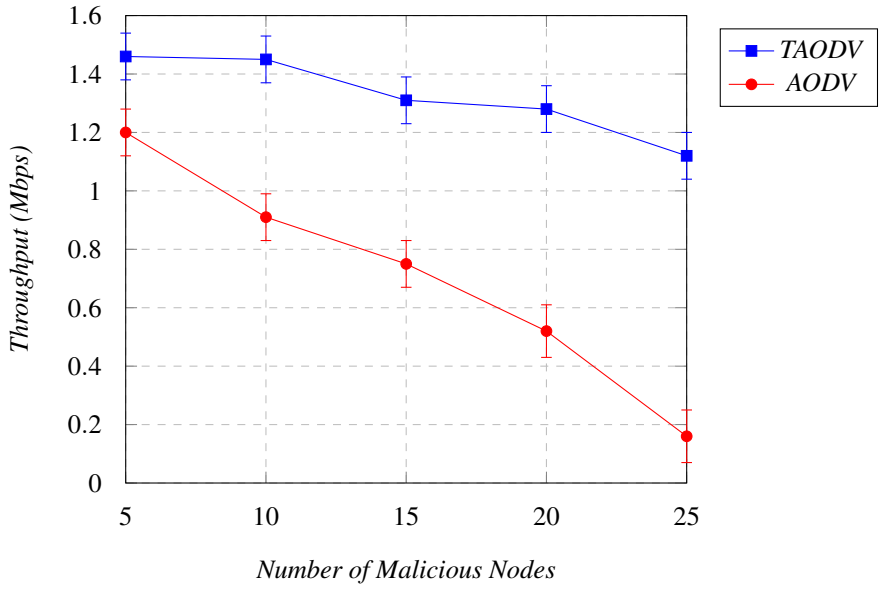

Fig. 5.7 Throughput of TAODV vs. AODV under selfish attack (95% CI)

5.2.11 PDR performance varying the number of malicious nodes

Figure 5.7 shows that the PDR for the TAODV protocol started at 80% with 5 selfish nodes and decreased to 60% with 25 malicious nodes. In TAODV, the PDR decreased, but the network still passed packets successfully, albeit with lower performance. For AODV, the PDR was almost 10%, demonstrating very poor performance close to DoS.

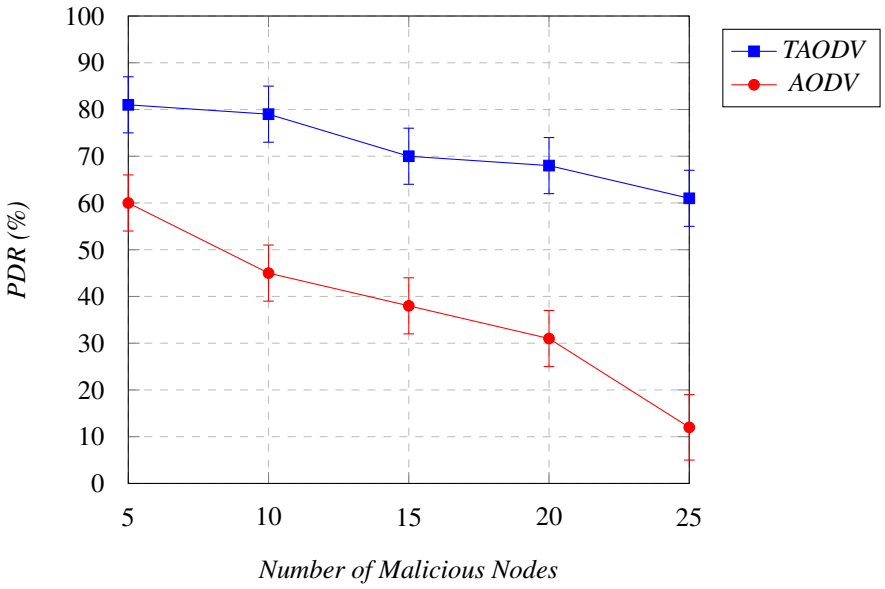

Fig. 5.8 PDR of TAODV vs. AODV under selfish attack (95% CI)

5.2.12 End-to-end delay in TAODV vs. AODV

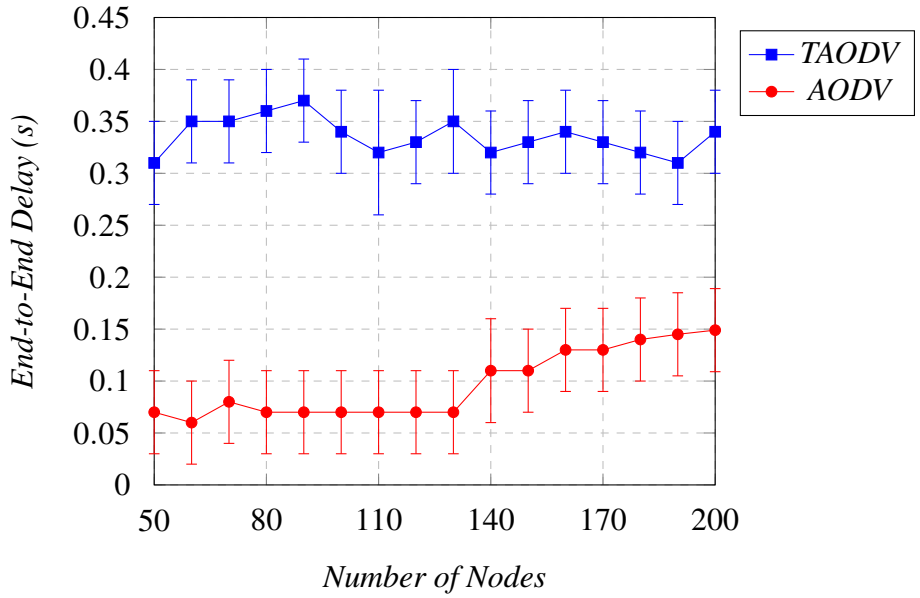

Fig. 5.9 End-to-end delay of TAODV vs. AODV under selfish attack (95% CI)

End-to-end delay is the time taken for a packet to be transmitted from the source node to the destination node [85]. Figure 5.8 Shows that end-to-end delay increased slightly after adding the trust management scheme to AODV. However, the end-to-end delay in TAODV was still at an acceptable level of between 0.3 and 0.36 s. Adding any algorithm to a protocol entails a greater load on nodes, which is likely to increase the time taken to deliver a packet.

5.3 Flooding attack

A flooding attack can be defined as sending a huge amount of multiple fake RREQs to random nodes aiming to consume the network resources and leading to DoS [51, 86]. This is a common type of attack in all kinds of network, including traditional networks with fixed infrastructure.

In a flooding attack, the malicious node floods the network with RREQs to nonexistent destinations to keep the nodes busy processing the fake packets. The aim of this is to consume the bandwidth, node memory, node power and computational resources and to prevent normal operations in the protocol [87]. The malicious node in this type of attack exploits the vulnerabilities in the route discovery phase, allowing an unlimited number of RREQs and accepting any RREQ without validation. In traditional networks, a way

of improving performance in the case of this type of attack is to install a firewall in the hardware, such as servers and routers [88].

In a flooding attack, the attacker selects an IP address that does not exist in the network to extend the search process in the network and consume the nodes' resources [54]. If the attacker knows the scope of IP addresses in the network, the attacker will select IP addresses from outside the range. If the attacker does not know the scope, he/she will select random IP addresses in the hope that they do not exist. If the IP address selected is outside the domain, no node can answer the RREQs. The AODV protocol does nothing to detect fake destination IP addresses because of the nature of MANETs, which allow nodes to join and leave the network freely at any time.

After selecting the IP addresses, the attacker generates a huge number of RREQs for the void IPs without waiting for the RREPs to arrive. When a flooding attack is launched successfully in a MANET, the whole network will be flooded with the fake RREQs sent by the attacker. Both the bandwidth and the resources of the nodes can be exhausted at the same time, which can easily lead to DoS [89]. To give a simple example of shutting down a MANET, each node's capacity for storing the routing table is extremely limited and if the node receives a huge number of RREQs over a short period of time, the routing table will be full and the node will not be able to receive any more RREQs. Hence the node will not be able to serve the real RREQs from cooperative nodes. Figure 5.9 shows how a flooding attack works in MANETs.

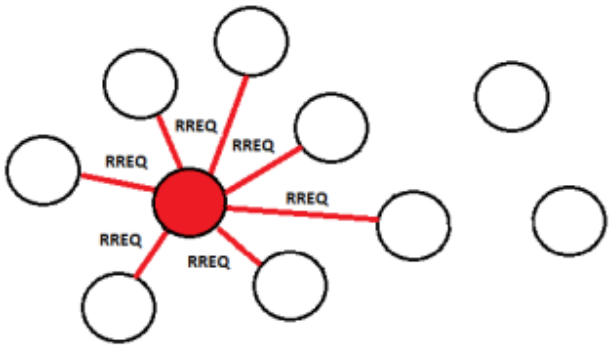

Fig. 5.10 Flooding attack in MANET

5.3.1 AODV vulnerability to flooding attacks

The flooding attack is a type of DoS attack that exploits the routing discovery process of reactive routing protocols such as AODV [54]. AODV already has a default mechanism to prevent flooding attacks at some level. However, this mechanism is vulnerable and it can be hacked by malicious nodes. The default mechanism of AODV has the following barriers to prevent a flooding attack:

1. Limiting the flooding rate:

   AODV has a method to reduce congestion in the network by limiting the number of RREQs a node can send per second. The number of RREQs per second allowed, *RREQ_RATELIMIT*, is 10 by default [74].

2. Limiting the number of routing request attempts:

   AODV limits the number of attempts to find a route made by a source node, by setting a threshold *RREQ_RETRIES*. When a node broadcasts an RREQ, it waits to get an RREP. If the RREP does not arrive within a given number of milliseconds, *NET_TRAVERSAL_TIME*, the node may try again and send another RREQ until the number of retries reaches the limit of *RREQ_RETRIES*, which has a default value of 2 [74].

3. Limiting the time an RREQ can live in the network:

   Time-to-live, *TTL*, is the maximum time allowed for an RREQ to live in the network before it is discarded. Each RREQ has a *TTL_START* value stored in its header. This value increases by *TTL_INCREMENT* each time the node tries to send a new RREQ and the total will be stored in the *TTL*. This continues until the the *TTL* set in the RREQ reaches the *TTL_THRESHOLD*, then the RREQ will be discarded [90].

Theoretically, these methods seem to work well to prevent flooding attacks as they limit the number of RREQs sent by any node in the network and do not allow any RREQ to live in the network forever. However, the malicious node can remove these restrictions by overriding the values of *RREQ_RATELIMIT*, *RREQ_RETRIES* and *TTL_THRESHOLD*. The parameters of this mechanism are accessible by the source node, which has full control to change their values. This allows the malicious node to remove the limitations on the number of RREQs allowed, the flooding rate and the period of time the RREQ can live in the network.

5.3.2 Proposed algorithm to enhance performance under flooding attack

The proposed scheme can validate the values for the *TTL* and *RREQ_RETRIES* parameters in the intermediate node. Thus, instead of relying on the honesty of the source node, the intermediate node checks the values and if the *TTL* value is greater than the *TTL_THRESHOLD* value or the *RREQ_RETRIES* value is greater than 2/s, the source node will be given a trust value of zero. Thus, the intermediate node should be able to detect the malicious behaviour of the flooding attacker and isolate it.

The parameter *RREQ_RETRIES* in AODV controls the RREQs packets that a node is allowed to send per second when attempting to discover a route to a destination. The value of this parameter is set to two packets per second in the default AODV implementation. The purpose of setting *RREQ_RETRIES* to two packets per second is to prevent malicious nodes from flooding the network with RREQs.

However, it is important to know that setting *RREQ_RETRIES* to two packets per second does not necessarily mean that any node attempting to send more than two RREQ packets is malicious. There can be some scenarios where a node may need to resend RREQ packets due to network congestion, temporary link failures, or other reasons. The value of two packets per second is chosen as a trade-off between preventing flooding attacks and allowing logical route discovery attempts to take place.

5.3.3 Implementation

To implement the scheme, a table needs to be added to each node to store the following information: (i) the IP address of the source node sending the RREQ; (ii) the number of RREQs received by the intermediate node; (iii) the trust value of the node. The number of RREQs received is zero by default and increases by 1 each time the intermediate node receives an RREQ. The number resets each second. The scheme works as follows:

1. The intermediate node checks the value of the TTL associated with the RREQ received. If the TTL value is greater than *TTL_THRESHOLD*, the RREQ will be discarded and the source node will be given a trust value of zero and isolated. The intermediate node can obtain the *TTL_THRESHOLD* on its own and compare it to the *TTL* of the RREQ received.

2. Each time the intermediate node receives an RREQ, it stores the IP address of the source node in the table and increases the number of RREQs received from this node by 1. If the number of packets received from the same source node exceeds two packets within one second, the node is trying to flood the network. The source node will be given a trust value of zero and be isolated. Figure 5.10 shows the algorithm of the proposed scheme where the number of RREQs received (n) starts at zero and resets every second.

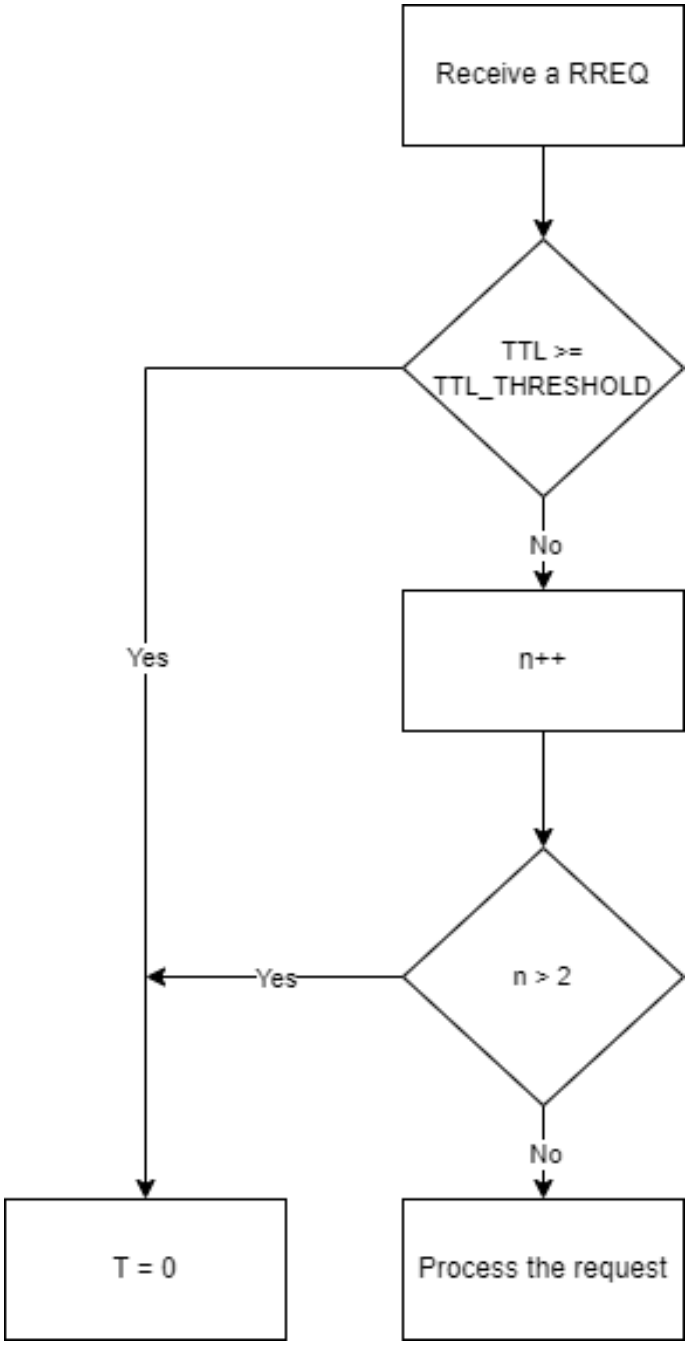

Fig. 5.11 Direct trust algorithm to detect flooding attacks

Algorithm 2 shows the logic of the proposed scheme to detect flooding attacks.

Algorithm 3 Proposed algorithm

n: Number of RREQs

TTL_THRESHOLD: The default threshold given in AODV

TTL: Time the RREQ can live

T: Trust value

```
1:  Receive the RREQ
2:  if TTL > TTL_THRESHOLD then
3:      T = 0
4:  else
5:      n++
6:      if n > 2 then
7:          T = 0
8:      else
9:          Process the RREQ
10:     end if
11: end if
```

5.3.4 Simulation and evaluation

The performance of the scheme is evaluated using the NS3.33 network simulator and measured using the following metrics: throughput, PDR and end-to-end delay. We compare the performance of the plain AODV with the modified version, TAODV, when both are under flooding attack. The parameters used to run this simulation are shown in Table 5.2.

Table 5.2 Parameters used to evaluate performance of the TAODV protocol under flooding attack

Parameter	**Value**	**Unit**
Simulator	NS-3.33	-
Packet size	512	byte
Simulation time	100	second
Simulation area	1200 * 1200	metre
Number of nodes	50, 80, ..., 200	-
Node speed	1	m/s
MAC protocol	802.11b	-
Transmission range	250	metre
Total of simulation runs	10	-
Malicious nodes	5	-
Attack type	flooding	-
Flooding rate	50	packet/s
Routing protocol	AODV, TAODV	-

5.3.5 Performance based on number of RREQs varying the number of nodes

The number of RREQs sent by nodes in the network can be used as a metric to measure the success of the scheme. The scheme should discard or at least reduce the number of fake RREQs by blocking the malicious nodes. That will lead to a reduction in the total number of RREQs living in the network.

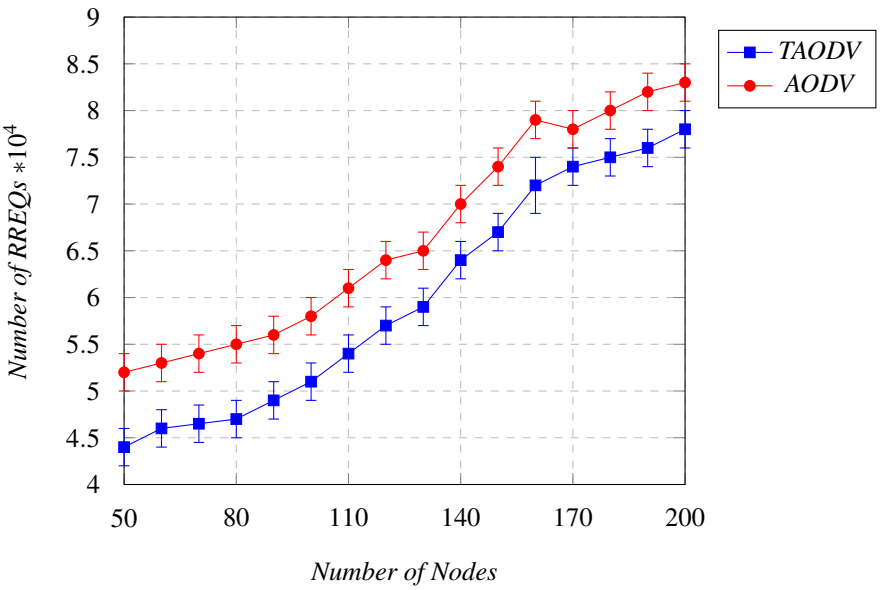

Fig. 5.12 Number of RREQs in TAODV vs. AODV under flooding attack (95% CI)

Figure 5.11 shows that the number of RREQs increases with an increase in the number of nodes in both AODV and TAODV. This is expected because having more nodes means more RREQs will be generated. The addition of the mechanism reduces the number of RREQs in TAODV by blocking the malicious nodes and stopping them from flooding the network with more fake RREQs. For example, if the number of nodes is 100, the number of RREQs is 51,000 in the case of TAODV and 57,000 in the case of AODV. Thus, the scheme was able to prevent 6,000 fake RREQs in 100 seconds.

5.3.6 Performance based on number of RREQs varying the number of malicious nodes

This simulation was run varying the number of malicious nodes from 5 to 25 with a constant number of 200 nodes and a flooding rate of 50 packets per second.

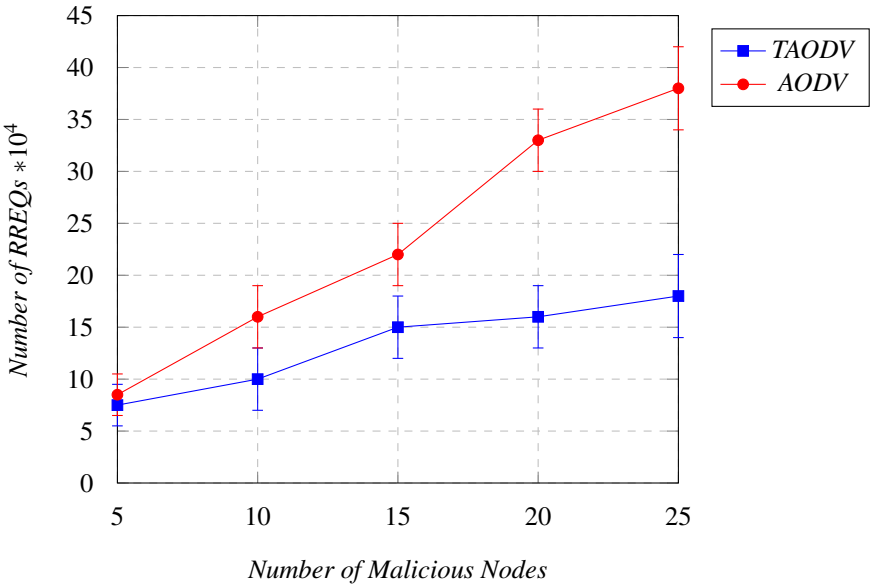

Fig. 5.13 Number of RREQs in TAODV vs. AODV under flooding attack (95% CI)

Figure 5.12 shows that increasing the number of malicious from 5 to 25 increases the number of RREQs from 85,000 to 380,000 in the AODV case, whereas in TAODV, the highest number of RREQs is fewer than 190,000 when the number of malicious nodes reaches the highest value of 25.

5.3.7 Performance based on number of RREQs varying the flooding rate

The simulation was run varying the flooding rate from 50 to 100 packets per second with a constant number of 200 nodes and 5 malicious nodes. The flooding rate is the number of fake packets sent by a malicious node per second [91]. Figure 5.13 shows that in AODV, increasing the flooding rate leads to a rapid increase in the number of RREQs, whereas the number of RREQs in TAODV seems more stable. The reason for this is that any node reaching the limit of permitted RREQs will be isolated regardless of the flooding rate.

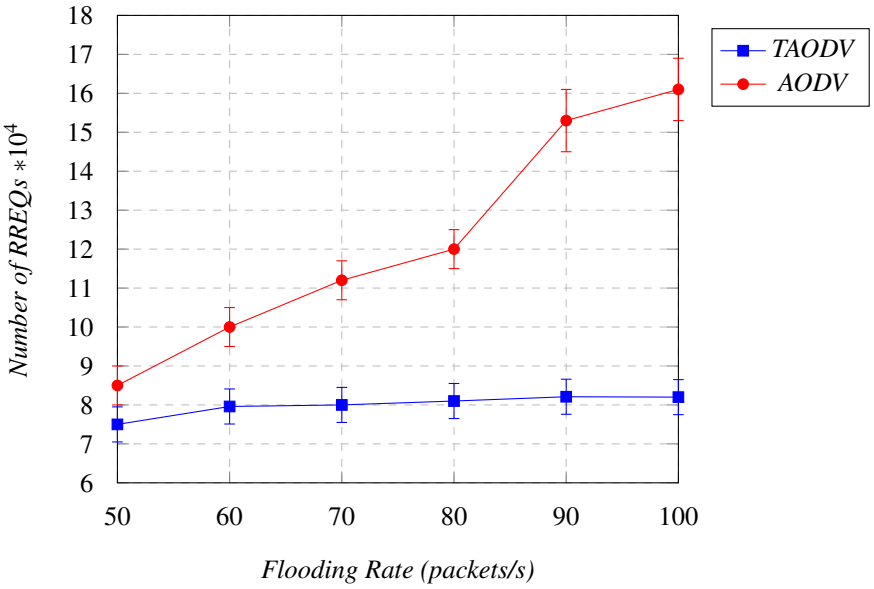

Fig. 5.14 Number of RREQs in TAODV vs. AODV under flooding attack (95% CI)

5.3.8 Throughput performance varying the number of nodes

Figure 5.14 shows that the direct trust management algorithm added to the AODV protocol improved performance in terms of throughput.

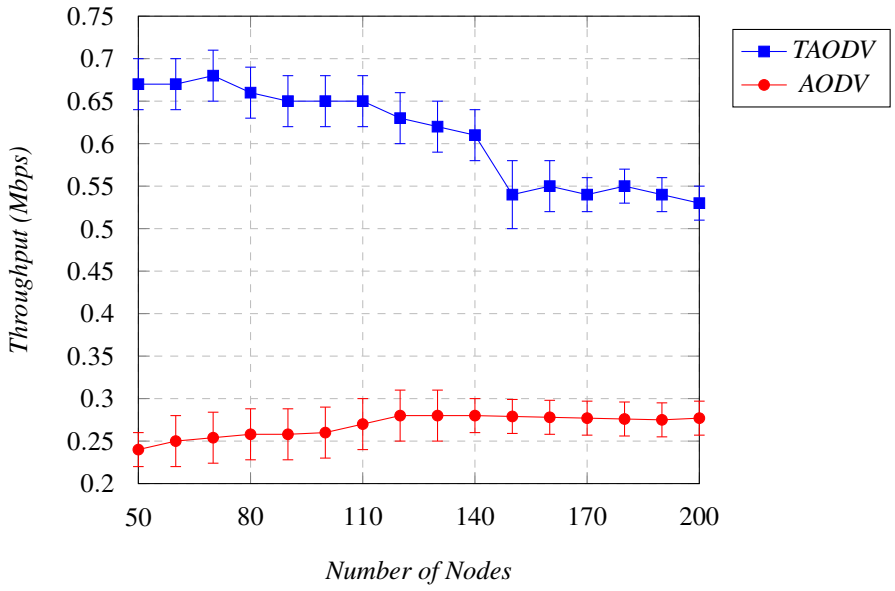

Fig. 5.15 Throughput of TAODV vs. AODV under flooding attack (95% CI)

During the simulation, the throughput of TAODV was better than AODV operating under flooding attack. The throughput improved as a result of isolating the 5 malicious

nodes and preventing them from flooding the network with fake RREQs.

5.3.9 Throughput performance varying the number of malicious nodes

The simulation was run varying the number of malicious nodes from 5 to 50 with a constant number of 200 nodes and a flooding rate of 50 packets per second.

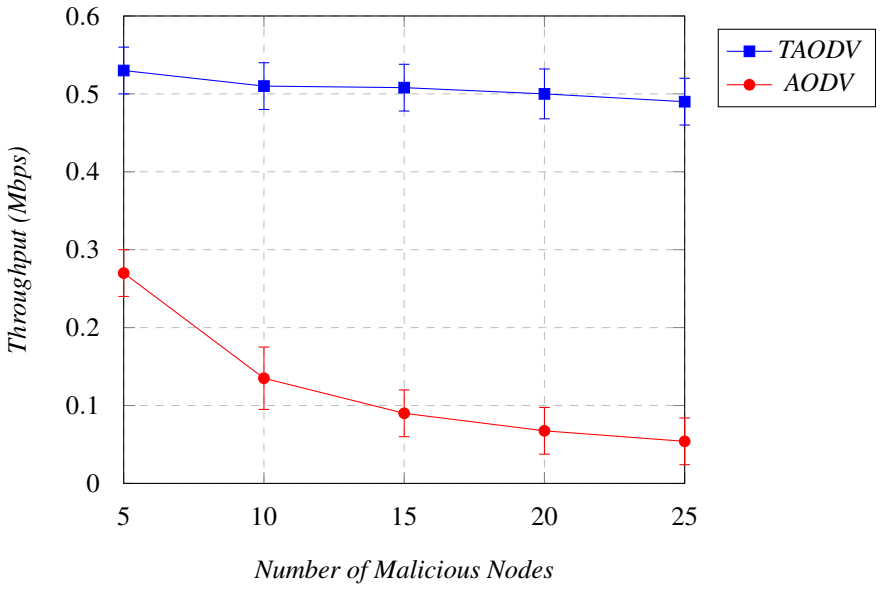

Fig. 5.16 Throughput of TAODV vs. AODV under flooding attack (95% CI)

Figure 5.15 shows that increasing the number of malicious nodes from 5 to 25 sharply reduces the throughput of AODV from 0.27 Mbps to around 0.054 Mbps. In contrast, there was a slight decrease in the TAODV case.

5.3.10 Throughput performance varying the flooding rate

The simulation was run varying the flooding rate from 50 to 100 packets per second with a constant number of 200 nodes and 5 malicious nodes.

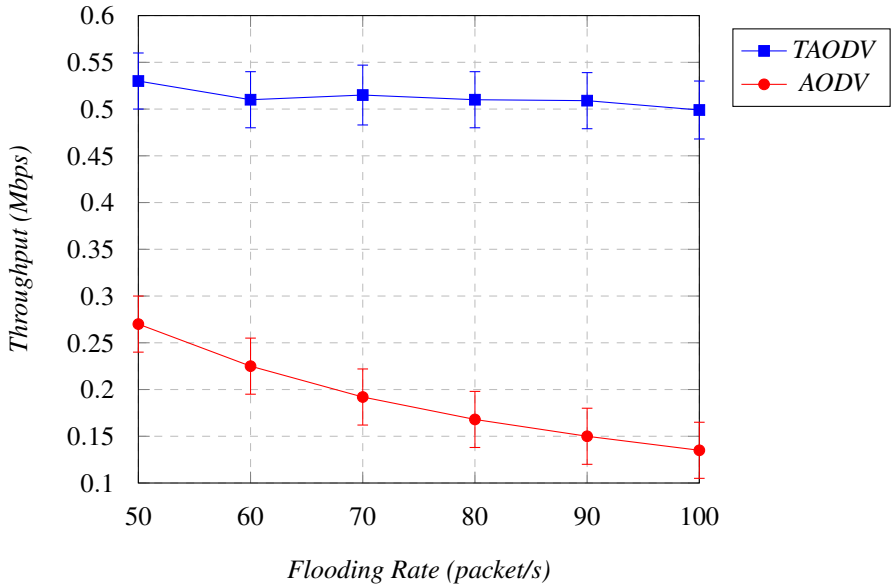

Fig. 5.17 Throughput of TAODV vs. AODV under flooding attack (95% CI)

Figure 5.16 shows that increasing the flooding rate from 50 to 100 packets per second reduces the throughput of AODV from 0.27 Mbps to less than 0.15 Mbps. In TAODV, the throughput looks more stable, with a slight decrease.

5.4 The overhead

End-to-end delay is one of the QoS metrics that can be used to measure the overhead [92]. QoS should be always measured when adding any kind of mechanism to a routing protocol. It is important to determine the trade-off between the cost and the improvement achieved.

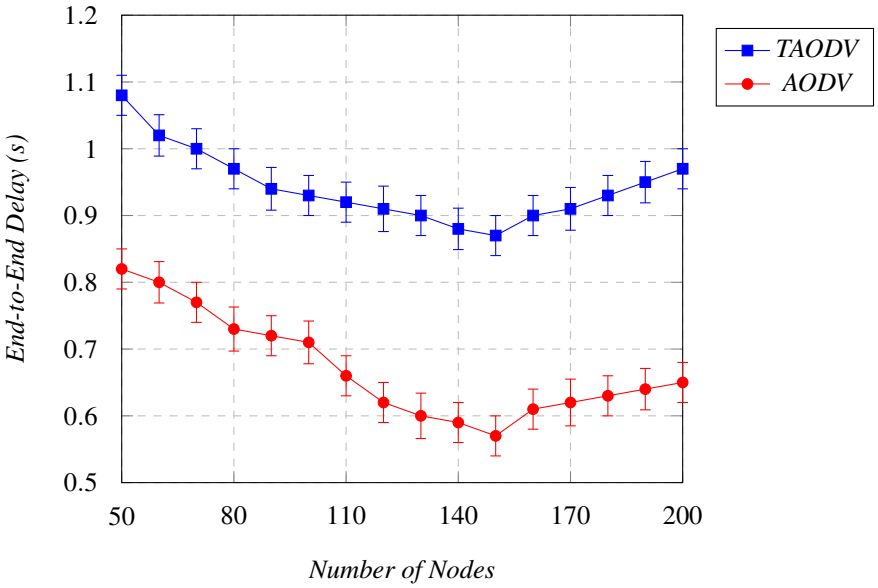

Fig. 5.18 End-to-end delay of TAODV vs. AODV under flooding attack (95% CI)

Figure 5.17 shows that the more work added to the AODV protocol, the greater the effect on the end-to-end delay metric. End-to-end delay is a very sensitive metric and is likely to be affected by adding any load to the protocol. The simulation shows that the end-to-end delay increased by around 0.2 s in TAODV, which seems slight compared to the improvement in the throughput metric and the reduction in the number of fake RREQs.

5.5 Summary

This chapter has evaluated the performance of two direct trust management schemes in the face of selfish and flooding attacks. In the selfish attack case, we used the same scheme as for black-hole attacks in Chapter 4. It showed better performance against selfish attack here than against grey-hole attack in Chapter 4. The reason for this is that a selfish node does not forward any packets to the neighbouring node and this makes it much easier to detect. In contrast, in a grey-hole attack, the malicious node forwards some packets and drops others, which makes it harder to detect.

In the case of flooding attack, we had to implement a different scheme. To detect a flooding attack, we needed to observe the incoming packets from the neighbouring node not the packets sent. The mechanism used a threshold to limit the number of RREQs sent by the source node and another threshold to limit the time an RREQ could live in the network. In Chapter 6, we move on to consider the next type of trust management scheme, namely indirect trust.

Chapter 6

Analysing the Performance of an Indirect Trust-Based Protocol Under Black-Hole Attack

6.1 Introduction

In Chapters 4 and 5, we proposed and evaluated the performance of four direct trust management schemes against black-hole, grey-hole, selfish and flooding attacks. The direct trust management schemes improved MANET performance in terms of the throughput and PDR, with some slight increase in overhead measured by the end-to-end delay metric.

This chapter addresses the second type of trust management scheme in MANETs, indirect trust. The indirect trust approach is more complex than the direct approach and should provide greater improvements in MANET performance. This chapter explains what indirect trust management is and how it differs from direct trust management, specifically in the implementation phase. We propose an indirect trust mechanism and evaluate its performance in the presence of black-hole attack. Then we compare the performance of the proposed indirect scheme to that of the direct scheme proposed in Chapter 4 taking the throughput, PDR and end-to-end delay as the performance metrics.

Comparing the two schemes will indicate which approach provides better QoS. Making the network very secure by adding strict rules to its protocols is not sufficient to guarantee decent QoS as these rules can affect some metrics negatively. This was observed with end-to-end delay in Chapters 4 and 5. The same can be said of the direct and indirect trust approaches: greater complexity does not mean that the indirect approach will provide better QoS than the direct approach in all scenarios.

We use the direct trust performance scheme proposed in Chapter 4 and the indirect scheme proposed in [3] to compare the performance of the principles of direct and indirect

trust. However, these principles are broad and open to the implementation of many concepts or technologies, such as encryption and machine learning. Thus, when comparing the performance of the two schemes, it is not the case that the results will apply to all direct and indirect trust schemes. The results will apply only to these two specific schemes, so at the end of this chapter, we wil not conclude that the scheme based on the indirect trust principle performs better than the schemes based on the direct trust principle or vice versa.

6.2 Quality of service (QoS)

QoS relates to the assurance or guarantee a network provides about the level of service afforded an application [5]. It can also be defined as the network's ability to provide a guaranteed level of service to the users and the applications running on that network [5]. The requirements of this service usually include performance metrics, such as throughput, jitter, reliability and end-to-end delay. Each additional algorithm or mechanism added to any routing protocol, such as a trust management scheme, affects one or more of these metrics. In MANET, the nodes have different capabilities and thus some nodes will yield low QoS when required to do additional tasks.

It is crucial that MANETs provide an appropriate level of QoS, especially when running real-time or multimedia applications. However, it is a challenge to maintain QoS at a high level when adding more load and tasks to the nodes, such as managing trust in the other nodes in MANETs. When implementing a trust management mechanism in a MANET routing protocol, it may be possible to detect and isolate malicious nodes and thus achieve better throughput and PDR when the network is under attack. However, the overhead added to the nodes might reduce the QoS and make the network unreliable for some applications.

The definition of QoS and its parameters different from one application to another depending on the specific requirements and needs of the application [92] . For example, in delay-sensitive applications, such as multimedia applications, a key QoS parameter is likely to be low end-to-end delay. In other applications, the QoS parameter could be the consumption of very little bandwidth. In applications that download files, QoS can be measured using the throughput parameter as when downloading files, more packets per second means better downloading and hence better QoS. Real-time text chat applications may not require high throughput, but they need the packets sent to arrive at the destination in the shortest possible time. For such applications, end-to-end delay can be used as the parameter to measure QoS.

When implementing trust management solutions, end-to-end delay is always a QoS parameter that is affected. If nodes are busy calculating the trustworthiness of other nodes, they will be unable to deliver packets as fast as they would without having to do those

tasks. In MANETs, end-to-end delay is the sum of the delays occurring sequentially across the intermediate nodes en route to the destination node [85].

6.2.1 Challenges in assuring QoS in MANETs

Puttamadappa et al.[5] outline seven important challenges in assuring QoS in MANETs:

1. Dynamic topology. The nodes in MANETs have no restrictions on travelling in random directions at random speeds, so the topology changes dynamically. A dynamic topology can cause frequent path breaks, making it necessary to establish new paths. This will increase the end-to-end delay.

2. Imprecise state information. The nodes in MANETs maintain state information on bandwidth, jitter, delay, loss rate, errors, stability, ID, source address and destination address. Due to the dynamic topology, this state information is inherently imprecise.

3. Lake of central coordination. The lack of any central coordination makes assuring QoS in MANETs more complicated.

4. Error-prone shared radio channels. The radio waves through the wireless medium suffer from interference and attenuation from other wireless devices operating in the same transmission range.

5. Hidden terminal problem. This problem occurs when two or more nodes send packets at the same time to one receiver node. This might cause crashes in the destination node and reduce QoS.

6. Limited resource availability. MANETs have limited resources, such as processing capability, battery life, bandwidth and storage space. These resources need to be used efficiently to provide a good level of QoS.

7. Insecure medium. A MANET is open for any node to participate in forwarding packets. This is considered a vulnerability that allows malicious nodes to connect to the network and launch attacks.

6.3 Indirect trust management

The indirect trust approach is based on receiving recommendations about the trustworthiness of other nodes in the network. This technique gives new nodes just joining a MANET the confidence to interact with unknown nodes and establish trusted links with them [93]. The indirect trust principle is similar to the direct trust principle, except that

in the indirect approach, the nodes share the trust values they have with other nodes in the network. In the direct approach, each node calculates the trust values itself without receiving recommendations from other nodes. In the indirect approach, each node shares its trust values with other nodes in the network, so when a node discovers a malicious node, it will inform all other nodes within its transmission range.

The indirect approach involves evaluating the trustworthiness of nodes based on their behaviours and interactions with other nodes in the network. In this approach, nodes do not trust each other directly, as in the direct approach, but instead depend on the recommendations received from neighbouring nodes to make trust decisions. The indirect approach thus uses a reputation system to collect feedback from multiple nodes in the network. The reputation of a node is a measurement of its past behaviour in forwarding data packets in the network. Reputation values are then used to make a decision about whether to trust or distrust a given node.

Despite the potential ability of the indirect approach to provide better performance in MANETs in the presence of attacks, it faces many challenges different from those faced by the direct approach. For example, a malicious node may broadcast fake positive recommendations about itself or about another malicious node. Thus, we may need to find a way to validate the recommendations received and ensure they are based on real observations.

6.4 Implementation of direct vs. indirect trust management

Chapter 2 outlined four steps in trust management implementation: trust initialisation, information collection, trust calculation and decision making [15]. In each of these steps, there are many differences between the direct and indirect trust management approaches.

1. **Trust initialisation:** In direct trust management, trust is initialised by setting a neutral value. For example, if the trust value lies between zero and 1, a node will assign an initial trust value of 0.5 to a node it has not interacted with before. This means no information about the node is available and the source node will need to take the risk and start interacting with it until some information about it becomes available. In the indirect approach, trust is initialised based on recommendations received from other nodes. This means that when a new node joins a MANET, it will use the recommendations received from others to initialise trust with its neighbours before starting any interactions. In the indirect approach, a node should distinguish cooperative nodes from malicious nodes before it interacts with them.

2. **Information collection:** In the direct approach, evidence is collected through direct observation. In the indirect approach, evidence is collected through direct observations and receiving recommendations. The direct approach does not depend on the reputation of the nodes, unlike the indirect approach.

3. **Trust calculation:** Trust is calculated by establishing certain algorithms and equations that run in each node in both types, direct and indirect. In this step, the only difference between the two types lies in storing the outcomes of the calculation. In direct trust management, the trust values are stored locally in the nodes, whereas in indirect trust management, the trust values are distributed over the nodes. In other words, direct trust is centralised and indirect trust is decentralised.

4. **Decision making:** Making the decision whether to interact with a certain node or not is the same in both types. Once the trust value is available, regardless of whethere it is calculated locally or received from other nodes, the decision is made by the node itself. In both types, each node should have a list that contains its neighbouring nodes' IP addresses and their trust values. Before interacting with a node, it checks the trust value to decide whether to interact with it or not.

Table 6.1 summarises the differences between the direct and indirect trust management approaches in terms of the steps in implementation.

Table 6.1 Direct vs. indirect trust management steps

Step	Direct trust	Indirect trust
Trust initialisation	Neutral value	Using recommendations
Information collection	Direct observation	Direct observation + Reputation built through other nodes
Trust calculation	Trust values stored locally	Trust values distributed over the network
Decision making	Made by the node	Made by the node

The indirect trust principle seems more complicated and effective, but at the same time it should add more overhead to the nodes than the direct approach [94]. This is the focus of the investigation in this chapter.

6.5 Proposed indirect trust management scheme

In the proposed indirect trust management scheme, the trust value of a target node is built upon its reliability during the packet routing processes. Each node in the network

monitors the reliability of its neighbouring nodes in terms of its ability to pass packets through the route. Each node records positive (α) and negative (β) observations of its neighbouring nodes and uses them to calculate trust values using Bayesian inference. Bayesian inference is a statistical method used to update the probability of an assumption when more information (evidence) becomes available [3]. The nodes share their trust values with other nodes and continuously update the trust values of target nodes as they receive more recommendations.

The parameters used in this mechanism are node battery, the node's participation in routing activities, packet forwarding ability and the packet forwarding queue length. Each observation can be either negative or positive and the number of observations is not limited. There is a certain condition for each parameter to gain a positive or negative observation. For example if the battery level is greater than 30%, it is considered a positive observation and α will increase by 1. If the battery level is less than 30%, it is a negative observation and β will decrease by 1. The trust values are calculated based on the number of positive (α) and negative (β) observations. The value calculated will be shared with other nodes within transmission range. Table 6.2 shows how each parameter gains a positive or negative observation.

Table 6.2 Observation parameters

Parameter	**Positive observation** ($\alpha++$)	**Negative observation** ($\beta++$)
Node battery	If node battery > 30%	If node battery <= 30%
Participation in routing activities	Initialising an RREQ or RREP	Initialising an RERR or dropping a control packet
Ability to forward packets	For each data packet forwarded	For each data packet dropped
Queue length	If queue is empty > 30%	If queue is empty <= 30%

The proposed trust management scheme for MANETs employs four relevant parameters to assess the trustworthiness of neighboring nodes: Node battery, Participation in routing activities, Ability to forward packets, and Queue length. These parameters collectively offer insights into a node's energy availability, engagement in routing tasks, packet forwarding efficiency, and queue congestion, allowing the source node to make informed decisions about node reliability. While these parameters are practical and effective for trust evaluation in the specific MANET scenario, their universal applicability in real-life situations may require adaptation and further research to suit varying network conditions and requirements.

The chosen threshold of 30 percent for battery level and queue length in the proposed direct trust management scheme for MANETs is specifically tailored for the purpose of this experiment and may not be universally suitable for real-life situations. While this

value demonstrated positive results in the controlled scenario with specific simulation parameters, its efficacy in real-world MANETs could be subject to various environmental and operational factors. Real-life deployments may necessitate extensive evaluation and calibration to determine the most appropriate threshold values based on the specific characteristics and requirements of the network.

6.5.1 Nodes' reliability or unreliability, or uncertainty

The four parameters in Table 6.2 are used to calculate the node's reliability *(r)* or unreliability *(n)*, or uncertainty *(u)*. Reliability *(r)* denotes the probability that the node provides reliable routing. Unreliability *(n)* denotes the probability that the node provides unreliable routing. Uncertainty *(u)* denotes the probability of being unable to determine whether the node is reliable or unreliable. These three values are calculated through direct monitoring at each interval (30 s) and are shared with the other nodes within transmission range as recommendations. Once a node has a sufficient number of observations concerning its neighbouring node, the α and β parameters are used in a probability distribution called the beta distribution to calculate the three values, r, n and u.

In the context of the trust management in MANETs using the beta distribution, the node's reliability (r), unreliability (n), and uncertainty (u) are defined as follows:

1. Reliability (r): The reliability of a neighboring node represents the likelihood that the node will perform its network responsibilities in a trustworthy and dependable manner. A higher value of reliability indicates that the node has demonstrated consistent positive behavior and can be trusted for data forwarding and routing tasks.

2. Unreliability (n): The unreliability of a neighboring node reflects the probability that the node will behave in an untrustworthy or malicious manner.A higher value of unreliability signifies a node that has exhibited frequent negative behaviors and is therefore deemed less reliable for network operations.

3. Uncertainty (u): The uncertainty associated with a neighboring node indicates the lack of sufficient information to confidently categorize the node as reliable or unreliable. In such cases, further observations are required to gather more data and make a decisive determination about the node's trustworthiness.

The relationships between these three parameters can be summarized as follows:

1. As the reliability (r) of a node increases, its unreliability (n) decreases. A highly reliable node will have a lower probability of exhibiting untrustworthy behavior.

2. As the unreliability (n) of a node increases, its reliability (r) decreases. A node with a higher unreliability score is more likely to be considered untrustworthy.

3. Uncertainty (u) arises when there is a balance between positive and negative observations (alpha and beta). As more observations are gathered, the balance tips towards either reliability or unreliability, reducing the uncertainty associated with the node.

6.5.2 Beta distribution

The beta distribution is a continuous probability distribution that is defined in the range [0, 1]. The two parameters, α and β, determine the shape of the distribution, which reflects a node's behaviour. When the number of positive α observations is greater than the number of negative β observations, the result will demonstrate that the node is exhibiting good behaviour and vice versa. The beta distribution function can be written as follows:

$$f(x|\alpha,\beta) = (1/B(\alpha,\beta)) \times x^{\alpha-1} \times (1-x)^{\beta-1} \tag{6.1}$$

Since the beta distribution is defined in the range [0,1], it is appropriate for use in generating trust values for MANETs. The α and β parameters accept infinity values, meaning a node's behaviour can be observed over a long time.

The use of the beta distribution in the proposed MANET trust management scheme is justified due to its adaptability, flexibility, and probabilistic nature. In the context of uncertain and dynamic trust observations, the beta distribution accommodates limited data and varying behaviors, making it suitable for modeling trustworthiness. Its ability to incorporate prior beliefs and provide probabilistic outcomes aligns well with the need for nuanced decision-making in uncertain environments. Furthermore, the beta distribution's transparency and applicability in real-world scenarios make it an effective tool for capturing the complexities of trust assessment in MANETs, enhancing the scheme's ability to evaluate and respond to node behaviors accurately.

6.5.3 Calculating a node's reliability or unreliability, or uncertainty based on direct observation

Using the beta distribution, we calculate *r*, *n* and *u*. Let us assume that we have two nodes, Node A and Node B. Node A observes the positive (α) and negative (β) activities of Node B and calculates the *r*, *n* and *u* for each interval (30 s), as follows:

1. First, Node A needs to calculate uncertainty (u) for Node B:

$$u = \frac{12\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{6.2}$$

The decision to multiply the numerator by 12 was strategically made to amplify the uncertainty parameter in situations where the number of positive α and negative β observations is relatively low. This adjustment is designed to reflect the intuitive notion that when facing fewer observations, a higher degree of uncertainty is warranted, as making an accurate decision becomes more challenging. By multiplying the numerator, the resulting uncertainty value is effectively elevated, ensuring that even with minimal alpha and beta values, the uncertainty attains its maximum possible value of one. This approach aligns with the principle of enhancing the uncertainty parameter's sensitivity to variations when the observational data is limited. Consequently, in scenarios characterized by the lowest possible values of α and β (both being one), the resulting uncertainty value is intentionally maximized, signifying the highest level of uncertainty.

Conversely, as the number of observations increases, the uncertainty value naturally diminishes, reflecting a higher level of confidence in decision-making. This mathematical adaptation serves to strike a balance between sensitivity to limited data and trustworthiness of assessments, offering a nuanced perspective that accounts for the inherent uncertainty stemming from limited observations.

2. After calculating the uncertainty, Node A can calculate the reliability of Node B:

$$r = \frac{\alpha}{\alpha+\beta}(1-u) \tag{6.3}$$

3. Node A then calculates the unreliability of Node B:

$$n = \frac{\beta}{\alpha+\beta}(1-u) \tag{6.4}$$

Node A will store these three values in its trust table and will also share them with all the nodes within its transmission range through observation packets. The values of u, r and n are always between zero and 1.

6.5.4 Modifying the AODV protocol

AODV uses hop counts as a parameter to select the route to the destination node. It selects the shortest route with the fewest hops. This mechanism is useful when all nodes in the network are cooperative, but not when a black-hole attack is running. We thus modify ADOV to make it take the reliability of the neighbouring node (r) into account when selecting the route, not only the number of hops. A node will use the following equation

to select a route when receiving more than one RREP, where P is the weight given to the number of hops and r is the reliability value of the node.

$$Trustworthiness\_value = \frac{P}{\text{number of hops}} + (1-P) \times r_i \tag{6.5}$$

The best result for this experiment is assigning 40% for the shortest route and 60% for r. Since the number of hops parameter is very important and cannot be ignored, the scheme includes it with a weight of 40%. After adding the weights, the equation looks as follows:

$$Trustworthiness\_value = \frac{0.4}{\text{number of hops}} + 0.6 \times r_i \tag{6.6}$$

6.6 Implementation and evaluation

This indirect trust management scheme was implemented in the AODV routing protocol using the NS-3.33 network simulator. The modified version of the protocol is called ITAODV. Here, the metrics used to measure the performance of ITAODV are throughout and PDR.

We compare the performance of the direct trust protocol TAODV proposed in Chapter 4 with ITAODV. Both protocols are evaluated based on the same parameters, shown in Table 6.3.

Table 6.3 Simulation parameters

Parameter	**Value**	**Unit**
Simulator	NS-3.33	-
Attack	Black-hole	-
Simulation time	180	second
Simulation area	800 * 800	metre
Malicious nodes	15	-
Mobility model	Random	-
Packet size	512	byte
Node speed	10 - 50	m/s
Pause time	10	second
Traffic type	UDP	-
Routing protocols	TAODV, ITAODV	-

6.6.1 Throughput performance varying the node mobility speed

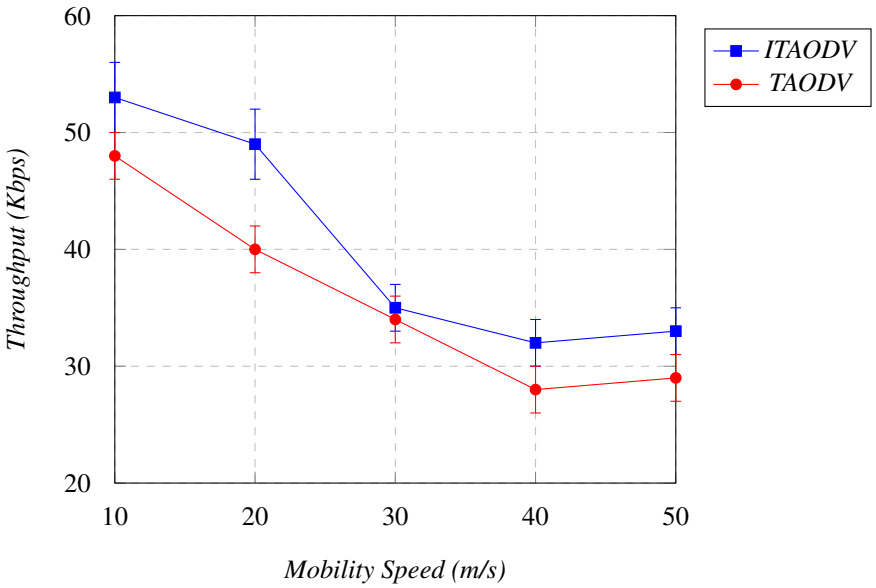

Fig. 6.1 Throughput of ITAODV vs. TAODV under black-hole attack (95% CI)

Figure 6.1 shows the impact of increasing the node mobility speed on the throughput of ITAODV and TAODV. The throughput of ITAODV drops from 53 to 33 Kbps, whereas the throughput of TAODV drops from 48 to 29 Kbps. ITAODV performs better than TOADV in terms of throughput when increasing the node mobility speed.

6.6.2 PDR performance varying the node mobility speed

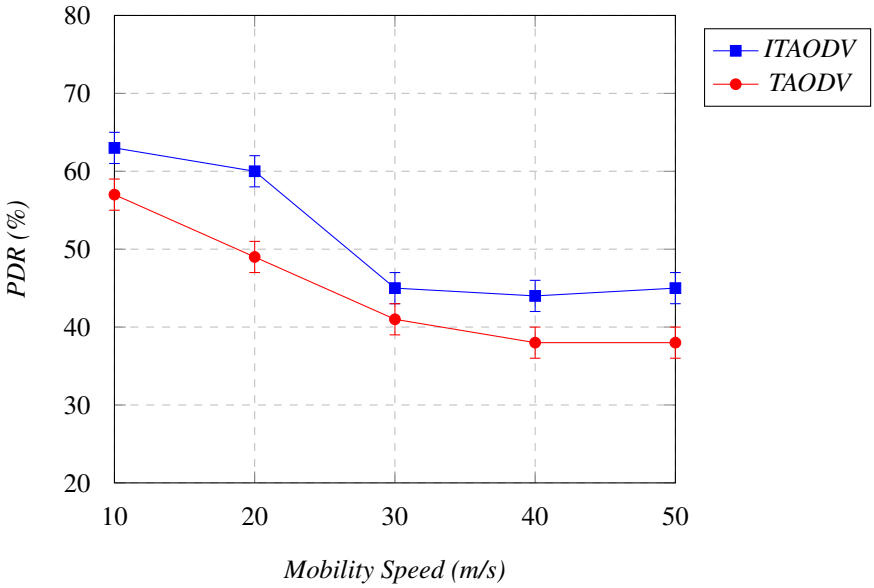

Fig. 6.2 PDR of ITAODV vs. TAODV under black-hole attack (95% CI)

Figure 6.2 shows that the PDR of ITAODV decreases from 63% to 45% with an increase in node mobility speed from 0 to 50 m/s. In TAODV, the PDR decreases from 57% to 38%. From the graph, we can see that the PDR is better in ITAODV than in TAODV.

6.6.3 Throughput performance varying the number of malicious nodes

In the following two simulations, we evaluate the performance of ITAODV and TAODV using the same parameters shown in Table 6.3, with a constant speed of node mobility of 15 m/s and varying the number of malicious nodes from 5 to 25.

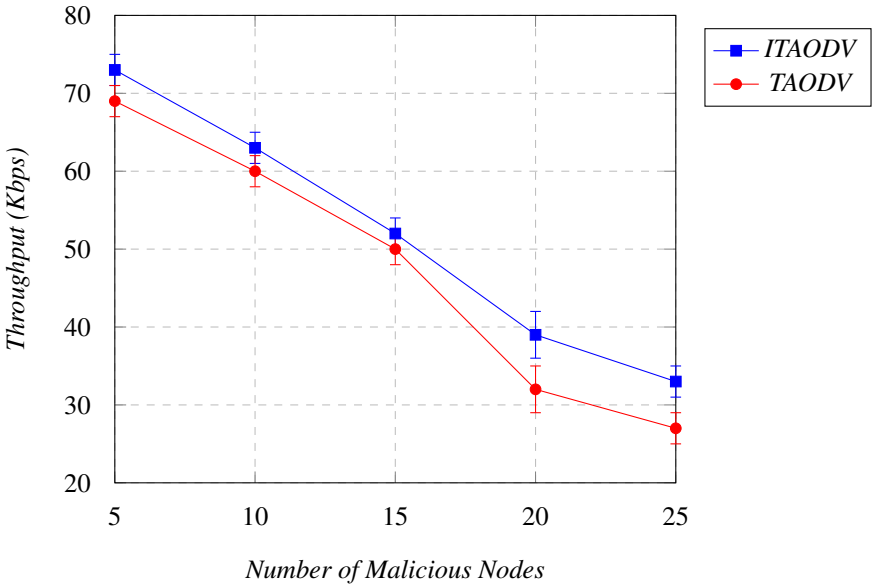

Fig. 6.3 Throughput of ITAODV vs. TAODV under black-hole attack (95% CI)

Figure 6.3 shows a decrease in the performance of both ITAODV and TAODV in terms of throughput when increasing the number of malicious nodes running black-hole attacks. When more than one node launches a black-hole attack at the same time, it takes more time for both ITADOV and TAODV to detect the malicious nodes and isolate them. During the time it takes to detect them, they can successfully drop a considerable number of packets, which will reduce the throughput. Since the nodes receive recommendations from each other in ITAODV, they take less time to detect the malicious nodes than in TAODV. This is one of the reasons why ITAODV performed better than TAODV from the beginning to the end of the simulation.

6.6.4 PDR performance varying the number of malicious nodes

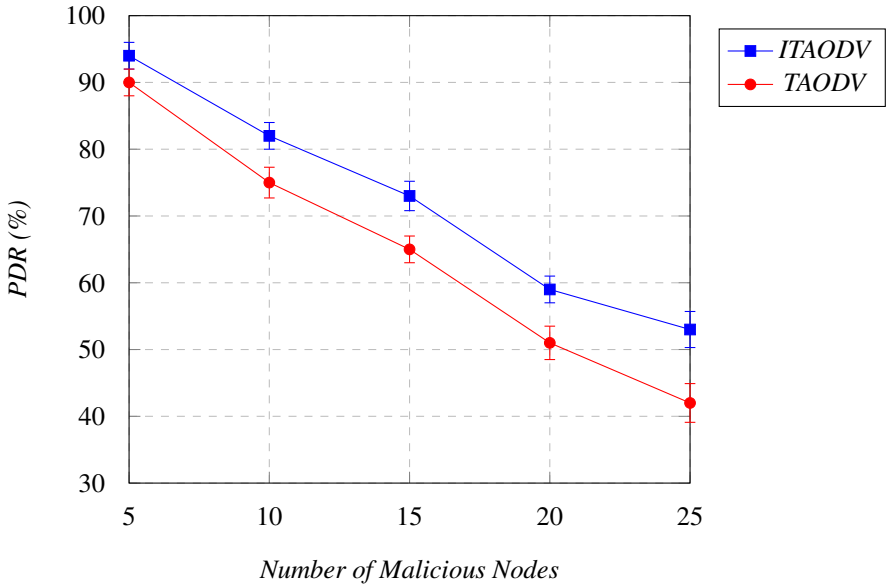

Fig. 6.4 PDR of ITAODV vs. TAODV under black-hole attack (95% CI)

Usually, PDR and throughput increase and decrease in tandem. Both ITAODV and TAODV started with a high PDR, 94% and 90% respectively, with 5 malicious nodes. This indicates that when there are few malicious nodes, it takes little time to detect and isolate them and not many packets are lost. However, at the end of the simulation, when the number of malicious nodes increased to 25, the PDR was considerable lower for both protocols. However, ITAODV performed better than TAODV from the beginning to the end of the simulation.

6.7 Overhead in the direct vs. indirect trust management schemes

In Chapter 4, we found that the additional delay caused by adding the direct trust management scheme to the protocol increased from 0.002 s to 0.006 s as the number of nodes increased throughout the simulation. This additional delay is very low and should not affect the QoS of the network. We now look at the same QoS parameter, the end-to-end delay, for the proposed indirect trust management scheme and compare it with the direct trust management scheme.

We ran a simulation to obtain the overhead of ITAODV under normal circumstances, with no attacks and all nodes cooperating in forwarding the packets. The simulation was run using the parameters given in Table 6.3.

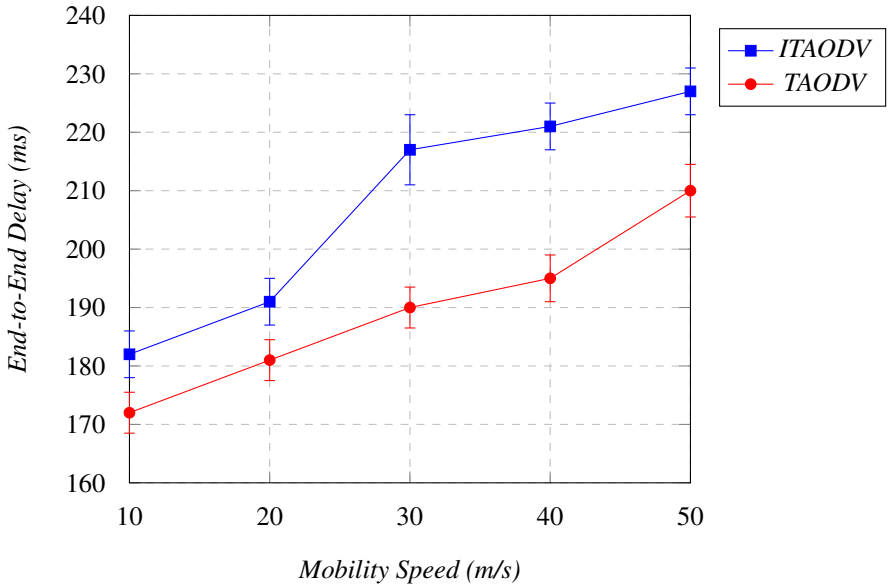

Fig. 6.5 End-to-end delay of TAODV vs. ITAODV under black-hole attack (95% CI)

The simulation shows that the overhead of ITAODV is grater than the overhead of TAODV in terms of the end-to-end delay throughout the simulation. The ITAODV end-to-end delay starts at 182 ms when the node mobility speed is 10 m/s and keeps increasing to reach 227 ms when the node mobility speed is 50 m/s. In the TAODV case, it starts at 172 ms when the node mobility speed is 10 m/s and keeps increasing over the simulation to reach 210 ms when the node mobility speed is 100 m/s.

The indirect trust management scheme requires the node to undertake more tasks than the direct trust management scheme. This is why the overhead in ITAODV is higher and the end-to-end delay indicates lower QoS for the indirect scheme than the direct scheme.

In the specific case of the two proposed mechanisms, direct and indirect, the reasons of the greater end-to-end delay in the indirect scheme are as follows:

1. The indirect trust mechanism involves calculations for six parameters, whereas the direct trust mechanism is based on only one. A node in the indirect trust scheme has to collect the following data about its neighbouring nodes; packet forwarding rate, available battery, battery draining rate and congestion around the node. In contrast, in the direct scheme, the node need only calculate the number of packets dropped by its neighbouring node.

2. The nodes' headers store more data in the indirect mechanism than in the direct mechanism.

3. The direct mechanism employs a less complicated algorithm to decide whether the target node is trustworthy or malicious.

6.8 Summary

This chapter has clarified the difference between the direct and indirect trust management approaches by undertaking a practical evaluation of both direct and indirect schemes. We compared the performance of the direct trust-based routing protocol TAODV with the indirect trust-based routing protocol ITAODV. The outcome was that ITAoDV is better in terms of throughput and PDR, but TAODV is better in terms of end-to-end delay.

The direct scheme is less complicated and more convenient if the node has sufficient observations about the participating nodes in the network. In large networks, direct trust may not be as useful as in small ones because there will be many nodes that have not been in direct contact with each other before. In this case, it is more effective to get recommendations about the target node in the trust initialisation phase. These recommendations can be combined with the node's own observations to calculate the trust values of the target node [95].

It is clear that there is no right or wrong option: each strategy has its advantages and disadvantages. The direct trust mechanism may be the better option if the network is quite small and the number of nodes is limited. When the same nodes communicate with each other frequently, there is no need to implement an indirect trust scheme to improve performance. As most nodes will already have communicated with each other before, they will already have trust values to draw on and have no need of recommendations. In this case, a direct scheme is better.

However, if the MANET is large, with many nodes joining and leaving the network continuously, an indirect trust scheme may be the better option. In large networks, most of the nodes will not have prior communications on which to base trust values and thus it is better for the performance of the network to receive recommendations rather than initialising trust with a neutral value, which has some level of risk.

Chapter 7

Conclusion and Future Work

This thesis has explained what MANETs are and how MANET routing protocols work. It has described the vulnerabilities that MANETs suffer from, which can be exploited by malicous nodes, leading to negative effects on performance. The thesis has explained how MANETs differ from traditional networks, which is why different solutions are needed for MANETs.

The thesis then introduced trust management as a promising principle that could solve many issues in MANETs. Four direct trust management schemes were proposed and their performance evaluated against black-hole, grey-hole, selfish and flooding attacks. In addition, an indirect trust scheme was proposed and evaluated in tackling a black-hole attack. The aim of this was to compare the outcome of the direct and indirect trust mechanisms in terms of throughput, PDR and end-to-end delay. The proposed schemes show a considerable improvement in the performance of MANETs in the presence of the four attacks, but with some overhead. The increase in overhead was expected as adding any tasks to the routing protocol will likely consume more resources.

Based on the results obtained in this thesis, we can determine the impact of each attack on MANETs, the possibility of detecting an attack using the trust-based management schemes and whether the attack could lead to DoS, as shown in Table 7.1.

Table 7.1 Impact of attacks and possibility detection

Network attack	Impact level	AODV vulnerability	DoS	Possibility of detection
Black-hole	High	Lack of forwarding confirmation	Yes	Possible
Grey-hole	Medium	Lack of forwarding confirmation	Yes	Less possible than for black-hole
Selfish	Medium	Lack of forwarding confirmation	No	More possible than for grey-hole
Flooding	High	Unlimited number of RREQs	Yes	Possible

The impact level is determined by the extent to which the attack can harm the throughput metric. DoS is determined by the PDR metric. If the attack succeeds in lowering the PDR to around 10%, DoS is possible. The possibility of detection is determined by the improvement in performance after implementing the trust schemes.

A selfish attack cannot cause DoS if there are sufficient cooperative nodes in the network to establish routes and pass data packets. The is because the selfish node does not attract traffic by sending misleading RREPs to the source. Black-hole and grey-hole attacks can lead to DoS as they attract traffic even if there are sufficient cooperative nodes. A flooding attack can easily cause DoS by flooding the network with fake RREQs.

7.1 Thesis summary

Chapter 1 started by explaining what MANETs are and why they are important. The chapter illustrated the research problem, aims, objectives, motivation, contributions and structure.

Chapter 2 provided detailed background on MANETs, describing in depth how they work and how they differ from traditional networks. The chapter explained the concept of trust management, MANET routing protocols and the methodology used in the thesis. The chapter included a brief explanation of the proposed approaches.

Chapter 3 identified the prime parameter to be used to evaluate the performance of the proposed schemes later. The chapter showed that node mobility has a significant effect on the performance of three common routing protocols, AODV, DSR and DSDV. Drawing on the results of the analysis, we concluded that node mobility speed is a very important parameter and should always be taken into account when evaluating proposed solutions for MANETs.

Chapter 4 set out the implementation and evaluation of the proposed direct trust management approach to detect black-hole and grey-hole attacks. The chapter started by describing the vulnerabilities of AODV that allow black-hole and grey-hole attacks and addressed how the proposed scheme reduced these vulnerabilities. The scheme was implemented in the AODV protocol and evaluated in the presence of black-hole and grey-hole attacks independently.

In **Chapter 5**, two more attacks were introduced, selfish and flooding. Because the selfish attack is very similar to the black-hole attack in terms of dropping all packets received, we used the same approach to detect it. The results showed that it is possible to detect both attacks, black-hole and selfish, using the same direct trust management algorithm, but it was less efficient in the selfish attack case. A different scheme was implemented and evaluated to detect flooding attacks.

In **Chapter 6**, a novel indirect trust management mechanism was proposed to improve MANET performance in the presence of a black-hole attack. The performance of this mechanism was compared to that of the direct mechanism proposed in Chapter 4. The results showed that the indirect trust mechanism performed better than the direct trust mechanism in the presence of the black-hole attack, but at the cost of higher overload.

7.2 Re-positioning the research and its outcomes in the context of related work

This thesis provides a comprehensive exploration of Mobile Ad Hoc Networks (MANETs), focusing on their routing protocols, vulnerabilities, and the challenges they pose due to their distinctive characteristics compared to traditional networks. The investigation uncovers vulnerabilities within MANETs, which malicious nodes can exploit to detrimentally impact network performance. It establishes the necessity for tailored solutions to address the unique challenges inherent to MANETs.

In response to these challenges, this thesis introduces an approach centered around trust management as a promising avenue for enhancing MANET security and performance. Four direct trust management schemes are proposed and evaluated in the face of a spectrum of attacks, including black-hole, grey-hole, selfish, and flooding attacks. Moreover, an innovative indirect trust scheme is presented, specifically designed to counter black-hole attacks. A critical objective here is to compare the efficacy of both direct and indirect trust mechanisms in terms of throughput, Packet Delivery Ratio (PDR), and end-to-end delay.

This research contributes to the broader landscape of MANET security enhancement by not only identifying vulnerabilities and challenges but also by proposing and meticulously evaluating trust management mechanisms as a proactive defense strategy. The presented

schemes exhibit significant enhancements in MANET performance under the influence of the aforementioned attacks, albeit with a certain degree of associated overhead. It is important to note that the anticipated increase in overhead aligns with the inherent resource consumption expected when integrating additional tasks into routing protocols.

By embedding these findings within the context of related work, this thesis advances the understanding of trust management's role in fortifying MANETs against security threats. It builds upon prior research that has explored MANET vulnerabilities and contributes a practical evaluation of proposed mechanisms. This work serves as a valuable stepping stone for future research in the dynamic field of MANET security and performance optimization.

7.3 Strengths and limitations

As in any academic work, this thesis has some strengths and limitations. This thesis has taken years to accomplish, but there is always more that could be added. The strengths are as follows:

1. The study ran very many simulations based on an in-depth understanding of MANET routing protocol mechanisms, vulnerabilities, trust management principles and attack mechanisms.

2. The results presented in this thesis demonstrate that the principles of direct and indirect trust management can be used in MANETs to enhance performance when the network is under attack.

3. The thesis has proposed a direct trust management scheme that improved the throughput and PDR of the AODV routing protocol in the presence of black-hole, grey-hole and selfish attacks. However, the scheme did not show a clear improvement in the case of grey-hole attack.

4. The thesis has proposed a direct trust management scheme that improved the throughput and PDR of the AODV routing protocol in the presence of a flooding attack. This scheme is not the same as the one used to address the three other attack types because the flooding attack has a completely different mechanism.

5. The thesis has proposed a novel indirect trust management to enhance MANET performance in the presence of black-hole attacks.

6. The results related to node mobility and performance in the case of black-hole and selfish attack and the indirect trust management scheme have been published.

The limitations of the study are as follows:

1. More network scenarios could be used to evaluate the schemes. For example, the number of nodes could be increased to 500 or more and the number of malicious nodes could also be increased. The area dimensions and the transmission ranges of the nodes could be broadened to generate more results. Testing the schemes in a variety of scenarios should provide more accurate results.

2. Running the schemes and attacks with real-life MANETs rather than using the simulations would lead to more robust results. However, that might be expensive.

3. This thesis uses the standard models for the four attacks, but they can be more complicated and harder to detect. For example, a black-hole node can change its IP address regularly to avoid being blocked by other nodes when it is detected. These additional aspects are not included in the attack models in this thesis.

4. The effect of mobility was evaluated in three routing protocols, AODV, DSR and DSDV, but the direct and indirect trust schemes were implemented only in AODV. The AODV protocol was chosen because it is the most commonly used. The same schemes could be applied with some changes and evaluated with other routing protocols.

7.4 Future work

I consider this thesis a strong and solid foundation for undertaking future work in this field. This study covered four types of attack and employed two types of trust management scheme: direct and indirect. However, this work could be extended in many directions. Based on the contribution of this thesis, many potential ideas and improvements can be derived, such as the following:

1. More types of attack could be considered, such as snooping, hijacking and traffic analysis. Just as we modified the direct trust management scheme to address grey-hole attack, it would be possible to amend it to tackle other attacks once we understand how these attacks work.

2. The indirect trust management scheme could be evaluated in the presence of more attacks, not only black-hole. This would take more time and effort, but would be worth it. The results of this work introduce useful content for publication.

3. The same principles of direct and indirect trust management can be implemented with more routing protocols with some changes in the algorithms to fit each protocol. This could be achieved through separate projects, each focusing on one protocol.

References

[1] A. Alzahrani, H. Jari, and N. Thomas, "Analysing the effect of mobility on the performance of MANET routing protocols," *35th Annual UK Performance Engineering Workshop UKPEW*, pp. 1–11, 2019.

[2] A. Alzahrani, H. Jari, and N. Thomas, "Performance evaluation of MANET trust-based aodv protocol in the presence of blackhole attacks," *36th Annual UK Performance Engineering Workshop UKPEW*, pp. 30–40, 2020.

[3] A. Alzahrani, H. Jari, and N. Thomas, "A novel indirect trust mechanism for addressing black hole attacks in MANET," *DIVANet '21: In Proceedings of the 11th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 27–34, 2021.

[4] A. Alzahrani and N. Thomas, "Direct trust management mechanism to detect selfish attacks on MANET," *38th Annual UK Performance Engineering Workshop UKPEW*, pp. 12–19, 2022.

[5] C. P. Subir Kumar Sarkar, T.G. Basavaraju, *Ad hoc mobile wireless networks principles, protocols, and applications*, 2nd ed. New York: CRC Press, 2016.

[6] S. H. Wu, J. P. Sheu, and C. T. King, "Unilateral wakeup for mobile ad hoc networks with group mobility," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 2507–2517, 2013.

[7] R. R. Roy, *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Springer US, 2011.

[8] Z. Chen, W. Zhou, S. Wu, and L. Cheng, "An adaptive on-demand multipath routing protocol with QoS support for high-speed MANET," *IEEE Access*, vol. 8, pp. 44 760–44 773, 2020.

[9] S. Katiyar and S. Kumar, "An efficient topology management algorithm in MANETs," *In Proceedings of IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 267–272, 10 2018.

[10] L. Qin and T. Kunz, "Mobility metrics to enable adaptive routing in MANET," *2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 1–8, 2006.

[11] S. Misra and S. Goswami, "Network routing: Fundamentals, applications, and emerging technologies," 2017.

[12] K. Du and Y. Yang, "A QoS routing for maximum bandwidth in ad hoc networks," *2010 2nd International Conference on Future Networks*, pp. 343–345, 2010.

[13] H. Wang, Y. Wang, and J. Han, "A security architecture for tactical mobile ad hoc networks," *2009 2nd International Workshop on Knowledge Discovery and Data Mining*, pp. 312–315, 2009.

[14] A. Sadiqui, "Computer network security, first edition," *ISTE Ltd and John Wiley and Sons, Inc.*, pp. 1–14, 2020.

[15] Z. Ullah, M. H. Islam, and A. A. Khan, "Issues with trust management and trust based secure routing in MANET," *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 402–406, 3 2016.

[16] P. B. Karthik, H. R. Nagesh, and N. N. Chiplunkar, "Mitigation and performance evaluation mechanism for selfish node attack in MANETs," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 9 2018.

[17] F. Maan, Y. Abbas, and N. Mazhar, "Vulnerability assessment of AODV and SAODV routing protocols against network routing attacks and performance comparisons," *2011 Wireless Advanced*, pp. 36–41, 2011.

[18] S. C. Li, H. L. Yang, and Q. S. Zhu, "Research on MANET security architecture design," *2010 International Conference on Signal Acquisition and Processing (ICSAP)*, pp. 90–93, 2010.

[19] S. B. Sharma and N. Chauhan, "Security issues and their solutions in MANET," *2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management*, pp. 289–294, 7 2015.

[20] P. B. Velloso, R. P. Laufer, D. D. O. Cunha, O. C. M. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Transactions on Network and Service Management*, vol. 7, pp. 172–185, 9 2010.

[21] S. A. Thorat and P. J. Kulkarni, "Design issues in trust based routing for MANET," *2014 5th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 11 2014.

[22] M. Das, B. Sahu, and U. Bhanja, "Mobility and its effect on the performance of MANET," *2015 IEEE Power, Communication and Information Technology Conference, PCITC 2015 - Proceedings*, pp. 871–877, 3 2016.

[23] Alamsyah, E. Setijadi, I. K. E. Purnama, and M. H. Purnomo, "Analysis of reactive routing protocols in MANET based on quality of service," *In Proceedings of 2019 International Seminar on Application for Technology of Information and Communication: Industry 4.0: Retrospect, Prospect, and Challenges (iSemantic)*, pp. 342–345, 9 2019.

[24] Y. Bai, Y. Mai, and N. Wang, "Performance comparison and evaluation of the proactive and reactive routing protocols for MANETs," *Wireless Telecommunications Symposium*, 6 2017.

[25] A. M. Kanthe, D. Simunic, and R. Prasad, "Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks," *In Proceedings of 2012 1st International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN)*, 2012.

[26] G. R. M. Reddy and K. M, "Mobile ad hoc networks bio-inspired quality of service aware routing protocols," 2017.

[27] B. H. Khudayer, M. Anbar, S. M. Hanshi, and T. C. Wan, "Efficient route discovery and link failure detection mechanisms for source routing protocol in mobile ad-hoc networks," *IEEE Access*, vol. 8, pp. 24 019–24 032, 2020.

[28] T. P. Venkatesan, P. Rajakumar, and A. Pitchaikkannu, "Overview of proactive routing protocols in MANET," *In Proceedings of 2014 4th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 173–177, 2014.

[29] M. Appiah and R. Cudjoe, "A comparative study of reactive and proactive routing protocols on a mobility model in mobile ad hoc network MANET," *In 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, pp. 1–7, 2018.

[30] E. Setijadi, I. K. E. Purnama, and M. H. Purnomo, "Performance comparative of AODV, AOMDV and DSDV routing protocols in MANET using NS2," *2018 International Seminar on Application for Technology of Information and Communication*, 2018.

[31] U. Draz, T. Ali, S. Yasin, and A. Shaf, "Evaluation based analysis of packet delivery ratio for AODV and DSR under UDP and TCP environment," *2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development (iCoMET)*, vol. 2018-January, pp. 1–7, 4 2018.

[32] T. H. Sureshbhai, M. Mahajan, and M. K. Rai, "An investigational analysis of DSDV, AODV and DSR routing protocols in mobile ad hoc networks," *In Proceedings of 2nd International Conference on Intelligent Circuits and Systems (ICICS)*, pp. 286–289, 10 2018.

[33] A. A. Pirzada and C. McDonald, "Trust establishment in pure ad-hoc networks," *Wireless Personal Communications*, vol. 37, pp. 139–168, 4 2006.

[34] J. H. Cho, A. Swami, and I. R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 13, pp. 562–583, 12 2011.

[35] "Direct trust-based detection algorithm for preventing jellyfish attack in MANET," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 749–753, 11 2020.

[36] Y. Zhou, Z. Xue, C. Fu, J. Li, Y. Wu, Y. Yuan, Y. Zhu, and Y. Shi, "A new scheme for indirect trust calculation to resist bigmouth attack in wireless ad hoc network," *In Proceedings of 2013 16th IEEE International Conference on Computational Science and Engineering (CSE)*, pp. 248–251, 2013.

[37] A. Esaid and M. Agoyi, "Avoid suspicious route of blackhole nodes in MANETs: Using a cooperative trapping," *Computer Systems Science and Engineering*, vol. 45, pp. 1901–1915, 2023.

[38] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "A novel approach for grayhole and blackhole attacks in mobile ad hoc networks," *In Proceedings of 2012 2nd International Conference on Advanced Computing and Communication Technologies (ACCT)*, pp. 556–560, 2012.

[39] B. Sharma, "A distributed cooperative approach to detect gray hole attack in MANETs," *In Proceedings of the Third International Symposium on Women in Computing and Informatics*, pp. 560 – 563, 8 2015.

[40] S. V. Vasantha and A. Damodaram, "Bulwark AODV against black hole and gray hole attacks in MANET," *2015 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2015*, 3 2016.

[41] R. H. Jhaveri, "MR-AODV: A solution to mitigate blackhole and grayhole attacks in AODV based MANETs," *International Conference on Advanced Computing and Communication Technologies (ACCT)*, pp. 254–260, 2013.

[42] G. S. Bindra, A. Kapoor, A. Narang, and A. Agrawal, "Detection and removal of co-operative blackhole and grayhole attacks in MANETs," *In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET)*, 2012.

[43] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *In Proceedings of the 6th international conference on mobile computing and networking*, pp. 255–265, 2000.

[44] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)," *In Proceedings of the 3rd ACM International Symposium on Mobile Ad hoc Networking and Computing*, pp. 226–236, 2002.

[45] R. Michiardi, P. Molva, "A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," *IFIP TC6/TCII Sixth Joint Working Conference on Communication and Multimedia Security*, pp. 107–121, 2002.

[46] T. M. K. Pissinou, N. Ghosh, "Collaborative trust-based secure routing in multi hop ad hoc networks. networking," *In Networking 2004: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Third International IFIP-TC6 Networking Conference Athens, Greece, May 9–14, 2004, Proceedings 3*, vol. 3042, pp. 1446–145l, 2004.

[47] V. Balakrishnan, V. Varadharajan, P. Lucs, and U. K. Tupakula, "Trust enhanced secure mobile ad-hoc network routing," *In Proceedings of 2007 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINAW)*, vol. 1, pp. 27–33, 2007.

[48] C. Wang, X. Yang, and Y. Gao, "A routing protocol based on trust for manets," *Sixth Annual International Conference on Grid and Cooperative Computing*, vol. 3795, pp. 959–964, 2005.

[49] X. Li, Z. Jia, P. Zhang, and H. Wang, "A trust-based multipath routing framework for mobile ad hoc networks," *In Proceedings of 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 2, pp. 773–777, 2010.

[50] P. Yi, Z. Dai, Y. Zhong, and S. Zhang, "Resisting flooding attacks in ad hoc networks," *International Conference on Information Technology: Coding and Computing, (ITCC)*, vol. 2, pp. 657–662, 2005.

[51] Y. Ping, H. Yafei, Z. Shiyong, and D. Zhoulin, "Flooding attack and defence in ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 17, 2006.

[52] J.-H. Song, F. Hong, and Y. Zhang, "Effective filtering scheme against rreq flooding attack in mobile ad hoc networks," *Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*, vol. 3326, 2006.

[53] R. Venkataraman, M. Pushpalatha, R. Khemka, and T. R. Rao, "Prevention of flooding attacks in mobile ad hoc networks," *the International Conference on Advances in Computing, Communication and Control (ICAC3)*, p. 525–529, 2009.

[54] A. Bandyopadhyay, S. Vuppala, and P. Choudhury, "A simulation analysis of flooding attack in MANET using NS-3," *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2011*, 2011.

[55] H. Khattak and Nizamuddin, "A hybrid approach for preventing black and gray hole attacks in MANET," *2013 8th International Conference on Digital Information Management (ICDIM)*, pp. 55–57, 2013.

[56] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *In Proceedings of the IEEE*, vol. 98, pp. 1755–1772, 2010.

[57] S. Lee, B. Han, and M. Shin, "Robust routing in wireless ad hoc networks," *In Proceedings of the International Conference on Parallel Processing Workshops*, vol. 2002-January, pp. 73–78, 2002.

[58] M. Al-Shurman, S. M. Yoo, and S. Park, "Black hole attack in mobile ad hoc networks," *In Proceedings of the Annual Southeast Conference*, pp. 96–97, 2004.

[59] P. N. Raj and P. B. Swadas, "DPRAODV: A DYANAMIC learning system against blackhole attack in AODV based MANET," *IJCSI International Journal of Computer Science Issues*, vol. 2, 2009.

[60] N. Mistry, D. C. Jinwala, and M. Zaveri, "Improving aodv protocol against blackhole attacks," *International MultiConference of Engineers and Computer Scientists (IMECS)*, p. 1–5, 3 2010.

[61] D. Kumar, *Research methods for successful PhD*. River Publishers, 2017.

[62] T. Issariyakul and E. Hossain, "Introduction to network simulator NS2," 2012.

[63] L. E. Quispe and L. M. Galan, "Assessment of throughput performance under NS2 in mobile ad hoc networks MANETs," *In Proceedings of 2013 5th International Conference on Computational Intelligence, Communication Systems, and Networks (CICSyN)*, pp. 338–343, 2013.

[64] K. Zaman and S. M. Khan, "Construction of confidence interval on mean value with interval data," *In Proceedings of the 2013 IEEE Symposium on Computational Intelligence for Engineering Solutions, CIES 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, pp. 157–162, 2013.

[65] D. G. Zhang, P. Z. Zhao, Y. Y. Cui, L. Chen, T. Zhang, and H. Wu, "A new method of mobile ad hoc network routing based on greed forwarding improvement strategy," *IEEE Access*, vol. 7, pp. 158 514–158 524, 2019.

[66] N. Thomas and M. Sokolowski, "The effect of mobility on connectivity in MANETs," 2016. [Online]. Available: https://www.researchgate.net/publication/307866668

[67] V. Anand and S. C. Gupta, "Performance of AODV, DSR and DSDV protocols under varying node movement," *2012 World Congress on Information and Communication Technologies (WICT)*, pp. 50–55, 2012.

[68] R. K. Gujral, J. Grover, Anjali, and S. Rana, "Impact of transmission range and mobility on routing protocols over ad hoc networks," *In Proceedings: Turing 100 - International Conference on Computing Sciences, ICCS 2012*, pp. 201–206, 2012.

[69] S. Gowrishankar, S. K. Sarkar, and T. G. Basavaraju, "Simulation based performance comparison of community model, GFMM, RPGM, manhattan model and RWP-SS mobility models in MANET," *2009 1st International Conference on Networks and Communications (NetCoM)*, pp. 408–413, 2009.

[70] P. Santi, *Mobility models for next generation wirless networks: ad hoc, vehicular and mesh networks*. John Wiley and Sons Ltd, 2012.

[71] R. M. Fujimoto, G. F. Riley, and K. S. Perumalla, *Network Simulation*. Springer International Publishing, 2007. [Online]. Available: https://link.springer.com/10.1007/978-3-031-79977-8

[72] T. A. Nitnaware, D., "Black hole attack detection and prevention strategy in DYMO for MANET," *3rd International Conference on Signal Processing and Integrated Networks*, 2016.

[73] G. S. Bindra, A. Kapoor, A. Narang, and A. Agrawal, "Detection and removal of co-operative blackhole and grayhole attacks in MANETs," *In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET)*, 2012.

[74] C. Perkins, E. Belding-Royer, and S. Das, "RFC3561: Ad hoc on-demand distance vector AODV routing," *RFC Editor*, pp. 1–37, 2003.

[75] A. Tripathi and A. K. Mohapatra, "Mitigation of blackhole attack in MANET," *In Proceedings of 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 437–441, 10 2017.

[76] G. Tan, Q. Zhang, L. Zhang, and Y. Li, "Characterizing the interference distribution in MANETs with different mobile models," *In Proceedings of 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 141–144, 9 2016.

[77] S. Chiyangwa and M. Kwiatkowska, "A timing analysis of AODV," *In Formal Methods for Open Object-Based Distributed Systems: 7th IFIP WG 6.1 International Conference, FMOODS 2005, Athens, Greece, June 15-17, 2005. Proceedings 7*, p. 306–321, 2005.

[78] P. Rani, Kavita, S. Verma, and G. N. Nguyen, "Mitigation of black hole and gray hole attack using swarm inspired algorithm with artificial neural network," *IEEE Access*, vol. 8, pp. 121 755–121 764, 2020.

[79] A. U. Khan, M. D. Chawhan, M. M. Mushrif, and B. Neole, "Performance analysis of adhoc on-demand distance vector protocol under the influence of black-hole, gray-hole and worm-hole attacks in mobile adhoc network," *In Proceedings of 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 238–243, 5 2021.

[80] M. J. Raju and P. Subbaiah, "An adaptive low overhead routing scheme with priority function for MANETs," *2013 15th International Conference on Advanced Computing Technologies, ICACT 2013*, 2013.

[81] J. P. Ghonge, M., "Selfish attack detection in mobile ad hoc networks," *International Conference on innovations in Information, Embedded and Communication System*, pp. 1–4, 2017.

[82] M. M. Ghonge, P. M. Jawandhiya, and V. M. Thakare, "Reputation and trust based selfish node detection system in MANETs," *In Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, pp. 661–667, 6 2018.

[83] M. M. Ghonge, S. Sant, G. Baba, P. M. Jawandhiya, and V. M. T. Head, "Selfish attack detection in mobile ad hoc networks," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017.

[84] P. Sankareswary, R. Suganthi, and G. Sumathi, "Impact of selfish nodes in multicast ad hoc on demand distance vector protocol," *2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, 2010.

[85] J. Liu, M. Sheng, Y. Xu, J. Li, and X. Jiang, "End-to-End delay modeling in buffer-limited MANETs: A general theoretical framework," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 498–511, 1 2016.

[86] M. Patel, S. Sharma, and D. Sharan, "Detection and prevention of flooding attack using SVM," *In Proceedings of 2013 International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 533–537, 2013.

[87] P. Choudhury, S. Nandi, A. Pal, and N. C. Debnath, "Mitigating route request flooding attack in MANET using node reputation," *IEEE International Conference on Industrial Informatics (INDIN)*, pp. 1010–1015, 2012.

[88] D. Pengfule, T. Zhihong, Z. Hongli, W. Yong, Z. Liang, and G. Sanchuan, "Detection and defense of syn flood attacks based on dual stack network firewall," *In Proceedings of 2016 IEEE 1st International Conference on Data Science in Cyberspace (DSC)*, pp. 526–531, 2 2017.

[89] H. Moudni, M. Er-Rouidi, H. Mouncif, and B. E. Hadadi, "Performance analysis of AODV routing protocol in MANET under the influence of routing attacks," *In Proceedings of 2016 International Conference on Electrical and Information Technologies (ICEIT)*, pp. 536–542, 7 2016.

[90] V. K. J. Gurjinder Kaur and Y. Chaba, "Prevention of flooding attacks in mobile ad hoc networks," *2nd International Conference on Wireless Intelligent and Distributed Environment for Communication WIDECOM 2019*, vol. 27, pp. 193–201, 2019.

[91] M. Rmayti, Y. Begriche, R. Khatoun, L. Khoukhi, and D. Gaiti, "Flooding attacks detection in MANETs," *In Proceeding of 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*, 9 2015.

[92] M. Rao and N. Singh, "Quality of service enhancement in MANETs with an efficient routing algorithm," *Souvenir of the 2014 IEEE International Advance Computing Conference (IACC)*, pp. 381–384, 2014.

[93] R. J. W. Fargo, "Application of indirect trust computation in MANET," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, pp. 2319–5940, 2014. [Online]. Available: https://www.researchgate.net/publication/262672712

[94] R. L. Gordon and D. S. Dawoud, "Direct and indirect trust establishment in ad hoc networks by certificate distribution and verification," *In Proceedings of the 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2009*, pp. 624–629, 2009.

[95] J. Huang, Z. Y. Sun, H. J. Zhang, J. Chen, and S. He, "An evaluation management mechanism based on node trust," *2021 International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, pp. 189–193, 2021.

Appendix A

Software Engineering Modifications for Implementing TAODV in NS2

The Network Simulators NS2 and NS3 provide a foundation for simulating network behaviors and protocols. To implement the Trust-based Ad hoc On-Demand Distance Vector (TAODV) protocol and evaluate its performance against black-hole, grey-hole, and selfish attacks, specific software engineering modifications were made within NS2 and NS3 to support the new protocol and its assessment.

A.1 Identifying Key Simulator Components:

1. Packet Structure Extension: The existing packet structure within NS2 and NS3 was extended to accommodate the inclusion of trust values and packet tracking information.

2. Routing Protocol Modules: Duplications of AODV routing modules were created to develop the TAODV protocol. Additional functions were added to support trust management.

3. Trust Scheme Functions: Six functions, namely insert, lookup, delete, update, flush, and count, were added to manage trust values within the proposed scheme.

4. Timer Variable Integration: A timer variable was introduced to track the time that the source node should wait before sending the next test packet.

5. Simulation of the black-hole, grey-hole, and selfish Attacks: A mechanism was created to simulate the attacks by designating certain nodes as malicious, causing intentional packet drops.

A.2 Actual Modifications Carried Out:

1. Packet Structure Extension: The packet structure in NS2 and NS3 was extended to include fields for trust values and packet tracking information. These fields facilitated the exchange and processing of trust-related data.

2. Routing Protocol Modules: The AODV routing module was duplicated and customized to form the TAODV module. Logic was implemented to support trust-based decision-making and management of trust-related functions.

3. Trust Scheme Functions: Six trust scheme functions (insert, lookup, delete, update, flush, count) were developed and integrated into the TAODV module. These functions enabled the management of trust values and packets list.

4. Timer Variable Integration: A timer variable was incorporated into the protocol to determine the time duration that the source node should wait before sending the next test packet, using NET_TRAVERSAL_TIME from the AODV protocol.

5. Simulation of the black-hole and grey-hole Attacks: A mechanism was designed to mark certain nodes as malicious nodes, resulting in deliberate packet drops. This facilitated the evaluation of the TAODV protocol's performance under attack scenarios.

A.3 Testing and Validation:

1. Simulation Setup: A simulation scenario was defined to evaluate both AODV and TAODV protocols in the presence of a black-hole attack. Network parameters, node placement, and a mobility pattern were configured accordingly.

2. Performance Metrics Definition: Key performance metrics, such as packet delivery ratio, end-to-end delay, and network throughput, were identified to quantitatively assess the protocols.

3. Simulation Execution: The simulation was conducted for both AODV and TAODV protocols with the black-hole attack scenario enabled. Performance data was collected during simulation runs.

4. Data Analysis: Simulation results were analyzed to compare the performance of AODV and TAODV under black-hole, grey-hole, and selfish attacks conditions. The impact of trust-based decision-making on network response to attacks was evaluated.

5. Validation: The improved performance demonstrated by TAODV, including higher packet delivery ratios and throughput compared to AODV, validated the efficacy of integrating trust management against black-hole, grey-hole, and selfish attacks.

Appendix B

Software Engineering Modifications for Supporting TAODV and the flooding attack in NS3

The proposed scheme aims to bolster Mobile Ad hoc Network (MANET) performance during flooding attacks. Rather than solely relying on the honesty of source nodes, this scheme empowers intermediate nodes to scrutinize parameters like TTL and RREQ_RETRIES. If these parameters are suspect, TTL exceeding TTL_THRESHOLD or RREQ_RETRIES surpassing 2/s—the source node gets assigned a trust value of zero. Consequently, the intermediate node gains the ability to detect and isolate flooding attackers.

B.1 Actual Modifications Carried Out:

1. Table for Storing Information: An additional table was seamlessly integrated into each node, serving as a repository for crucial information. This table included fields such as the IP address of the source node, the number of received RREQs, and the trust value attributed to the node.

2. Algorithm Integration: The algorithm, as elucidated in the scheme, was diligently implemented within the AODV routing module. This necessitated the meticulous inclusion of the prescribed checks and actions. The module was fortified with the capacity to evaluate TTL values, monitor received RREQs, and judiciously assign trust values in adherence to the proposed criteria. (may paraphrase)

3. Handling RREQs: The inherent logic governing the AODV protocol's management of incoming RREQ packets was meticulously enhanced. The revised protocol now diligently applies the stipulated checks and measures as detailed in the proposed

scheme. It dynamically assesses the veracity of TTL and RREQ_RETRIES values for each received RREQ packet.

4. Trust Value Management: An ingenious framework was engineered to dynamically manage trust values based on the outcomes of the comprehensive RREQ assessments. The intricate logic was executed to accurately adjust trust values according to the defined guidelines, effectively characterizing the integrity of the source nodes.

5. Table Reset: The elegant implementation included an ingenious mechanism to seamlessly reset the cumulative count of RREQs received from each individual source node. This novel feature automatically triggers every second, successfully aligning with the scheme's stipulations.

B.2 Testing and Validation:

1. Configuring the Simulation: A comprehensive simulation scenario was meticulously devised within the NS3 platform to accurately emulate real-world conditions.

2. Defining Performance Metrics: The criteria for evaluating the scheme's effectiveness were thoughtfully defined. This encompassed metrics such as packet delivery ratio, network throughput, and end-to-end delay.

3. Running the Simulation: The NS3 simulator was utilized to execute the simulation scenarios while actively integrating the proposed scheme. This allowed for the observation of its performance under varying flooding attack scenarios.

4. Analyzing and Interpreting Data: The results obtained from the simulations were analyzed. A comparison between the performance of the AODV protocol with and without the scheme shed light on its efficacy in battling flooding attacks.