Trust Management and Security Analysis of Opportunistic Routing Protocols in Multihop Wireless Networks

by

Mahmood Salehi

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Routing in wireless sensor, mobile ad-hoc, and vehicular networks is a crucial operation having significantly attracted scientists' attention during the past years. Opportunistic routing, however, is a novel and promising approach, which is still an ongoing research paradigm. The purpose of opportunistic routing protocols is to increase the reliability of delivering data packets to their destination by utilizing the broadcast nature of wireless networks and selecting a set of nodes, instead of only one, as potential next-hop candidates. Apart from the reliability, security and trustworthiness of communications is also a challenging task. More specifically, almost all traditional and opportunistic routing protocols require cooperation of all network nodes to complete the routing process. However, some nodes in the network may be compromised and avoid collaboration with others due to various selfish or malicious motivations.

The focus of this thesis is on modeling security challenges and developing trust-based opportunistic routing protocols. This way, communication between nodes will not only benefit from the reliability of opportunistic routing methods, but also from the security of trust and reputation management schemes. For this purpose, a novel trust-based opportunistic routing protocol is proposed that introduces three different candidate selection metrics known as RTOR, TORDP, and GEOTOR. Such metrics have been designed to address specific characteristics of opportunistic routing protocols. In continue, a watchdog mechanism is proposed to assist nodes in monitoring their candidates, recalculating their trust value, and finally replacing malicious candidates with benign ones. Afterwards, an analytical approach is introduced using Discrete-Time Markov Chain to demonstrate the effect of malicious nodes on different parameters of a wireless network that uses opportunistic routing. In this model, a new method of calculating packet drop ratio is introduced to represent the effect of attackers. The model is then applied on different well-known opportunistic routing protocols and results of the model are verified by simulation. Finally, the proposed analytical model is extended to include a defensive mechanism against adversary nodes. In fact, a packet salvaging mechanism is developed through which backup candidates attempt to save some of the maliciously dropped data packets. Different related network parameters such as delivery ratio, salvage ratio, direct-delivery ratio, etc. are then introduced, calculated, and reported using the introduced model.

Publications

The following publications by the author are relevant to this thesis:

Journals

- Mahmood Salehi, Azzedine Boukerche. *Secure Opportunistic Routing Protocols: Challenges, Classification, and Comparison.* Wireless Networks, Submitted, 2017
- Mahmood Salehi, Azzedine Boukerche. *A Novel Packet Salvaging Model to Improve the Security of Opportunistic Routing Protocols.* Computer Networks, Under the second round of review, 2017
- Mahmood Salehi, Azzedine Boukerche, Amir Darehshoorzadeh. *Modeling and performance evaluation of security attacks on opportunistic routing protocols for multihop wireless networks.* Ad Hoc Networks, vol 50, Pages 88-101, 2016
- Mahmood Salehi, Azzedine Boukerche, Amir Darehshoorzadeh, Abdelhamid Mammeri: *Towards a novel trust-based opportunistic routing protocol for wireless networks.* Wireless Networks, vol 22(3), Pages 927-943, 2016

Conferences

- Mahmood Salehi and Azzedine Boukerche. *Pasor: A packet salvaging model for opportunistic routing protocols.* Accepted in 2016 IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), IEEE, Brasilia, Brazil, Pages 1-8, 2016.
- Mahmood Salehi, Amir Darehshoorzadeh, Azzedine Boukerche. *On the Effect of Black-hole Attack on Opportunistic Routing Protocols.* In Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN âĂŹ15, ACM, New York, NY, USA, Pages 93-100, 2015.
- Mahmood Salehi, Azzedine Boukerche: *A Comprehensive Reputation System to Improve the Security of Opportunistic Routing Protocols in Wireless Networks.* In 2015 IEEE Global Communications Conference (GLOBECOM), IEEE, San Diego, USA, Pages 1-6, 2015.

- Mahmood Salehi and Azzedine Boukerche. *Trust-aware opportunistic routing protocol for wireless networks.* In Proceedings of the 10th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet âĂŹ14, ACM, New York, NY, USA, 2014, Pages 79-86, 2014.

Acknowledgements

I would like to sincerely appreciate my supervisor, Professor Azzedine Boukerche, for providing advice, life lessons, encouragement, and financial support during the entire period of my Ph.D. program. I am proud that I have studied and worked under his professional supervision and guidance.

I truly thank my friends and colleagues at the PARADISE research laboratory, University of Ottawa. They always assisted whenever I asked for help, supporting me in achieving a productive Ph.D. work. Special thanks to Dr. Amir Darehshoorzadeh and Dr. Robson Eduardo De Grande for their willingness to share thoughts.

I am incredibly thankful to my family for their lifetime helps. They have always assisted me with whatever they could in this long journey. Last but not least, I deeply appreciate and owe my gratitude to my dear wife Sanaz Ahmadvand for her patience and precious love. I would have not been able to make it this far without her continuous encouragement and support.

Glossary of Terms

CC: Candidate Coordination
CS: Candidate Selection
Dest: Destination
DP: Distance Progress
DTMC: Discrete-Time Markov Chain
EAX: Expected Any-path Transmission
EDP: Expected Distance Progress
ETX: Expected Number of Transmissions
GEOTOR: Geographical Trust-based Opportunistic Routing
ID: Identifier
MANET: Mobile Ad-hoc Network
OR: Opportunistic Routing
ReTx: Retransmission
RTOR: Reliable Trust-based Opportunistic Routing
Src: Source
TORDP: Trust-based Opportunistic Routing Metric with Minimum Distance Progress
VANET: Vehicular Ad-hoc Network
WMN: Wireless Mesh Network
WSN: Wireless Sensor Network

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Opportunistic Routing

Research and technology in different aspects of wireless networks have been significantly progressing in the past few years. Hop by hop routing as well as security challenges, amongst others, have considerably drawn the attention of researchers. To be more specific, reliable routing is a challenging process and ensuring the reliability of end-to-end packet delivery is of high importance. *Opportunistic Routing* (OR) protocols are designed to address such a reliability requirement. As discussed in [25] and [90], the main purpose of OR protocols is to more effectively use the broadcast nature of packet transmissions in wireless networks. In fact, upon transmission of a packet in wireless networks through a specific node, all neighboring nodes that are geographically located in the transmission radius of the sending node may receive a copy of this packet. In contrast to traditional routing schemes such as DSR, AODV, OLSR, DSDV, etc., [50, 133, 66, 26, 18, 94, 20, 21, 19] which suggest selecting only one node in each hop of the routing process to operate as the actual next-hop forwarder and to assist in transmitting the packet towards its destination, OR methods select a subset of neighbors known as the *candidate set* as potential next hop forwarders. The candidate set of each node is selected using a candidate selection algorithm, and different aspects are considered for candidate selection to increase the likelihood of delivering packets to their final destination. Amongst the nodes in the candidate set, however, one node will act as the actual next hop forwarder. This node transmits the packet one hop further on its way towards the destination. The actual next hop node is determined by performing a *candidate coordination* mechanism

between nodes in the candidate set. Upon transmission of the packet, such a candidate should wait to receive an acknowledgment from one of its own candidates to ensure that the packet has been successfully forwarded; otherwise, it tends to retransmit the packet for the maximum number of a predetermined repetitions. This occurs for the purpose of increasing the probability of delivering each packet to its destination. As mentioned in [90], most of the research in the area of OR is focused on different candidate selection and coordination methods.

1.2 Secure Routing in Wireless Networks

Although a tremendous amount of research has been conducted in the interest of various candidate selection and coordination methods for OR protocols [25], the research in this area is still ongoing in terms of security and trustworthy aspects. To be more specific, almost all of the well-known proposed OR and traditional routing protocols operate following a strong assumption that all nodes in the network are benign and cooperative nodes. In real world situations, however, the scenario might be absolutely different, and nodes may not behave as expected when it comes to cooperation in network operations [33, 38]. This adversary behavior occurs as a result of varying malicious or selfish intentions and can have devastating effects on the performance of communications in wireless networks. For example, in *black-hole* attack [111] which is categorized as a Denial of Service (DOS) attack, malicious nodes seek selection by other nodes in the network as next-hop forwarders, and attract as many data packets as possible. Instead of forwarding, however, such nodes will tend to discard all of the received packets and decrease network performance. [106] demonstrated how devastating black-hole nodes can be in mobile ad-hoc networks [65]. Therefore, a great deal of research has been performed in the literature to assist in forwarding and routing packets more sophisticatedly in a hostile environment. The main purpose of such algorithms is to improve the security of communications [35, 29]. Trust and reputation management protocols are a subset of security mechanisms having been developed for secure packet forwarding [46]. The general idea behind trust management systems in wireless networks is to enable benign nodes to detect uncooperative individuals and to prevent from communicating with such malicious nodes once they are recognized [48, 16, 104, 34, 47].

1.3 Thesis Problem Statement

Although trust and security of traditional routing protocols have been considerably studied in recent years, to the best or our knowledge, the effects of malicious nodes in OR protocols and defensive mechanisms against such nodes have not been significantly studied and investigated. This fact motivated us to direct our research towards proposing protocols that investigate, evaluate, and defend against malicious nodes in wireless networks, when OR protocols are applied to route data packets. Our research in this area falls into three different categories as follows.

- **Proposing a trust model to defend against malicious nodes:** In this phase, we are going to propose a novel trust model which can be applied in OR protocols. Having done so, the proposed protocol will not only benefit from the higher reliability of OR protocols, but also from the security of trust models. More specifically we will design and propose a trust-based OR protocol, which enables benign nodes to recognize malicious nodes, and to isolate them in the network accordingly. Some contributions of the proposed trust model are as follows.
  - Three novel metrics for candidate selection are proposed considering various parameters such as the link delivery probability between nodes, their trust value, and geographical location. Each metric attempts to select next hop candidates in a different way, according to different application requirements.
  - A trust-based OR protocol is proposed that selects candidate set using any of the proposed metrics. By applying this model, nodes will be able to create their routing tables and to forward incoming data packets to the set of candidates they have selected for each destination. The set of candidates in each node for a particular destination may change in the course of time, according to the behavior of such candidates.
  - A watchdog mechanism is proposed and customized for OR protocols. The introduced watchdog mechanism assists in monitoring the behavior of candidate nodes more realistically. Moreover, it feeds the trust model by providing information about the behavior of each candidate. This way, each node will be able to decide whether to reward a candidate node for its cooperation, or punish it for its adversary behavior. Punishment of malicious nodes will downgrade the metric value that benevolent nodes compute for such nodes. This

means malicious nodes will lose their chance of being selected as candidate nodes in the next round of candidate selection.

- Finally, an extensive set of simulations is designed, implemented, and executed to evaluate the performance of the proposed trust model.

- **Introducing an analytical model to demonstrate the effects of malicious nodes:** In this step, a novel analytical model is introduced to represent the behavior of malicious nodes in OR-based wireless networks, where candidates follow perfect coordination. As explained in [25] and in Chapter 2 Section 2.1, perfect coordination implies a situation in which data packet forwarding through the highest-priority candidate that has received the packet results in notification of this transmission to all other candidates, including the previous-hop; this prevents them from forwarding the same packet again. More specifically, a new model to represent an OR-based wireless network is introduced using Discrete-Time Markov Chains (DTMC), where malicious nodes exist in the system and prevent data packet forwarding. This model is considered a comprehensive method for investigating the behavior of adversary nodes, when such nodes probabilistically prevent from collaboration with benign nodes. In this model, a novel method and model for calculating the packet drop ratio is proposed. This makes it possible to more accurately measure the effects of malicious nodes on data packet drop rates. Furthermore, the model is applied to different well-known OR protocols. Thus, it becomes possible to study the behavior of different protocols using the proposed analytical model in presence of malicious nodes. Finally, a comprehensive set of evaluations is performed and studied creating the possibility of investigating the proposed model more extensively under different network conditions. Some of the most important contributions of this proposed model can be listed as follows.

  - First, a novel model to represent the OR-based wireless mesh network is introduced using Discrete-Time Markov Chain (DTMC) for use where malicious and uncooperative nodes exist in the system and prevent data packet forwarding. A set of scenarios and applications is also provided to indicate how the proposed model can be applied.
  - Second, using the proposed model, a new approach for the calculation of *packet drop ratio* parameter is introduced. Drop ratio assists in measuring the effects

of uncooperative nodes in an OR-based wireless mesh network.

- Third, a novel version of the black-hole attack is introduced, in which the malicious node receives a data packet from a previous hop, drops the packet, and pretends that it has already forwarded the packet so that other candidates do not repeat the transmission.
- Fourth, the effects of the proposed model are evaluated by performing a comprehensive set of performance evaluation tests that consider different network parameters, using both analytical and simulation results. The evaluation is applied and studied on four well-known OR protocols.

- **Proposing a packet salvaging analytical model:** In this phase, we propose an extended version of the proposed analytical model to salvage a proportion of the packets that are dropped by malicious nodes. The introduced model enables nodes in the candidate set to collaborate with each other with greater sophistication, and to save some of the packets that are maliciously dropped or manipulated by attackers. Furthermore, in addition to the proposal of a novel approach to calculate different network parameters including *delivery ratio*, *drop ratio*, *expected number of transmissions (ETX)*, and *hop count* specifically for the proposed model, two new parameters are introduced, known as *direct-delivery ratio* and *salvage ratio*. A comprehensive set of performance evaluation experiments is also conducted, reported, and analyzed. Some of the most important contributions of this model are as follows.

  - A novel DTMC-based packet salvaging model is proposed, designed, and implemented, enabling candidate nodes to save some of the packets dropped by malicious nodes.
  - Using the proposed model, a new approach is introduced that calculates different network parameters such as the probability of packet delivery, drop ratio, expected number of transmissions (ETX), and hop count.
  - Two novel parameters are introduced and calculated from the proposed analytical model, known as packet salvage ratio and direct delivery ratio.
  - The proposed model is applied to a well-known OR protocol, using the proposed analytical approach, as well as the use of network simulation. Finally,

a comprehensive set of evaluations is conducted considering different parameters, and evaluation results are reported.

1.4 Challenges in Secure and Reliable Opportunistic Routing

As mentioned in Section 1.1, and 1.2, OR protocols investigate reliable routing of packets between the source and the destination based upon different aspects such as quality of links between nodes, geographical location of them, and so forth. On the other hand, ensuring security and integrity of delivering packets is significant. Trust and reputation management protocols are designed to address such a security requirement. In order to investigate and benefit from both the reliability of OR methods as well as the security of trust management schemes, a set of protocols is proposed in three different phases as described in Section 1.3. In each phase, however, we are going to tackle various challenges and obstacles. Some of the most important challenges with regards to the proposed protocols are described following in this section.

- **Modeling OR protocols containing malicious nodes:** This step focuses on developing an analytical model to represent the behavior of OR protocols in presence of malicious nodes. We propose to use Discrete-Time Markov Chain (DTMC) as a modeling tool. For this purpose, it is a challenging task to demonstrate the state machine of the system according to the behavior of OR protocols, to represent malicious nodes in the system, and to extract different parameters of network such as packet delivery ratio, drop ratio, end-to-end delay, and so forth.

- **Packet salvaging in presence of malicious nodes:** One of the interesting characteristics of OR protocols, unlike traditional routing approaches, is that routing operations depends upon a group of nodes known as the candidate set. In this scenario, candidates are able to monitor each other and attempt to transmit a packet, if malicious candidates are not collaborating properly. Developing a model which is accurate and applicable in real-world applications is a challenging task. More specifically, the model should be sophisticated enough to recognize malicious nodes properly, progress packets towards their destination, and finally, prevent from wasting network bandwidth and other resources. Furthermore, introducing

different relevant network parameters and extracting them from the model is a significant challenge to tackle.

- **Developing and modeling a trust management scheme specifically for OR protocols:** Trust management schemes are designed to recognize and to defend against malicious nodes in wireless networks. However, employing trust management protocols for OR protocols is a very challenging task. More specifically, the trust value that nodes calculate for each other can play a significant role in the performance of OR protocols. This value can be utilized when selecting candidates or at the time of candidate coordination. On the other hand, the proposed trust model should be designed considering specific characteristics of OR protocols, when evaluating the behavior of other nodes and calculating their trust value.

- **Designing a proper monitoring module specifically for OR:** OR protocols are proposed to increase the reliability of communications. For this purpose, and in order to represent how reliable different protocols are, OR protocols are usually evaluated in an unreliable environment, where wireless links between nodes suffer from the noise in the channel. This assumption is realistic according to the real-world applications and scenarios. Furthermore, trust management protocols generally benefit from a monitoring module that is responsible to supervise the behavor of nodes in the network and to determine if they are cooperative or malicious nodes. Designing an efficient monitoring module for OR protocols, that considers the behavoir of nodes in a noisy wireless channel is a challenging task that should be dealt with.

- **Generalizing the proposed scheme to be applied in any wireless or mobile environment:** After modeling the proposed trust management approach for wireless sensor or mesh networks, the idea is to generalize such a method to be applied in mobile ad-hoc and vehicular networks. Designing a general model for mobile networks is a challenging task. This is due to the fact that mobility of nodes leads to continuous changes in the quality of links between nodes, and introduces a higher level of complexity to the proposed models.

- **Considering scalability of proposed schemes:** Since proposed protocols are going to be comprehensive methods applicable in any wireless network, the scalability becomes an important challenge. After designing and implementing each model,

different aspects of scalability considerations need to be studied and investigated for introduced models.

- **Considering energy efficiency of proposed protocols:** In wireless networks, specially wireless sensor networks, nodes have limited energy and processing power. This means any algorithm proposed for such networks should take into account energy constraints. Therefore, energy consumption aspects of proposed protocols are to be researched and investigated compared to the existing works in the literature.

1.5 Thesis Organization

The structure of this thesis is organized as follows. Related works in both areas of OR protocols and trust management protocols are studied in Chapter 2. Chapter 3 contains the proposed OR-based trust model. Chapter 4 includes the proposed analytical approach to model security challenges of OR protocols. In Chapter 5, the proposed packet salvaging protocol is thoroughly explained and elaborated. Finally, Chapter 6 concludes the thesis and introduces some future works.

Chapter 2

Related Works

As mentioned in Chapter 1, this thesis proposes novel models and protocols for investigating different security aspects of OR-based wireless networks. Therefore, this is of significant importance to provide an overview of OR protocols and elaborate the state of the arts in areas of OR protocols, as well as secure routing in wireless networks. In the following subsections, some of the most important research findings in the mentioned areas are briefly investigated.

2.1 An Overview of Opportunistic Routing

OR protocols operate in three phases: candidate selection, coordination, and performing retransmissions. Various OR protocols have been proposed in the literature, considering different candidate selection or coordination mechanisms. Furthermore, such protocols can be used in multicast or unicast applications, and take advantage of a single or multiple channel rates. Sections 2.1.1, 2.1.2, and 2.1.3 provide a brief overview of different challenges regarding OR, while Figure 2.1 visualizes different aspects of OR protocols considered in this thesis.

2.1.1 Operational Phases of OR

This section provides an overview of different operational steps for OR protocols. Furthermore, various methods of performing each step is explained and investigated in greater details.

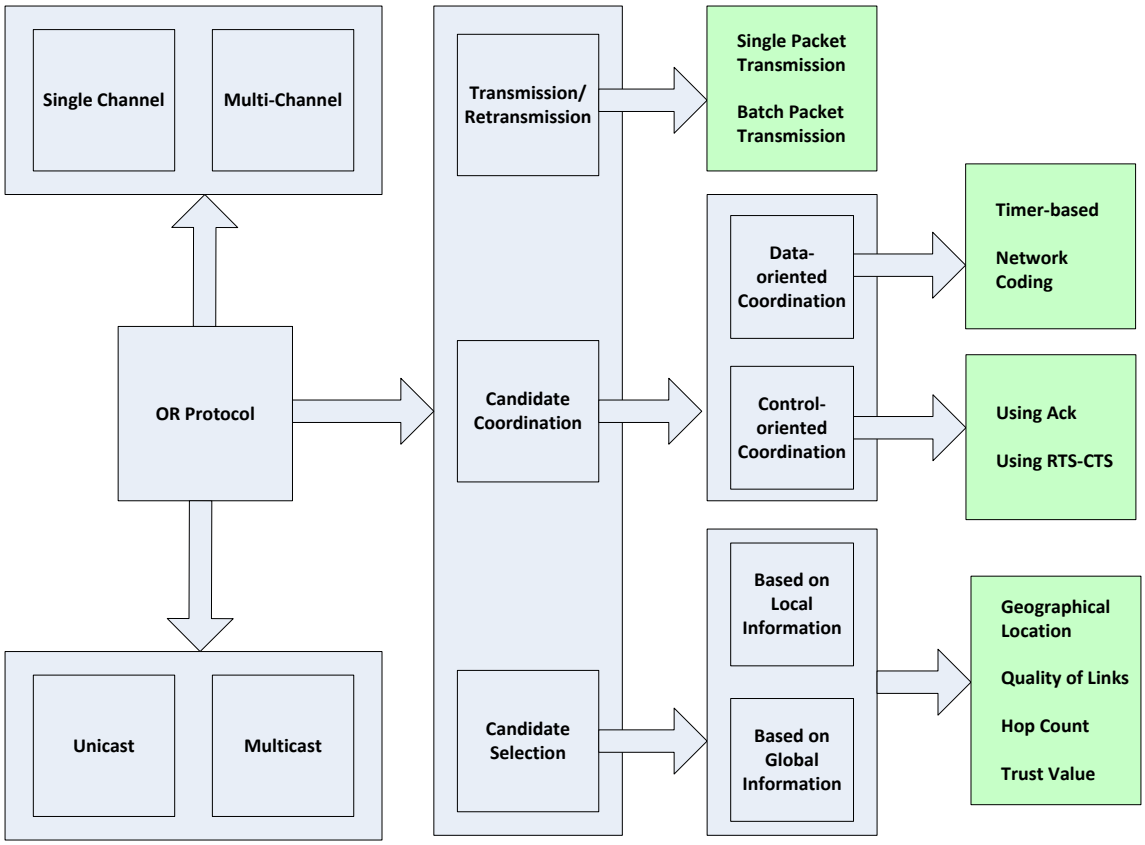

Figure 2.1: An schema of different aspects of OR protocols

Selecting the Set of Candidates

Candidate selection is the most important step in OR. During this functional step, nodes run a candidate selection algorithm and choose a set of nodes as their candidates. Candidate nodes are responsible for receiving packets from the sender node, and forwarding them towards their destination. Candidate selection algorithms, however, use a metric to select candidates and prioritize them accordingly. Candidate priority specifies which should take action if multiple candidates receive a copy of the same packet. As shown in Figure 2.1, two categories of metrics are known for candidate selection. The first category selects candidates using local information, while the second one benefits from global information. In fact, global candidate selection algorithms operate utilizing topological

information such as the quality of links between nodes. In contrast, local algorithms use geographical information of neighboring nodes. For instance, Expected Transmission Count (ETX) and Expected Any-path Transmission (EAX) are topology-based metrics that are calculated based on the quality of links, while Distance Progress (DP) and Expected Distance Progress (EDP) are local metrics that are extracted using the geographical location of neighboring nodes. EXOR [12] used ETX metric, MTS [88] proposed candidate selection using EAX, POR [127] utilized DP, and DPOR [60] used EDP for candidate selection.

Performing a Coordination Mechanism

Candidate coordination is the second operational phase in OR protocols. Although the main objective of OR is to increase the throughput and the reliability of routing, control of the propagation of duplicate packets and the use of network bandwidth is important as well. Candidate coordination mechanisms are designed to address the problem of transmitting duplicate packets. In OR protocols, two categories of candidate coordination approaches have been proposed, known as data-oriented and control-oriented coordination algorithms [25]. An efficient coordination algorithm should attempt to choose the best candidate as the relay node, and should properly notify other candidates to prevent transmitting the same data packet. In control-oriented methods, candidates exchange control messages between each other when receiving or forwarding a data packet. However, data-oriented approaches perform the coordination process indirectly, with the use of information propagated in the data packets.

In control-based coordination, candidates broadcast an acknowledgment (ACK) message once they receive a data packet to transmit. Propagation of ACK messages occurs based upon the priority of nodes in the candidate set. Therefore, the previous-hop node can determine which candidate can act as the actual relay node. Alternatively, if candidates are located close to each other geographically, they might receive ACK messages from other candidates. In this case, lower-priority candidates might drop data packets to avoid duplicate transmissions. EXOR [12] applied an ACK-based coordination mechanism using a modification to the MAC layer of 802.11 protocol. In another control-based algorithm known as RTS-CTS acknowledge-based coordination, the sender node sends an RTS message to all nodes in the candidate set, while determining the order of candidates in the RTS message. Candidates that receive RTS messages will then respond

by sending back CTS messages to the sender node. Transmission of CTS messages is performed using a timer-based method, in which the higher-priority candidate sends the CTS message earlier than the lower-priority ones. Finally, the sender node starts sending data packets once it receives the first CTS message, and the originator of this CTS will be responsible for acting as the relay node [25].

Data-based coordination prevents nodes from sending extra control messages when selecting the relay node. Instead, via the propagation of data packets, candidates can decide whether they should transmit or discard them. Timer-based coordination and network coding methods are two well-known categories of data-based coordination. According to [25], in timer-based coordination, the source node generates the data packet and sends it to the list of prioritized candidates. Furthermore, a deadline is assigned to each candidate, after which they are allowed to forward the packet. Candidate nodes keep listening to the wireless channel and transmit the packet after the deadline has passed, only if they do not hear the transmission of the same packet by higher-level candidates. In the network coding approach, the source node aggregates some data packets into a single batch, splits this batch into several coded packets, and transmits a linear combination of them to the network. Candidates tend to forward packets only if they are linearly combined. Finally, the destination node will be required to decode the original set of packets, once it receives enough number of independent data packets. As one might note, both timer-based and network coding coordination methods might suffer from the propagation of duplicate packets. Furthermore, network coding approaches also introduce the cost of encoding and decoding data packets.

Transmission or Retransmissions

OR protocols are designed and proposed to be applied even in noisy and unreliable networks. In such networks, according to the quality of links between nodes, data packets might be lost due to channel obstructions. As explained in Section 1, this issue can be addressed as follows. If a sender node does not receive a notification from one of the candidates, it tends to retransmit the same packet for a maximum number of known trials. Such retransmissions are performed to increase the reliability of OR protocols, and to boost their throughput. If the source node does not receive a notification from any candidate after maximum retransmissions, it will permanently discard the packet from the network. However, some methods of packet transmission work with individual

packets, whereas others combine a group of packets and send them as a batch. Authors of [89] introduced CodeOR, which is an OR protocol combined with segmented network coding.

2.1.2 Unicast versus Multicast OR Protocols

Most of the OR protocols proposed in the literature consider unicast transmission, in which the data packet is supposed to reach only one destination. However, a few OR protocols, such as ROMP [120] and MSTOR [83], investigated the use of OR in multicast applications. Logically, the existence of more than one destination for the same data packet results in more complex candidate selection as well as coordination mechanisms. Most multicast OR protocols propose the creation of a multicast tree between the source node, and all of the desired destinations. Subsequently, they use OR protocols to propagate data packets over the created multicast tree.

2.1.3 Single-channel versus Multi-channel OR Protocols

Although most of the research in the area of OR focuses on developing protocols for Wireless Mesh Networks (WMNs) using a single channel, some work has involved applying OR in multi-channel scenarios. The idea of multi-channel OR protocols is to use the advantage of the wireless channel more efficiently, by utilizing the opportunity of working on various bit rates, as explained in 802.11 protocol suit. [81] suggested a multi-rate OR protocol in which nodes choose a transmission rate and a candidate set to send packets to a specific destination. Authors proposed a theoretical approach to optimize both the candidate set and the transmission rate. Furthermore, [122] introduced a game-theory-based approach in multi-rate and non-cooperative OR protocols, where nodes might decide to attack the network by violating OR rules.

2.1.4 OR Protocols in the Literature

Most of the research in the area of OR focuses on various methods of performing candidate selection or coordination. Amongst the two, however, candidate selection methods have received more attention from researchers in the past years. Some of the most important proposed protocols in the literature are briefly studied following in this section, and a summary of main characteristics of such protocols are depicted in Table 2.1.

2.1.5 OR Protocols based on Global Information

Extremely Opportunistic Routing (EXOR) [12] is one of the primarily published works in OR protocols, which uses the Expected Transmission Count (ETX) as a metric for candidate selection. As explained in [12, 25], ETX value between two nodes determines how many times a packet needs to be transmitted or retransmitted by a sender node over a link until the receiver node successfully receives the packet. More specifically, ETX between nodes i and j is calculated considering the link delivery probability between such nodes following $ETX_{i,j} = \frac{1}{link_{prob}(i,j)}$ equation where $link_{prob}(i,j)$ determines the link delivery probability between such nodes. First, EXOR tries to establish the shortest path between each node and its destination, then finds the first neighbor of each node as a potential candidate in this shortest path. Afterwards, if the ETX value from such a candidate to the destination is smaller when compared to the current node, this candidate will be added to the candidate set for the current node. This process is repeated until the maximum number of candidates is selected.

Similar to EXOR, Simple Opportunistic Adaptive Routing Protocol (SOAR) [105] uses the ETX metric for candidate selection. In SOAR, the shortest path between source and destination nodes is first calculated using the ETX metric. The set of candidates is then selected by adding nodes that are close to such a shortest path. Least-Cost Opportunistic Routing (LCOR) [64] is also a well-known OR protocol that uses the Expected Any-Path Transmission (EAX) metric for candidate selection. EAX has been developed considering the characteristics to OR protocols. The authors of LCOR have proven that this algorithm is capable of finding the optimum set of candidates in terms of expected number of transmissions by performing analysis on a network topology graph. In Opportunistic Any-Path Forwarding (OAPF) [135], nodes are first forced to select an initial list of candidates using the ETX metric. Afterwards, the primarily selected nodes will choose their candidates. In the end, the source node completes the process of candidate selection using the EAX metric. Similar to LCOR, MTS, which stands for Minimum Transmission Selection [88], proposes an optimum OR protocol using the EAX metric for candidate selection. MTS initiates the process of candidate selection from the destination node's neighbors, adds the destination node to the candidate set of all such neighbors and assigns the cost of each link using the EAX of each candidate to the destination. Afterwards, the algorithm iteratively finds the node with a minimum amount of EAX to the destination, for example bestNode, and adds bestNode and all

of its candidates to the initial candidate set of all bestNode's neighbors. Finally, the optimum candidate set is selected from this initial candidate set. It has been proved in [88] that this algorithm selects an optimum combination of candidates with regards to the expected number of transmissions (ETX). CORMAN [116] is an OR protocol being applicable in mobile ad hoc networks. This protocol uses a light-weight proactive routing protocol which allows nodes other than predetermined intermediate nodes in the routing path to retransmit the packet if it is necessary to do so. The work extends EXOR [12] to support mobility in routing.

2.1.6 OR Protocols based on Local Information

Unlike previously described protocols, which consider only the quality of links between nodes for candidate selection, there are a category of other protocols which focus on other aspects such as the geographical location of nodes. In CBF [69], which has originally been proposed as a routing algorithm for mobile ad-hoc networks [32, 28], the source node initiates the routing process by locating its own location information, as well as that of the destination node, inside the data packet. Afterwards, the source node broadcasts such a packet, and neighboring nodes receiving this packet will forward it towards the destination, following the timer-based coordination method and according to their distance to the destination as described in [25]. GOR [129], however, claims that choosing closest nodes to the destination does not guarantee selecting a suitable set of candidates. Instead, [129] introduces a new metric known as EOT and proposes a candidate selection algorithm with the purpose of maximizing the bit-meter progress per second using EOT metric. POR [127] is a simple OR protocol that selects the candidate set for each node by considering only the geographical location of nodes. To be more specific, each node in the network selects the neighbor, which results in gaining the highest amount of distance progress towards the destination, and adds this neighbor to its set of candidates. This process continues until the maximum number of candidates has been selected, or the number of neighbors is too small to meet demands. The basic concept behind POR algorithm is to decrease the number of required hops to send a packet by selecting the nearest neighbors for each particular node to the destination. Similar to POR, DPOR [60] benefits from the geographical location of nodes for its candidate selection. DPOR introduces a metric known as Expected Distance Progress (EDP), and attempts to establish a balance between the amount of achievable distance progress through each neighbor and the

link delivery probability between them. In fact, DPOR suggests candidate selection not only on their ability to progress the packet towards the destination, but which also have an acceptable link quality. On the other hand, [71] proposes an energy-efficient OR protocol customized for wireless sensor network environments [23], and [55] proposes an OR approach that takes into account quality of service considerations, while routing packets towards their destination. Table 2.1 shows some of the most significant OR protocols proposed in the literature and represents their important characteristics.

Table 2.1: Comparison of Opportunistic Routing Protocols

Protocol	Topology Knowledge	Metric	Application	Security	Coordination	Optimum
EXOR [12] Biwas et. al. (2004)	Global	ETX	Wireless Network	No	Timer-based	No
SOAR [105] Rozner et. al. (2009)	Global	ETX	Wireless Network	No	Timer-based	No
LCOR [64] Dubois et. al. (2007)	Global	EAX	Wireless Network	No	Timer-based	Yes
OAPF [135] Zhong et. al. (2006)	Global	EAX	Wireless Network	No	Timer-based	No
MTS [88] Li et. al. (2009)	Global	EAX	Wireless Network	No	Timer-based	Yes
CORMAN [116] Wang et. al. (2012)	Global	ETX	Wireless Network	No	Timer-based	No
GOR [129] Zeng et. al. (2007)	Local	EOT	Ad hoc Network	No	Ack-based	No
CBF [69] Fusler et. al. (2003)	Local	DP	Ad hoc Network	No	Timer-based	No
POR [127] Yang et. al. (2009)	Local	DP	Wireless Network	No	Timer-based	No
DPOR [60] Darehshoorzadeh et. al. (2012)	Local	EDP	Wireless Network	No	Perfect	No

2.2 Secure Routing in Wireless Networks

Apart from reliability requirements, consideration of security measurements is also of great significance. To be more specific, even the most reliable routing algorithms [115, 44] will not be able to effectively operate in presence of malicious nodes and attackers in the network. Authors of [80] and [3] provided reviews on security challenges in wireless sensor networks and mobile ad-hoc networks [49]. Regarding security obstacles in a wireless

network, plenty of research findings exist which consider cryptography solutions as a defensive mechanism against malicious nodes. Although such techniques can guarantee the safety and integrity of data transmission between nodes, they are incapable of defending against attacks related to collaboration of nodes in the network. Particularly, a separate category of misbehavior can be introduced when it comes to collaboration of nodes in hop-by-hop routing. For example, some malicious nodes may inject false information in the network, or prevent from forwarding packets as an intermediate node at the required time. Therefore, a different suit of security solutions should be devised to empower routing protocols against collaboration-related routing attacks in wireless networks. In the following subsections, security challenges and routing attacks for wireless networks are first explained in Section 2.2.1. Trust and reputation management protocols, that are known as the most significant approach to recognize and isolate malicious nodes, are studied in Section 2.2.2. Finally, security-related protocols that are proposed specifically for OR algorithms are explained in Sections 2.2.3, 2.2.4, and 2.2.5

2.2.1 Security Misbehavior in Routing

Various security attacks can be designed and launched for traditional and opportunistic routing protocols. For instance, impersonation, black-hole, worm-hole, gray-hole, packet modification, routing loop, Sybil, on-off, and replay attack are some of the most well-known routing attacks applicable in wireless networks [3, 56]. In impersonation, a malicious node tries to attack the network, for example by injecting false information, using unreal identities [3]. In black-hole, an attacker node attempts to absorb as much data packets as possible towards itself by advertising false information in the network; instead of forwarding such packets, however, it tends to drop all of them with the aim of decreasing network performance. [106] showed how destroying black-hole attack can be for the performance of mobile ad-hoc networks. Fig. 2.2 shows how black-hole attack can be applied to OR protocols. In fact, when the source node S sends a packet to its candidate set, say nodes $\{1, 2, 3\}$, assuming that node 1 is a black-hole node and receives the transmitted packet, it eliminates such a packet from the network, and sends false notifications to other candidates indicating that it has forwarded the received data packet. Consequently, such data packets will be lost from the network since none of the other candidates will attempt to forward them towards destination. Gray-hole attack can also be considered as a specific case of black-hole, in which the attacker node selectively and

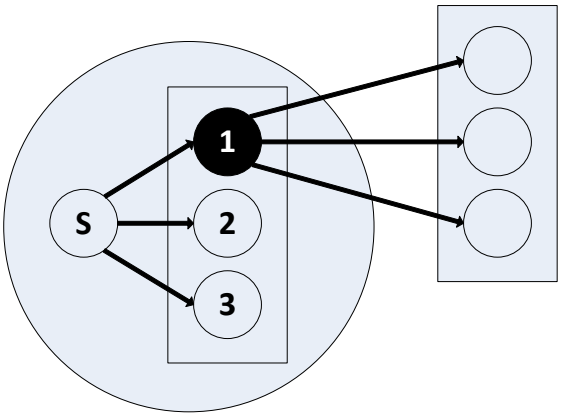

Figure 2.2: Node C_1 behaves maliciously and drops data packets instead of forwarding them

probabilistically forwards some of the received data packets, whereas it drops the rest of them. This behavior makes it more challenging to detect gray-hole nodes compared to black-hole attackers [3, 78].

Worm-hole is an example of collusion attack in which two malicious nodes collude with each other and attack the network collaboratively. In worm-hole, when one of the malicious nodes receives a data packet, it uses a private tunnel to transmit the received packet to the other malicious node, that is located at a different region of the network. Afterwards, the second malicious node receives and transmits the packet in its neighborhood [77]. Fig. 2.3, demonstrates how worm-hole attack is applied to OR protocols. In this example, node 1 tunnels all of the packets received from S to node 7, and coordinates with other candidates so they do not repeat transmitting the same packet. Afterwards, node 7 can either drop the packet or broadcast it in its neighborhood. This situation simulates that nodes 1 and 7 are neighbors while they are not. In routing loop attack, a malicious node tries to access and manipulate the routing information so the packet always rotates in a loop of nodes and never reaches its destination [121]. Similarly, in packet modification attack, an adversary node modifies the content of the packet and injects false information to the network with the purpose of wasting resources [126]. On the other hand, a malicious node that follows replay attack tries to reintroduce routing packets in the network repetitively. This will result in the injection of false and expired information into the routing table of network nodes. Finally, in on-off attack [11, 56],

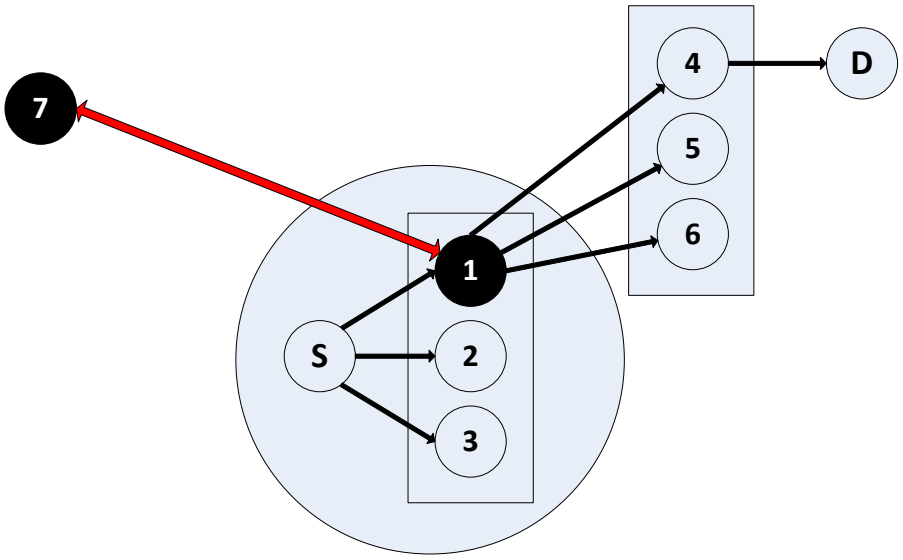

Figure 2.3: Nodes 1 and 7 attack the network following worm-hole attack

a malicious node switches its behavior between acting collaboratively and maliciously. In fact, the attacker decides to collaborate with some nodes in some circumstances, and attack them in other situations.

2.2.2 Trust and Reputation Management for Traditional Routing Protocols

Trust and reputation management systems are designed to assist in improving the security of communications in wireless networks.Articles published in [128, 57, 130, 124] provided surveys on trust management systems for wireless communications, mobile ad hoc networks, vehicular networks [1, 41], and Internet of Things. Similarly, [8] refers to a survey on the security challenges of cognitive radio networks and [76] introduced a review on different techniques of detecting malicious data injections by malicious or faulty nodes in wireless sensor networks [40]. In [51], CONFIDANT algorithm has been introduced. In CONFIDANT, the behavior of nodes is monitored in terms of packet forwarding. More specifically, nodes start to listen the wireless channel to ensure if next hop nodes collaborate in forwarding packets. Afterwards, upon observing a malicious behavior, nodes send an ALARM message to other previously known friends to punish uncooperative nodes.

Finally, a reputation value is calculated for each entity and nodes with lower reputation level are excluded from routing paths. In [96], CORE protocol has been introduced in which nodes benefit from watchdog mechanisms to establish trust. CORE came up with three different types of reputation called task-specific, direct, and indirect reputation. Nodes calculate a value for each of the three mentioned reputation types. Finally, by giving a weight to each reputation type, an ultimate reputation value for each node is computed and utilized. [110] proposed a trust establishment protocol by enforcing nodes to collaborate with each other in routing of mobile ad-hoc networks in order to recognize selfish nodes. Similarly, [2] proposed a trust-based mechanism to nullify the effect of selfish nodes in DSR protocol for mobile ad-hoc networks [22, 10].

The article published in [93] proposed a reputation system that works using watchdog mechanisms. Nodes in the proposed model only use direct interactions to collect and compute reputation information. After calculating the reputation value, misbehaving nodes are excluded from routing paths and, as a result, they are isolated in the network. The effect of the proposed model has been implemented and assessed using DSR algorithm [50]. Similarly, [102] benefited only from direct observations to calculate trust values. The proposed model attempted to find and establish secure end-to-end paths between the source and the destination nodes. Such paths are supposed to be free of attacker nodes. AODV [50] routing protocol was selected to implement and evaluate the proposed protocol. In [117], the trust value of nodes has been calculated using uncertain reasoning and the protocol has been applied in routing of mobile ad-hoc networks. This protocol used Bayesian Inference to calculate direct trust value of nodes. Furthermore, [117] benefited from the Dempster-Shafer theory of belief (DST) to deal with uncertainty of information. On the other hand, [48] presented a distributed and adaptive computational trust model, which establishes trust relationships between nodes to prevent malicious nodes from attacking the system. Furthermore, in [72] the concept of confidence level for trust calculation was introduced. The proposed model presented a completely distributed approach for trust calculation. In this protocol, once calculated, the trust value is weighted using the mentioned confidence level. Similarly, [17] introduced a trust management system for wireless sensor networks with the purpose of decreasing message and time overhead [62].

In SORI [74], it was assumed that nodes work in the promiscuous mode and are able to listen the wireless channel to evaluate the behavior of other nodes in the neighborhood.

SORI tired to encourage all nodes to collaborate in the network by proposing an objective reputation system. In this model, nodes periodically update the reputation value of their partners and broadcast such values to their neighborhood. Neighbor nodes that receive such information can update their indirect trust values. Finally, a punishment and rewarding mechanism was proposed allowing nodes to establish trust level of other nodes according to their behavior. [87] proposed a new trust model being applicable in opportunistic networks [100, 97]. In this algorithm, the concept of Positive Feedback Message (PFM) was introduced, in which once a node forwards a packet to the next hop, the next hop node generates a PFM and sends it back to the previous hop in order to verify the behavior of the current forwarder. Having done so, the history of interactions between nodes is created. Trust values in this model are calculated using both direct and indirect observations. On the other hand, [125] proposed a trust management framework being applied in software-defined networks [95]. The proposed model utilized cloud computing approaches to apply different trust-based services in a wireless network. Finally, a plenty of work has been done in the interest of trust management in vehicular ad hoc networks (VANETs) [6, 58, 132] considering specific characteristics of such networks. Articles published in [75, 103, 101, 118, 119, 92, 53] are some of the most important efforts conducted in this area. Table 2.2 represents some of the works published in the area of trust management protocols, and compares them together considering various features.

2.2.3 Trust Management for OR Protocols

Unlike previous subsection that focused on trust management proposed for traditional routing protocols, in this section we provide an overview on the most significant trust management schemes proposed for OR protocols. Considering that the focus of this thesis is on investigating security-related protocols and trust management for OR, related works in such areas are explained in greater details in the following subsections. This creates the possibility of understanding the idea behind such protocols more concretely.

Trust-based Minimum Cost Opportunistic Routing (MCOR)

MCOR [15] introduced a trust-based OR protocol for mobile ad-hoc networks. This protocol suggested using both direct and indirect interactions between network nodes to establish a trust value for each neighbor. The trust value is created using the history of interactions between nodes, and also the recommendations of others. Following literature

Table 2.2: Comparison of Trust Management Protocols

Protocol	Source of Information	Attacks	Routing Protocol	Performance Metric	Reliability
Marti et. al. (2000) [93]	Direct/Indirect Interactions	Black-hole	MANETs (DSR)	Throughput, Overhead	No
CONFIDANT Bucheg-ger et. al. (2002) [51]	Direct/Indirect Interactions	Black-hole, Malicious packet forwarding	MANETs (DSR)	Throughput, Dropped Packets, Overhead	No
CORE Michiardi et. al. (2002) [96]	Direct/Indirect Interactions	Selfish nodes Gossipping	MANETs	NA	No
SORI He et. al. (2004) [74]	Indirect Interactions	Selfish nodes, Impersonation, Black-hole	MANETs (DSR)	Throughput, Overhead	No
Pissinou et. al. (2004) [102]	Direct Interactions	Black-hole, Selfish nodes	MANETs (AODV)	Route overhead, Time complexity	No
TAODV Ghosh et. al. (2005) [72]	Direct/Indirect Interactions	Black hole, Gray hole, False accusation	MANETs (AODV)	Overhead, Routes selected, Route errors	No
Soltanali et. al. (2007) [110]	Direct/Indirect Interactions	Selfish nodes	MANETs (DSR)	Throughput, Detection percentage	No
Li et. al. (2008) [85]	Indirect Interactions	Selective forwarding, On-Off attack	MANETs	Trust/reputation	No
Adnane et. al. (2009) [2]	Direct/Indirect Interactions	Fake identity, Routing misbehavior	MANET (OLSR)	NA	No
Chen et. al. (2010) [53]	Direct Interactions, Role-based trust	Malicious message propagation	VANET	Relay effectiveness, Propagation distance of spam, Number of messages received per peer	No
Bo et. al. (2011) [14]	Direct/Indirect Interactions	Black-hole, Gray-hole	OR-based network (LCOR)	Throughput, Detection ratio, End-to-end delay, Expected ETX, Expected cost	Yes
Li et. al. (2013) [87]	Direct/ Indirect Interactions	Black-hole	OppNets (PROPHET)	Ratio of data attracted, Average Delay	No
Wei et. al. (2014) [119]	Direct/ Indirect Interactions	Black-hole, Packet modification	MANET (OLSRv2)	Delivery ratio, End-to-end delay, Throughput, Overhead	No
Wei et. al. (2015) [118]	Direct/Indirect Interactions	Packet dropping, Packet modification	VANET	Trust value, Probability of Successful Messages, Overhead	No

in the field of trust management protocols, [15] proposed to use the similarity of trust values, in order to prevent false accusations or collusions between nodes. Using the calculated trust value, a list of nodes with a trust value higher than a threshold is created. This list is used to create a trusted least-cost OR protocol that takes into account the quality of links to minimize the cost of reaching the destination. In fact, MCOR proposed that nodes only collaborate with neighbors that have a trust value higher than a specific threshold. Authors of MCOR formally calculated the cost of applying MCOR, and proved that their protocol is a minimum-cost trusted OR in terms of the EAX metric. Furthermore, authors considered the existence of black-hole and gray-hole nodes in the network, and evaluated the performance of the proposed protocol under the effect of such attacks. Compared to related works, [15] represented that MCOR has a higher throughput and detection ratio than its competitors.

Secure and Scalable Geographic Opportunistic Routing for WSNs (SGOR)

SGOR [91] focused on the security of geographical routing protocols for wireless sensor networks [36, 31, 98]. More precisely, SGOR was a distributed location verification protocol that utilized the Received Signal Strength (RSS) in the physical layer to detect attacks related to the location of nodes. Furthermore, SGOR proposed integrating trust management into geographical OR protocols, in order to recognize nodes that follow gray-hole attack. Authors of SGOR assumed that malicious nodes can compromise benign nodes. Once compromised, benign nodes will turn to attackers and act maliciously following gray-hole attack. Technically, SGOR was designed to operate using periodic broadcast of beacon messages in the one-hop neighborhood, which are sent by all nodes in the network. Upon reception of a beacon, nodes extract the reported location of the sender node, and calculate an estimation of the distance to the sender node using the RSS value. If the reported location does not match the calculated distance, the node is considered as a malicious node. Apart from the location verification, SGOR benefited from a trust model with the use of watchdog mechanisms in routing operations. More precisely, nodes observe the forwarding behavior of their neighbors, and assign them a trust value according to their level of cooperativeness. In fact, nodes combine direct trust values with indirect ones and calculate a final trust value for their neighbors. Finally, SGOR suggested using a combination of neighbors' trust values, and their distance to the sink node as the candidate selection metric. Following a security analysis, authors

provided extensive experimental results demonstrating resilience of SGOR against Sybil, gray-hole, and black-hole attacks. Although the paper was designed and evaluated to resist against location spoofing and gray-hole attacks, authors discussed that SGOR can be applied to detect and nullify other attacks including rushing, wormhole, replay, and collusion.

2.2.4 Game-theory-based Security Enhancements in OR Protocols

Game-theory-based security solutions for OR attempt to model the collaboration of network nodes as a game. In such approaches, nodes are players that always try to optimize their utility in the game. The idea of game-theory-based OR protocols is to establish a utility function through which nodes will be most benefited, only if they decide to collaborate with others. In fact, non-collaboration of nodes will result in the loss of their credit obtained in the network. Some game-theory-based OR protocols introduced in the literature are explained following in this subsection.

Enforceable Scheme for Packet Forwarding Cooperation (INPAC)

INPAC [54] proposed an incentive-compatible algorithm using network coding and opportunistic routing. The protocol was built on top of MORE [52] for wireless mesh networks. INPAC introduced incentive compatibility in routing as well as data forwarding phases, as described in the following. In the routing phase, each node decides how many transmissions should take place on each received packet, based upon the link loss reported by other nodes. In the forwarding phase, however, the actual packet transmission between nodes is taken into account. Therefore, in the routing process, users might prevent reporting their actual link loss value and confuse the routing process. INPAC designed an incentive algorithm in the routing phase that enforces nodes to truthfully report their link loss probabilities. Regarding the forwarding phase, the incentive mechanism is responsible for guaranteeing that each intermediate node performs the number of required transmissions as expected.

INPAC first focused on the economic aspect of the proposed approach, and modeled different transmissions of a packet using the concept of replaying the same game. The objective of the proposed protocol was to design a payment formula in which the source

node pays packet forwarders for their transmissions. Authors demonstrated that, following INPAC algorithm, nodes will be benefited the best if they perform exactly the expected number of required transmissions of a data packet. Second, the paper focused on the security aspect of the incentive model. In this phase, nodes monitor the forwarding behavior of their neighbors and report results. Such results are then used to pay neighbors. However, reporting false results may lead to wrong payment decisions. The paper used a combination of game theoretic and cryptographic approaches to allow nodes to punish peers that produce false reports. Furthermore, INPAC used the concept of credits in order to motivate the collaboration of nodes in the network. More specifically, once a node receives a packet, it should forward it exactly X times, where X is calculated using MORE [52] algorithm. It is worth noting that for each transmission, nodes will face a cost. However, the source node will pay intermediate nodes to compensate this cost, only if at least one of the receiver nodes in the path receives the transmitted packet. The difference between the gain and the cost will generate the utility of each transmission. INPAC designed a utility formula that maximizes when the node performs exactly X transmissions. Furthermore, authors proposed methods to prevent false reports (over-reporting), and to recognize nodes that ignore submitting reports (under-reporting).

A Game-theoretic Approach for Joint Multirate Opportunistic Routing and Forwarding (COMO)

In [123], authors proposed a game-theory-based cooperation-optimal incentive protocol, known as COMO, for multi-rate OR protocols. The focus of the proposed protocol was to defend against selfish nodes when such nodes prevent providing faithful reports regarding their link loss probabilities. In fact, link loss probabilities are a main source of operation in some opportunistic routing protocols such as MORE [52]. Therefore, COMO tried to mitigate this attack using a game theoretic approach for multi-rate OR protocols. Authors represented that if all nodes follow the routing protocol and the proposed incentive mechanism properly, the benefit of all nodes will be maximized in the network.

COMO was designed to be applied to an OR protocol that works based upon Expected Any-path Transmission Time (EATT) [82] metric. Authors designed COMO as a strategy game in which intermediate nodes are players. Similar to INPAC, the source node is

responsible for paying intermediate packet forwarders using a virtual currency for their collaboration in transmitting packets. Furthermore, a cost is assigned to each packet transmission. The difference between the payment and the cost will result in the utility of collaborating in routing operations. Authors assumed that all players attempt to maximize their utility, and used Nash equilibrium in game theory for modelling this assumption. Moreover, COMO proposed the concept of cooperation-optimal protocol for the created strategy game. A protocol is cooperation-optimal if it can reach a socially efficient, and also a strongly Pareto efficient Nash equilibrium. Based on the strongly Pareto strategy, nodes cannot increase their utility unless they decrease the utility of other nodes. Social efficiency refers to the maximization of the end-to-end throughput between source and destination nodes.

COMO utilized probe messages to measure the link loss probabilities of nodes. Furthermore, a cryptography approach was used to prevent the corruption of probe messages. Finally, a model was designed that guarantees nodes will not benefit if they deviate from following the proposed model. COMO consisted of three components known as link loss probability measurement, payment cap computation, and payment determination. Furthermore, COMO proposed using encrypted probe messages through which every intermediate node sends a specific number of probe messages. In addition, all of the receivers report the number of received messages to the source node using a traditional routing protocol. The source node will then be able to properly calculate the link loss probabilities. The payment cap was introduced to motivate participation of selfish nodes in the network. Finally, in order to enforce intermediate nodes, a payment mechanism was designed that connects forwarding behavior of nodes to their actual payment. Authors formally proved that COMO is cooperation-optimal in the strategic game of multirate OR. In their evaluations, authors represented that the utility and end-to-end throughput of nodes that faithfully follow the protocol will be higher than the ones that deviate by following selfish behaviors.

Auction Incentive Mechanism in Wireless Networks (AIM)

AIM [131] proposed an incentive mechanism to stimulate cooperation between nodes in an uncooperative wireless network, where nodes follow their individual interests. AIM introduced an auction game to assign node incentives. Authors proved that each relay node can only maximize its benefit if the game reaches the Bayesian Nash equilibrium.

In AIM, that is an extension to SOAR [105], source nodes pay intermediate nodes for transmitting packets. Each step of the forwarding process in AIM was modeled using an auction game. More specifically, once the set of candidates is determined, nodes act as bidders and determine a price by which they are willing to forward a packet. In addition, the energy of nodes is modeled and considered in the bidding process. The method of transferring the payment to each intermediate node is calculated similarly to a resale process. More particularly, the source node pays the first relay node the total amount required to deliver a packet to its final destination. This relay node will keep the amount related to its profit, and pays the rest to the next forwarder. This process continues until the packet reaches its final destination. From the routing point of view, the source node first notifies the candidate set that it has a packet to transmit. Candidates then calculate and report back their prices based on the estimated ETX to the destination, and their energy level. The source node then uses these prices to determine a prioritized set of candidates. Candidates then follow timer-based coordination to forward packets. Once a packet is forwarded, the relay node will receive its payment, and other candidates will drop the packet. The lower the price offered by a node, the higher the probability of this node acting as the relay node.

2.2.5 Other Security Challenges of OR Protocols

Apart from trust management and game-theory-based approaches, there are some works done examining different security considerations of OR protocols. For instance, as will be discussed in Chapter 4, we proposed an analytical model focusing on the performance evaluation of OR protocols under the effect of malicious nodes. The article published in [109], however, introduced a security enhancement for OR. Furthermore, we have developed a novel packet salvaging analytical model for OR that is going to be elaborated in greater details in Chapter 5.

Communication Privacy Protection using Encoded Opportunistic Routing (A-WEOR)

Authors of [109] proposed A-WEOR protocol. A-WEOR, short for Anonymous Wireless-mesh Encoded Opportunistic Routing, utilized network coding and opportunistic routing. The focus of A-WEOR was to resist against traffic analysis attacks. Nodes that follow a traffic analysis attack attempt to analyze traffic patterns at the physical or MAC

layer, and extract information accordingly. A-WEOR was constructed in addition to MORE protocol while concentrating on the anonymity of end-to-end communications. A-WEOR utilized the broadcast characteristic of data-oriented OR protocols, where no acknowledgment packets are sent between candidates for coordination. Therefore, malicious nodes will not be able to gain any information using traffic analysis attacks. A-WEOR was designed as an intermediate layer between IP and 802.11 MAC layers. This protocol took advantage of the quality of links, as well as network coding approaches. In particular, when a node sends some packets to its candidate set, only a linear combination of packets is transmitted by candidates, to avoid wasting network resources. The source or the relay node, however, must anonymously inform candidates about the proportion of packets they should forward towards the destination. In A-WEOR, the source node takes into account ETX values for each possible packet forwarder, and sends them encrypted Initial Request Packets (IRP). IRP packets indicate the proportion of encoded packets that each candidate should transmit towards the destination.

2.2.6 Comparison of Security-related OR Protocols

Previous subsections briefly discussed different protocols regarding the security challenges of OR. As explained in Sections 2.2.3, 2.2.4, and 2.2.5, security-related OR protocols have been classified into three categories: trust-based solutions, game-theory-oriented approaches, and other protocols. In this section, all of the investigated works are compared using different comparison metrics. Table 2.3 provides a summary of different protocols and their main features. In Table 2.3, the *CS Metric* column demonstrates the metric used in the OR protocol for candidate selection, whereas *CC* shows the candidate coordination method. If the proposed protocol is an extension to an existing OR protocol, the column labelled with *Basic OR* indicates the original OR protocol. Furthermore, the attack models considered in different protocols are shown in the *Attack Model* column, and the basic security solution used to defend against such attacks are presented in the *Security Mechanism* field. Finally, all of the evaluated metrics related to each protocol, along with their advantages and disadvantages, are labelled as *Evaluated Metrics*, *Pros*, and *Cons* respectively.

- **Protocol:** This column specifies a protocol using its abbreviated name, if applicable, and the related reference number.

Table 2.3: Comparison of Secure OR Protocols

Protocol	CS Metric	CC	Basic OR	Attack Model	Security Mechanism	Evaluated Metrics	Advantages	Disadvantages
MCOR [15]	EAX	Timer-based	LCOR	Black hole, Gray hole	Trust management	Throughput, end-to-end delay, ETX, Detection ratio, Security gain, Routing cost	Utilizing both direct and indirect interactions in calculating trust, Minimum cost	Depends on a threshold for selecting trusted neighbors, no trust-based metric introduced
SGOR [91]	Combination of distance and trust value	Timer-based	N/A	Location spoofing attack, gray hole attack, black hole attack	Trust management, RSS analysis using majority voting	Packet delivery ratio, end-to-end delay, overhead	Scalable, secure against location related attacks, resistant against Sybil attack	Possibility of creating single point of attack to sinks, dependent on transmission power of sender node
INPAC [54]	ETX	Network coding	MORE	False punishment, non-cooperation in packet forwarding	Game theory and cryptography	Utility of nodes following attacks, protocol running time	Performance efficient, successfully punishes malicious nodes	Depends on a central authority
COMO [123]	EATT	Network coding	MORE	Selfish nodes, deviating from routing protocol, not working on the optimal bit rate	Game theory and cryptography	Node utility, end-to-end throughput, communication overhead	Prevents selfish nodes, optimal performance, low communication overhead	Depends on traditional routing protocols
AIM [131]	ETX	Timer-based	SOAR	Selfish nodes	Incentive mechanism	Delay, throughput, number of invalid nodes, income, energy consumption	Simple algorithm, energy efficient	Unclear results in controlling malicious nodes, depends on honesty of nodes in making payments
A-WEOR [109]	ETX	Network coding	MORE	Traffic analysis attack	Group shared security keys	Throughput, KL anonymity measure	Ensures privacy of messages, resistant against statistical traffic analysis attacks	Vulnerable to denial of service attacks, needs centralized calculations, no active mechanism to exclude attacker nodes

- **CS Metric:** Different candidate selection metrics for different protocols are indicated in this column. As shown, INPAC, AIM, MCOR, and A-WEOR used ETX metric, whereas MCOR and COMO used EAX and EATT.

- **CC:** This field demonstrates different candidate coordination mechanisms used by each protocol. As shown, INPAC, COMO, and A-WEOR used network coding for coordination, while AIM, MCOR, and SGOR used timer-based coordination.

- **Basic OR:** This column indicates if a protocol has been developed as an extension to one of the existing OR protocols. As observed, three protocols, say INPAC, COMO, and A-WEOR, have extended MORE. AIM has been built on top of SOAR, and MCOR has been created as an extension to LCOR.
- **Attack Model:** This field demonstrates different attacks that have been investigated by different protocols. Packet dropping attacks including black-hole, gray-hole, and selfish nodes have been considered by most protocols including MCOR, SGOR, INPAC, COMO, and AIM. Some protocols have considered other security attacks such as traffic analysis, location spoofing, and so forth.
- **Security Mechanism:** In order to defend against malicious nodes, various security solutions have been proposed, as discussed in the previous sections. INPAC, COMO, and AIM proposed game-theory and incentive-based security enhancements, while MCOR and SGOR proposed trust-based solutions. In addition, A-WEOR used group shared security keys to improve the security of OR protocols.
- **Evaluated Metrics:** This field demonstrates what sort of parameters have been selected and evaluated in each of the discussed protocols. Logically, the performance of each protocol has been evaluated focusing on the security enhancement feature introduced by that protocol.
- **Advantages and Disadvantages:** The last two columns provide a summary of the advantages and disadvantages of each protocol. Using such fields, it would be possible to quickly compare the strengths and shortcomings of the different protocols. As observed, although each protocol focuses on specific security enhancements and challenges, all of them are still vulnerable to some sort of security attack. For instance, A-WEOR is resistant to statistical traffic analysis attacks, and ensures the privacy of messages. However, this protocol is still vulnerable to denial of service attacks, and has no mechanism for isolating malicious nodes.

2.3 Discussion

As explained and elaborated in Sections 2.2.3, 2.2.4, and 2.2.5 specific characteristics of OR protocols necessitate novel security solutions customized for such protocols. Although some works have been done focusing on security aspects of OR, such protocols

suffer from different problems as demonstrated in Table 2.3. Therefore, it would be of interest to develop a more extensive research on security aspects of OR algorithms. For instance, we believe there has not been enough effort conducted in the literature on designing comprehensive trust management algorithms for OR protocols. For this reason, a novel trust establishment method is designed and implemented for OR, introducing different candidate selection metrics and considering different application requirements. The effects of such metrics on various performance-related parameters are then evaluated and investigated. Chapter 3 contains more information regarding the proposed algorithm. Furthermore, it is important to study the effects of malicious nodes on OR-based wireless networks, and to design analytical models that assist in evaluating the negative effects of malicious nodes on such networks. To address this requirement, we propose an analytical approach that demonstrates the behavior of malicious nodes in wireless networks. This model has been elaborated in Chapter 4. Finally, a comprehensive packet salvaging mechanism is modeled, implemented, and evaluated for OR protocols. This model not only demonstrates the existence of malicious nodes in the network, but also introduces a mechanism that allows backup nodes in the candidate set to save some of the packets dropped by attackers. This model is explained thoroughly in Chapter 5.

Chapter 3

Trust-based Opportunistic Routing Protocol

This chapter focuses on proposing a trust and reputation management scheme for OR protocols. In fact, Section 3.1 proposes a trust model that only benefits from direct interactions between nodes, whereas Section 3.2 suggests to extend the proposed trust model to utilize both direct and indirect interactions when calculating the trust value of nodes. Finally, Section 3.3 summarizes the chapter.

3.1 Trust-based OR Protocol

Generally, trust management protocols motivate the principle of collaboration between nodes when it comes to packet forwarding and routing. Nodes, according to what they observe, independently decide whether to reward or punish other peers in the network. Naturally, punished nodes will have less chance to be selected in future interactions whereas rewarded nodes will be recognized as cooperative and trusted entities. As mentioned in Chapter 2 Trust and reputation management systems adopt different sources of information. Direct observations that are extracted from first-hand interactions between nodes as well as indirect observations being obtained from recommendations are the most important samples.

In this chapter, a novel trust-aware OR protocol is proposed being enable to route data packets towards their destination not only using the link delivery probability between nodes but also according to the trust value of relay nodes. As depicted in Fig. 3.1,

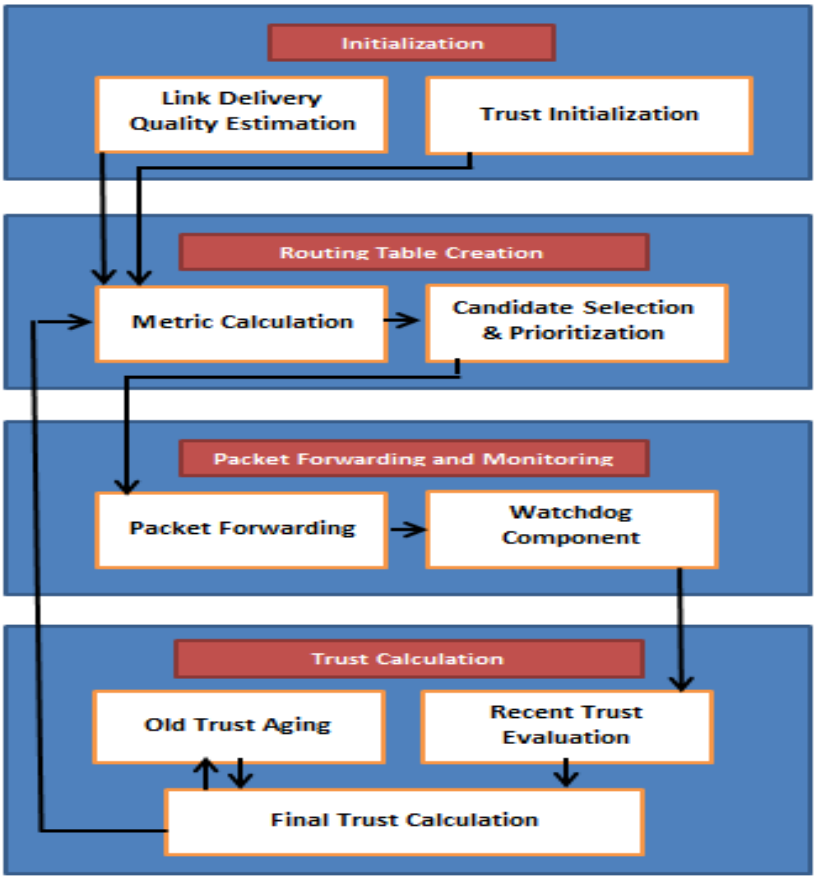

Figure 3.1: System Components

the introduced method is composed of four main operational modules known as initialization, routing table creation, packet forwarding and monitoring, and trust calculation. The initialization component in each node assigns an application-based default trust value to all nodes in the network representing that nodes have no information about other peers. Furthermore, the link-delivery quality estimation component in each node calculates the link delivery probability between this node and its neighbors according to the application and network environment. Details of the calculation of link delivery probability will be discussed in Section 3.1.4. Afterwards, based on the obtained information and using the metric calculation component, the routing table creation module assigns a rating value to all neighboring nodes and assists in prioritizing neighbors for candidate selection. In continue, the candidate set of each node for each particular destination is computed and the routing table of nodes is created using such information.

Following the routing table creation phase, the packet forwarding and monitoring module in each node starts handling transmission of data packets according to the created routing table. Meanwhile, the watchdog component keeps monitoring the behavior of candidate nodes. This component reports the results of its observations periodically to the trust calculation module. By receiving data from the watchdog component, the trust calculation module evaluates the most recent trust value of nodes and, after discounting the previously calculated trust value by a forgetting factor, the final trust value of nodes is computed. Finally, the routing table creation process is repeated. This enables nodes to select a new candidate set, which may be different from the previously calculated one. Greater details of each module can be found in the following subsections.

3.1.1 Routing Table Creation

As explained in Chapter 1.1 and following [25], candidate selection is considered as one of the most important phases in OR protocols. During this phase, each node selects a set of candidates for each particular destination to collaboratively route packets. Therefore, the list of the selected candidates for each possible destination is stored in the routing table of each node. As mentioned in Chapter 1.1, candidate selection is conducted utilizing a metric, through which nodes will be able to rank other nodes and select their candidates. To the best of our knowledge, however, most of the proposed candidate selection algorithms in the literature do not consider the security and trust-related measurements as an important parameter. This may result in selecting untrustworthy candidates and, consequently, in decreasing network performance.

In this section, three different candidate selection metrics are introduced through which nodes will be able to calculate the priority of each neighbor and select their best n candidates. All of the three proposed metrics are local metrics. This conveys that nodes in the network only need to access the location information of their neighbors, and that of the destination. Having such information, according to [59], as will be discussed in section 3.1.4, nodes will be able to calculate the link delivery probability for each neighbor node. Finally, by incorporating the link delivery probability of neighbors with other parameters such as the trust value or the geographic location, different metrics can be defined and, as a consequence, various candidate sets can be selected. A more detailed explanation on characteristics of each metric can be found in the following subsections. Prior to discuss about the proposed metrics, as an example, suppose that

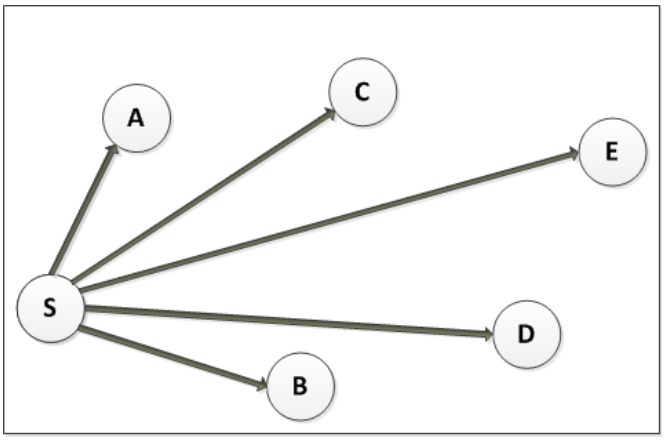

Figure 3.2: Sample topology of node S as its neighbors

Table 3.1: Available information to node S regarding its neighbors

Node	Trust Value	Delivery Probability	Progress
A	0.85	0.90	30
B	0.60	0.85	50
C	0.90	0.80	70
D	0.85	0.75	90
E	0.75	0.70	110

Fig. 3.2 represents a network in which node S intends to select its candidate set and nodes A, B, C, D, and E are the list of the qualified neighbors. For all of the proposed metrics, the list of the potential candidates only contains nodes that are closer than the current node to the destination. Furthermore, suppose that Table 3.1 represents the hypothetical link delivery probability, trust value, and achievable distance progress of all neighbors through node S. The distance progress (DP) is computed using Equation 3.1 inspired by [60], where $D_{s,d}$ demonstrates the distance between the source and the destination nodes, and $D_{n_i,d}$ accounts for the distance between neighbor n_i and the destination.

$$DP_{n_i}^{s,d} = D_{s,d} - D_{n_i,d} \tag{3.1}$$

RTOR Metric

Reliable Trust-based Opportunistic Routing (RTOR) metric, proposes to select neighbors having high link delivery probability and trust value as candidate nodes. This means to select nodes that are reasonably trustable in terms of packet forwarding, and also reliable when it comes to link delivery probability. Equation 3.2 shows how each node can calculate the RTOR metric for each of the neighbors, where T_{S,n_i} represents the trust value of node n_i from the perspective of node S and p_{S,n_i} shows the link delivery probability between such nodes. For instance, assuming that each node selects the maximum of two candidates, node S will choose set $\{A,\ C\}$ as its candidate set after calculating RTOR metric for all of its neighbors, and choosing the best peers.

$$RTOR(S, n_i) = T_{S,n_i} \times p_{S,n_i} \tag{3.2}$$

TORDP Metric

Although the RTOR metric tends to select candidates from the most reliable and trustworthy nodes, it should be noticed that such nodes are the ones that are close enough to the source node. More specifically, according to RTOR, the closer a node is to the source node, the higher its delivery probability, and the higher its chance to be selected as a qualified node in the candidate set. The issue is intensified if the trust value of neighbor nodes tends to be very close to each other. For instance, upon the initialization phase, when nodes do not have any information about each other, they use a predetermined default trust value for all nodes in the network. This situation results in forwarding packets from long routes between the source and the destination and, consequently, higher amount of the end-to-end delay compared to other proposed metrics.

$$TORDP(S, n_i) = \begin{cases} T_{S,n_i} \times p_{S,n_i} & DP_{n_i}^{S,D} \geq MinDP \\ 0 & Otherwise \end{cases} \tag{3.3}$$

TORDP stands for Trust-based OR metric which also considers a minimum amount of distance progress as threshold. TORDP, similar to RTOR, takes into account the reliability of links between nodes as well as their trust value. The only difference is that TORDP limits the set of qualified neighbor nodes to the ones being able to progress packets for at least a minimum distance progress (MinDP) amount. MinDP is a system parameter that can be adjusted according to different application scenarios and

environments. Equation 3.3 represents how TORDP metric is calculated. For example, assuming that MinDP is set to fifty meters, in Fig. 3.2 node A will be disqualified when node S decides to select its candidate set and nodes $\{C,\ D\}$ will be qualified and selected instead.

GEOTOR Metric

Finally, GEOTOR is a metric that incorporates trust level of nodes not only with their link delivery probability, but also with the achievable distance progress through each neighbor. According to equation 3.4, GEOTOR metric aims to establish a balance between all of the mentioned parameters by selecting nodes which have reasonable trust value, link delivery probability, and distance progress. In Equation 3.4, $DP_{n_i}^{S,D}$ accounts for the distance progress that is obtained through neighbor n_i. Considering Fig. 3.2 and Table 3.1, the set $\{D,\ E\}$ forms the candidate set of node S based on the GEOTOR metric.

$$GEOTOR(S, n_i) = T_{S,n_i} \times p_{S,n_i} \times DP_{n_i}^{S,D} \tag{3.4}$$

3.1.2 Trust Calculation Method

By completing the candidate selection process using one of the proposed metrics, source nodes start sending their data packets towards the destination and candidate nodes will be responsible to coordinate with each other and cooperatively forward packets. Similarly, all intermediate nodes should follow the same process when receiving a packet to forward. However, in realistic scenarios, some nodes may not cooperate with others for selfish or malicious purposes. As explained in Chapter 1 Section 1.2 and Chapter 2 Section 2.2, such nodes can easily betray the system, for example, by blocking the communication route between source and destination nodes and dropping received data packets. Therefore, this is very important for all nodes in the network to detect such uncooperative nodes and avoid further communications with them.

In this chapter, we propose to equip benign nodes in the network with a trust model, enabling them to recognize and isolate malicious nodes properly [17]. More specifically, when nodes transmit data packets, they start monitoring the behavior of their candidates, and collecting information about their packet forwarding behavior during predetermined time intervals. Afterwards, nodes will be able to calculate or recalculate the trust value of

their candidates, and repeat the process of candidate selection. This way, nodes will have a chance to detect malicious nodes and exclude them from their set of candidates. In order to calculate the trust value, nodes in the initial version of the proposed model benefit only from direct observations. Section 3.2 demonstrates how to incorporate indirect interactions into the introduced trust model.

Following literature in the area of trust management in wireless networks such as [70, 87], the Beta distribution, $Beta(\alpha, \beta)$, is selected for trust computation, where α accounts for the number of successful interactions that one node observes about another node whereas β, in contrast, demonstrates the number of unsuccessful interactions. In this protocol, α and β are calculated periodically in each Δt period of time and a new trust value is calculated for each interval. An extensive explanation of how to calculate the values of α and β in networks with unreliable links is presented in the following subsection. Once α and β are obtained, it would be possible to map such parameters to the trust value of nodes using Beta distribution. However, in realistic scenarios and especially when the physical links between nodes is not reliable, there will always be some amount of uncertainty in calculation of α and β that should be considered as well. For this purpose, a method for uncertainty modeling is applied inspired by [84, 87, 117] extracted from the Dempster-Shafer's Theory of belief (DST) [108]. In fact, having the values of α and β, it would be possible to map such parameters into three different values demonstrating belief, disbelief, and uncertainty of calculations. Belief $b_{i,s}$ represents the belief of node i to node j for its reasonable forwarding behavior, while disbelief $d_{i,j}$ accounts for disbelief of an acceptable forwarding behavior. Finally, $u_{i,j}$ demonstrates the amount of uncertainty in reasonable forwarding behavior. The values of $b_{i,s}$, $d_{i,s}$, and $u_{i,s}$ should be calculated such that $b_{i,j} + d_{i,j} + u_{i,j} = 1$ is satisfied. Following equations are applied to calculate the mentioned parameters inspired by [70, 87].

$$b_{i,j} = \frac{\alpha_{i,j}}{\alpha_{i,j} + \beta_{i,j}} \times (1 - u_{i,j}) \tag{3.5}$$

$$d_{i,j} = \frac{\beta_{i,j}}{\alpha_{i,j} + \beta_{i,j}} \times (1 - u_{i,j}) \tag{3.6}$$

$$u_{i,j} = \frac{12 \times \alpha_{i,j} \times \beta_{i,j}}{(\alpha_{i,j} + \beta_{i,j})^2 \times (1 + \alpha_{i,j} + \beta_{i,j})} \tag{3.7}$$

Upon calculation of the uncertainty, it would be possible to map the belief value into a trust value for the latest Δt as represented in Equation 3.8, where σ represents the

relative atomicity factor of the system. Relative atomicity determines what proportion of the calculated uncertainty value should be allocated to belief or disbelief of nodes [87]. This value can be adjusted empirically for different environments and applications.

$$Trust^{direct}_{new}(i,j) = b_{i,j} + \sigma \times u_{i,j} \tag{3.8}$$

As stated above, however, the most recently computed trust corresponds to the latest Δt time interval. Therefore, this value should be incorporated with the previously calculated trust value to compute the final trust value of nodes for the whole time. This is motivated by the fact that different nodes in the network may act differently at various times and situations. Therefore, the final trust value of nodes is calculated using a weighted averaging method by allocating proper weights to the most recent trust value, and the previously accumulated one. Furthermore, following the literature regarding trust management in wireless networks, the evaluated trust value of nodes is forgotten in the course of time to demonstrate higher importance for the most recent trust values. The following equation represents how to incorporate the most recent trust value with the previously calculated one, while the fading and ageing characteristics of trust have been modelled. In Equation3.16, the value of $0 < \theta < 1$ represents the ageing factor, and $0 < \omega < 1$ is used to determine the weight of the most recent trust value. θ and ω are parameters in the system which can be tuned according to different system requirements and applications. The next subsection demonstrates how to obtain and estimate the values of α and β particularly when the link between nodes is not reliable.

$$Trust^{direct}_{final}(i,j) = \omega \times Trust^{direct}_{new}(i,j) + \theta \times (1-\omega) \times Trust^{direct}_{old}(i,j) \tag{3.9}$$

3.1.3 Watchdog Component for Trust Model

Watchdog is a well-known approach in the field of trust management through which nodes would be able to monitor the behavior of others and evaluate them accordingly. According to watchdog systems, when a node sends a packet to be forwarded by a next-hop neighbor, it starts listening to the wireless channel for a limited amount of time to receive a passive acknowledgement. A passive acknowledgement demonstrates if the next hop node properly forwards a packet towards the destination or not. Having done so, the sending node can determine whether or not the receiving node has received and

cooperated in forwarding packets. By applying the watchdog mechanism, each sending node will be able to create a history of interactions for neighbors that it has directly communicated with them in the past. Similarly, the watchdog idea can be modified and applied in the context of OR protocols. In fact, by transmitting a packet, a node listens to the wireless channel for a limited time interval until it receives a passive acknowledgement from one of the nodes in the candidate set. If the sending node does not hear the transmission of its packet by one of candidate nodes, it tends to retransmit the packet for a limited number of trials.

The watchdog mechanism works well for scenarios where the communication channel between nodes is reliable and nodes can guarantee the reception of their packets to neighbors. In realistic scenarios, however, it is possible that transmitted packets by one node are not delivered to other nodes. This occurs due to a collision in the network, or as a result of having noise in the communication channel. In fact, the link quality between nodes can considerably affect the quality of a watchdog system. and this parameter should be considered when monitoring neighbor nodes. More specifically, the fact that a sending node does not receive a passive acknowledgement might occur as a consequence of packet losses in the wireless links between nodes.

For example, in figure 3.3, suppose that node S selects the next-hop nodes using GEOTOR metric. As discussed earlier, nodes $\{D, E\}$ are selected in the candidate set. Therefore, node S expects to hear passive acknowledgement messages from either E or D. However, considering node E as the higher-priority candidate, it should be noticed that 30 percent of packets over the link $S - E$ may be lost due to wireless channel obstructions. Therefore, it is possible that node E does not even receive some of the packets from node S. Similarly, it is possible that E receives and transmits packets but S does not hear the passive acknowledgement regarding such transmissions. As a consequence, S will not be able to have a clear understanding about forwarding behavior of its candidate set, and will not be able to correctly compute the trust value of its candidates. The reason is that node S can not ratify whether its candidates have not received transmitted packets, or they have received and maliciously dropped them.

Furthermore, although both nodes $\{D, E\}$ have been selected as candidate nodes, they do not have similar roles concerning packet forwarding when they receive a packet. In fact, such nodes coordinate with each other and one of them will forward the packet. For example, considering timer-based coordination for packet forwarding, and assuming

that node E is selected as the highest-priority candidate, this node will be responsible to forward all of the received packets. Node D transmits a packet if and only if it does not hear the transmission of the same packet by node E, after a predetermined coordination time. This conveys that trust value of different candidates should be calculated differently according to their priority in the candidate set.

In this chapter, a novel method for watchdog component is introduced, which makes each node of the system enable to accurately monitor the behavior of its candidates not only according to the link delivery probability between candidates, but also using their priority in the candidate set. More specifically, having the link delivery probability between neighbors, it is possible to calculate the probability that each candidate will send a packet. Afterwards, the sending node can compute the probability of receiving a passive acknowledgement from each candidate, and for every packet it transmits. Assuming the symmetric link delivery probability between nodes, as shown in the Fig. 3.3, the following equations represent the probability of receiving a passive acknowledgement for each packet given that node S has selected nodes $\{E,\ D\}$ as its candidates using GEOTOR metric, and E is the highest-priority candidate.

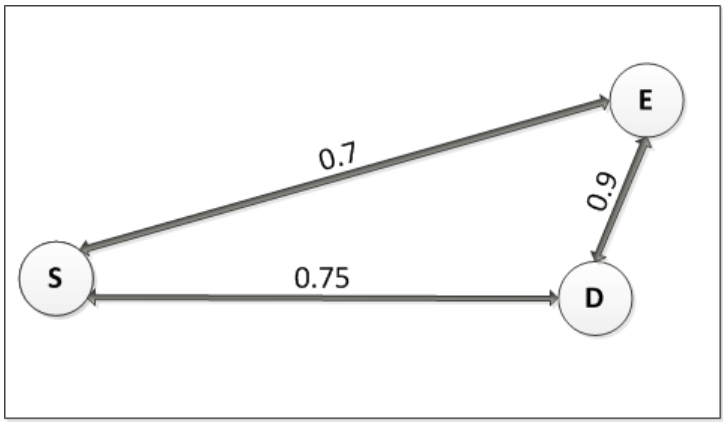

Figure 3.3: Link quality between S and its candidates

$$Prob_{S,E}^{ACK} = p_{S,E}^2 \tag{3.10}$$

$$Prob_{S,D}^{ACK} = p_{S,D}^2((1 - p_{S,E}) + (p_{S,E} \times (1 - p_{E,D}))) \tag{3.11}$$

In Equation 3.10, $Prob_{S,E}^{ACK}$ accounts for the probability of receiving a passive acknowledgement by node S from node E for each packet. To be more specific, $Prob_{S,E}^{ACK}$ is the probability that node E receives the transmission of node S, and node S also hears the transmission of the same packet by node E. On the other hand, as the second-priority node in the candidate set, given that it has received the transmission of a particular packet from node S, node D will tend to forward the packet only if it does not hear the transmission of the same packet from higher priority candidates after the coordination time. The transmission of D in this example occurs if either node E does not receive a packet from S, this takes place with the probability of $(1 - p_{S,E})$, or if D does not hear that node E has sent the packet. This second scenario occurs with the probability of $(p_{S,E} \times (1 - p_{E,D}))$. Finally, if node D, as the second candidate, forwards the packet, this transmission should still be heard by node S to be considered as a passive acknowledgement, and to avoid S from retransmission of the same packet.

In continue, having the probability of receiving a passive acknowledgement from each candidate for every packet, and assuming that each node can choose the maximum of two candidates, nodes would be able to calculate the expected number of received passive acknowledgements from each candidate for the current trust calculation interval. This is conducted by keeping the track of the number of transmitted packets for each candidate in every Δt period of time and, using the above equations, calculating the expected number of received acknowledgements for each candidate. It is important to emphasize, however, that calculation of the expected number of received acknowledgements for each candidate is done only for the first transmission of packets, and retransmissions are not considered in computations.

Finally, each node will be able to calculate the values of α and β by comparing the expected number of passive acknowledgements with the actual number of received ones. For instance, considering packet droppings as an attack in the network, assume that node S tracks the transmission of sending n packets to its candidates and node E acts as a packet dropper, which receives and just drops received packets. Therefore, S will not hear any passive acknowledgement from E, whereas it expects to receive some based upon the above equations. Therefore, S calculates the value of α for the mentioned period regarding node E as zero, which leads to a significant decrease in the final trust value of E. Consequently, in the next round of routing table creation, node E which has been assigned a lower level of trust value compared to the previous round will have a smaller

chance to be selected as a candidate node by S, and may be replaced by another node. This results in an isolation of such a malicious node from the network if it continues to act maliciously in the future.

3.1.4 Performance Evaluation

This section contains a performance evaluation on the proposed model. Section 3.1.4 consists of some details regarding the simulation setup and results are explained in Section 3.1.4.

Simulation Setup

In order to implement and examine the effects of the proposed trust model, we have used network simulation [27] using Network Simulator (ns-2.35) [68] tool. Additionally, to evaluate the effect of any trust model, the proposed model should be assessed in presence of malicious nodes in the network. For this reason, the packet dropping or black hole attack has been selected in this thesis. As explained in Chapter 2 Section 2.2.1, a black hole node receives all of the data packets it is responsible to route. Instead of forwarding, however, it drops them to decrease the network performance. Black hole attack is considered as a Denial of Service (DoS) attack. Therefore, trust models should attempt in detecting black hole nodes and isolating them in the network.

Table 3.2 represents selected simulation parameters after performing several initial experiments to adjust such values. As discussed in earlier sections, all nodes of the network are considered to be benign at the start. This can be performed by allocating an initial trust value to all nodes. In the course of the time, however, the trust calculation module with the aid of watchdog component gathers information regarding candidate nodes, and tries to exclude malicious nodes as explained earlier. Furthermore, timer-based coordination method has been implemented between candidate nodes. According to [25], using timer-based coordination, when a candidate node receives a packet, it tends to transmit this packet after a period of time, determined according to the priority of this node in the candidate set. Finally, all simulations have been performed in a static wireless network. The following subsections contain propagation model setup, and describe different investigated simulation scenarios.

- **Propagation Model Settings:** In this thesis, the shadowing propagation model

Table 3.2: Simulation Parameters

Parameter	Value
Propagation Model	Shadowing
MAC	802.11
Number of Nodes	40 to 100
Field Size	(1000×1000)
Number of Candidates	2
Number of Connections	1 to 10
Maximum Number of Retransmissions	3
Transmission Rate	4 Packets/Second
Trust Calculation Interval (Δt)	20 Seconds
Default Trust Value	0.8
New Trust Value's Weight (ω)	0.3
Trust Aging Factor (θ)	0.98
Coordination Delay	15 Milliseconds
Number of Malicious Nodes	0 to 30
Simulation Time	600 Seconds

is exploited to simulate a realistic environment containing noise in the wireless channel. In the proposed trust model, as discussed before, it is supposed that all nodes are able to access the location information of their neighbors as well as that of the destination node. This can be done if nodes execute a distributed location service such as the one introduced in [86] or using a location database available in the network. Having location information, nodes will be able to compute the Euclidean distance to their neighbors. Therefore, using the standard shadowing propagation model parameters indicated in [68], and following [60], Equation 3.12 is applied. Using Equation 3.12 each node will not only be able to calculate the power received of a packet to each neighbor according to the transmission power, but also it can estimate the link delivery probability accordingly.

$$P_r(d)|dB = 10log_{10}\left(\frac{P_t G_t G_r \lambda^2}{L(4\pi)^2 d^\beta}\right) + X_{dB} \tag{3.12}$$

Table 3.3: Propagation Model Parameters

Parameter	Value
P_t	0.28183815 Watt
RXThersh	3.652×10^{-10} Watt
G_t, G_r, L	1
f	914 MHz
β	2.7
σ	6.0

In Equation 3.12, P_t and P_r account for the transmitted and received power of a signal at distance d respectively. Similarly, G_t is the transmission and G_r is the reception antenna gain. λ shows the wavelength of the signal ($\lambda = c/f$ where $c = 3 \times 10^8$, and f is the signal frequency) and L accounts for the system loss. Finally, as two of the most important parameters in the propagation model, β is considered to simulate the path loss exponent and X_{dB} represents a Gaussian random variable having zero as the mean and σ_{dB} as the standard deviation. According to Equation 3.12, the higher value of β simulates the existence of greater noise and obstructions in propagation of wireless signals and, as a result, decreases the received power at the receiver side. More specifically, the larger the value of β, the higher the likelihood of signal loss in the channel [60].

Technically, packets at the receiver side of a link are safely arrived if the received power of a signal is equal or higher than a specific threshold known as $RXThresh$. Considering this threshold, and assuming that the receiver node is at distance d, it is possible to calculate the link delivery probability between of this node, say $p(d)$, using Equation 3.13. Following [68], other related parameters are listed in Table 3.3 demonstrating standard parameters for the shadowing propagation model, assuming that the network is running in an outdoor environment.

$$p(d) = Prob(P_r(d)|dB) \geq 10log_{10}(RXThresh)) \tag{3.13}$$

- **Simulation Scenarios** In order to assess the performance of the proposed trust model in different situations, three different scenarios have been considered in simulations. For each scenario, the number of malevolent nodes, the density of nodes,

and the number of data connections have been considered as variable parameters. On the other hand, four different protocols have been implemented and compared. The first protocol, labelled as Baseline Protocol, is the one which selects candidates using RTOR metric but does not have any trust model implemented against attacker nodes. RTOR, TORDP, and GEOTOR represent different variations of the proposed trust model, which operate utilizing RTOR, TORDP, and GEOTOR metrics for candidate selection, respectively.

Regarding data communication between nodes, the first source-destination connection in the network is formed while the source node is located at the bottom-left corner of the simulation area, and the destination node is situated at the top-right corner. This setting represents the largest possible distance for data communication in the network. Apart from this special connection that starts at the beginning of simulation, other connections are established between nodes located at random locations in the network. The minimum distance progress (MinDP) for TORDP algorithm in all of the scenarios is set to 50 meters. This conveys that candidates should be selected given that they provide at least 50 meters of distance progress. This value was selected experimentally after performing a set of initial simulations.

- **Scenario 1 (Effect of Malicious Nodes):** In this scenario, the number of malicious nodes is considered as a variable parameter in simulations. The number of nodes for this case is set to 100 where 10 different source-destination data connections are created between nodes. The purpose of this scenario is to evaluate the resistance of each protocol, when the density of malicious nodes changes from zero to 30 nodes.
- **Scenario 2 (Effect of Nodes Density):** In the second scenario, the number of malevolent nodes is set equal to 20 percent of the total number of nodes in the network. However, the density of nodes in the system is changed between 20 to 100 in a 1000×1000 squared meter simulation field. The goal of this scenario is to evaluate different variations of the proposed algorithm by changing the density of nodes. In fact, the scalability of the system under different node densities is measured and analysed.
- **Scenario 3 (Effect of Data Connections):** Finally, the third scenario is designed to assess the performance of the proposed algorithms by changing

the number of source-destination data flows from 1 to 10, while setting the number of attacker nodes to 20, and the number of nodes in the network to 100. This scenario evaluates the efficiency of the proposed models under the effect of different data traffic loads in the network.

Results

This subsection consists of several graphs representing the results of performed simulations under various scenarios. Each plot is depicted and extracted as the average of 100 simulation executions, and considering 95% confidence intervals. Packet delivery ratio, end-to-end delay of transmitting packets, as well as the average hop count have been selected and evaluated as simulation parameters.

- **Packet Delivery Ratio**

  Fig. 3.4 shows the packet delivery ratio under the first scenario, where the number of malicious nodes changes. As observed, RTOR protocol demonstrates the best delivery ratio compared to other variations. The baseline protocol, which selects candidate nodes similar to RTOR, shows the same delivery ratio when there is no attackers in the system. However, its performance decreases as the number of malicious nodes increases in the network. TORDP and GEOTOR represent an acceptable performance compared to the Baseline Protocol when the number of malevolent nodes is greater than 8 nodes. GEOTOR, however, delivers slightly higher number of packets than TORDP. This is due to the fact that GEOTOR tries to establish a balance between trust value, link delivery probability, and distance progress. More specifically, GEOTOR dynamically adapts its candidate selection method according to the currently-accessible neighbouring nodes. Furthermore, RTOR reasonably demonstrates the best packet delivery ratio compared to other algorithms by taking advantage its most reliable candidate selection algorithm. Finally, the delivery ratio for all of the protocols decreases by raising the number of malicious nodes. However, the slope of this decrease is by far smaller for protocols that benefit from the proposed trust model, compared to the baseline protocol.

  Fig. 3.5 represents the delivery ratio for the second scenario where the density of nodes is changing. Here too, RTOR shows the best performance compared to other protocols and GEOTOR ranks second. It can be observed that by increasing

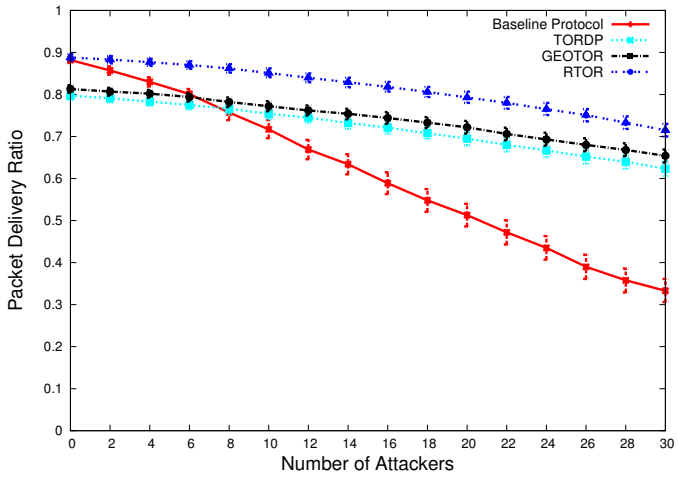

Figure 3.4: Packet delivery ratio by changing the number of malicious nodes

the number of nodes in the network, the packet delivery ratio for all protocols grows. The reason is that, by increasing the node density, benign nodes will have the chance to select their candidates from a larger set of nodes. Furthermore, the ratio of increase in the delivery ratio of protocols that benefit from the proposed trust model is higher than the baseline protocol. This is due to the fact that in the baseline protocol, if a malicious node is selected as a candidate, it has the chance to drop all of the received packets. However, for RTOR, GEOTOR, and TORDP the behavior of malicious nodes is monitored and their trust value is be recalculated. This may lead to substitution of malevolent nodes with benevolent ones, as discussed in the previous section.

In order to evaluate the delivery ratio of packets under the third scenario, Fig. 3.6 has been plotted. The delivery ratio for all of the models shows almost a constant trend specially when the number of connections is greater than two. The reason is that the first connection between nodes in all of the scenarios is established considering the largest possible path in the network. This results in loosing higher number of packets due to signal fading as well as malicious packet droppings. Here too, RTOR shows the highest packet delivery ratio, GEOTOR ranks second, and the baseline protocol which can not defend against malicious nodes shows the works performance.

- **End-to-end Delay** End-to-end delay is investigated as the second parameter to

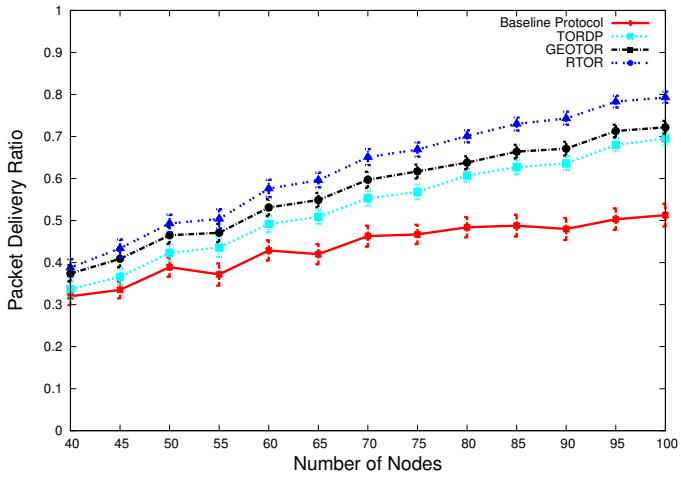

Figure 3.5: Packet delivery ratio by changing the number of nodes

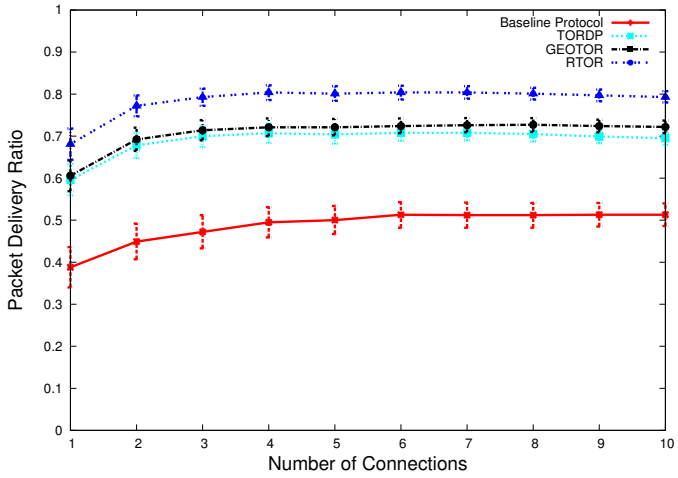

Figure 3.6: Packet delivery ratio by changing the number of source-destination data connections

compare proposed protocols. Here too, the end-to-end delay of the received packets is depicted for all described scenarios. However, it is necessary to emphasize that end-to-end delay is only calculated for packets that are received to their final destination and no delay is calculated for dropped packets.

As observed in Fig. 3.7, the end-to-end delay for GEOTOR algorithm shows the best performance when it comes to comparison with RTOR, TORDP, and the Baseline Protocol. The reason is that GEOTOR attempts to make a balance between all of the three important parameters containing trust value, link delivery

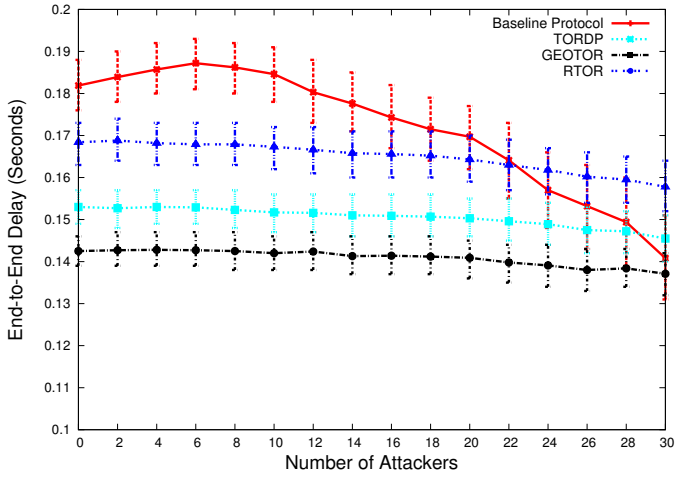

Figure 3.7: End-to-end delay by changing the number of malicious nodes

probability, and the distance progress. Incorporating the distance progress as a dynamic parameter in candidate selection results in having a smaller delay in delivering packets. Amongst protocols which have a defensive mechanism against malicious nodes, RTOR has the highest amount of end-to-end delay. The reason is that RTOR does not consider the location of nodes when selecting candidates. According to RTOR, as described before, nodes select the most trustworthy and reliable nodes as their candidates. Such nodes are normally the ones that have a reasonable trust value, and are close enough to the source node. This results in a higher hop count and, as a result, a higher end-to-end delay. The end-to-end delay for RTOR, GEOTOR, and TORDP shows a slightly decreasing trend by increasing the number of malicious nodes. The reason is that by growing the number of malevolent nodes, higher number of packets are captured and dropped by such nodes, and smaller number of packets are received by the destination nodes. In fact, packets that need to travel longer distance suffer from the higher probability of being captured by malicious nodes. In contrast, the situation is reversed for packets that do not need to travel long to arrive their destination. This means a shorter amount of time is consumed for packet transmission and candidate coordination, which results in a smaller end-to-end delay.

Fig. 3.8 shows the end-to-end delay of packets for the second scenario, where the density of nodes changes. As observed, GEOTOR shows the best performance compared to TORDP and RTOR. Overall, the end-to-end delay of delivering pack-

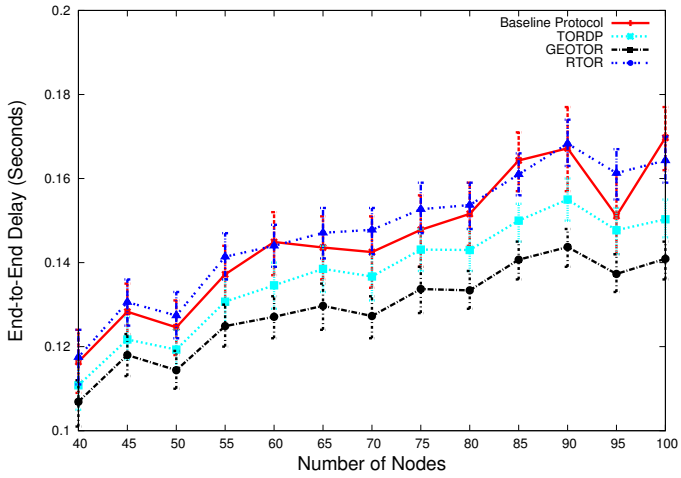

Figure 3.8: End-to-end delay by changing the number of nodes

ets is increasing as the number of nodes grows. This occurs as the consequence of delivering greater number of data packets to their destination. In fact, having a higher density of nodes not only provides the opportunity of creating higher variety of routes towards the destination, but also decreases the amount of obstructions in the wireless channel. Therefore, all of the protocols, especially the ones being able to detect and replace malicious nodes, will be able to deliver larger number of packets to the destination, which leads to a greater end-to-end delay. Furthermore, the trend of changes in Baseline Protocol follows a similachapter:trust:sec:r trend as that of the RTOR. The reason is that both protocols utilize the same method for selecting their candidates.

Finally, Fig. 3.9 shows the end-to-end delay of nodes under the third simulation scenario, where the number of source-destination data flows changes. As observed, similar to the previous scenarios, GEOTOR demonstrates the best performance when it comes to comparison with other protocols, owing to its candidate selection method. TORDP and RTOR reasonably rank as the second and third protocols. Baseline protocol, however, has the worst end-to-end delay due to the fact that it requires higher number of retransmissions, and greater amount of coordination delay. In fact, if a malevolent node is selected as the highest-priority node in the candidate set, all of the packets will have to be sent through the second-priority candidate after the coordination delay is passed. Moreover, if the benign candidate does not receive a particular packet, a retransmission needs to take place by the

previous hop, which leads to a higher amount of delay. Additionally, the end-to-end delay for all protocols is reasonably high when there is only one connection between nodes. However, similar to the packet delivery ratio, the trend of changes in the delay becomes almost constant by increasing the number of connections.

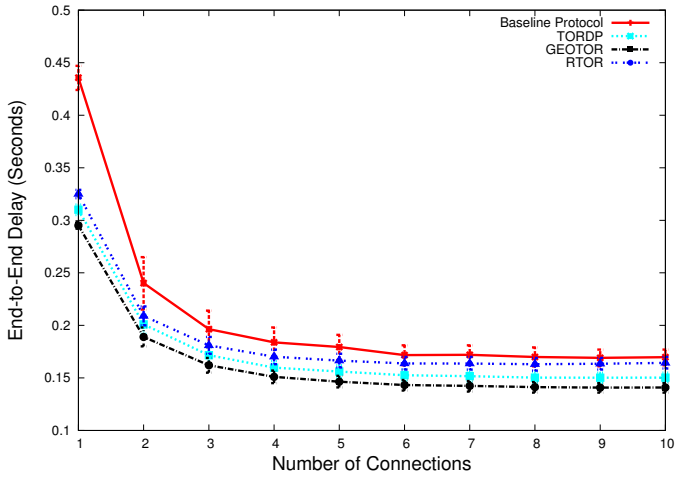

Figure 3.9: End-to-end delay by changing the number of source-destination data connections

- **Hop Count** The average hop count of sending packets to their destination is the last evaluated parameter in this section. Fig. 3.10 represents the hop count for the first scenario, when the number of attacker nodes changes. As shown, by increasing the number of malicious nodes, hop count decreases. This is due to the fact that by growing the number of malicious nodes, the probability of a packet drop also grows. To be more specific, packets which need to travel a longer distance have a higher chance to be captured by malicious nodes, whereas the situation is exactly opposite for other packets. This will lead to a smaller hop count for all algorithms. Regarding protocols that benefit from the proposed trust model, however, the effects of attacker nodes can be gradually cancelled by re-evaluating their trust value and replacing them with other nodes. Therefore, greater number of packets will have the chance to arrive their destinations compared to the Baseline protocol. It is also interesting to observe that GEOTOR and TORDP operate more efficiently compared to other protocols. In fact, not only such protocols deliver almost higher number of packets to destinations compared to the Baseline

protocol, but also such packet are delivered with a lower average hop count. More specifically, even having 30 malicious nodes, when packets with a short distance between the source and the destination in the Baseline protocol have a higher likelihood to be successfully delivered, such packets travel from larger number of intermediate hops compared to GEOTOR and TORDP.

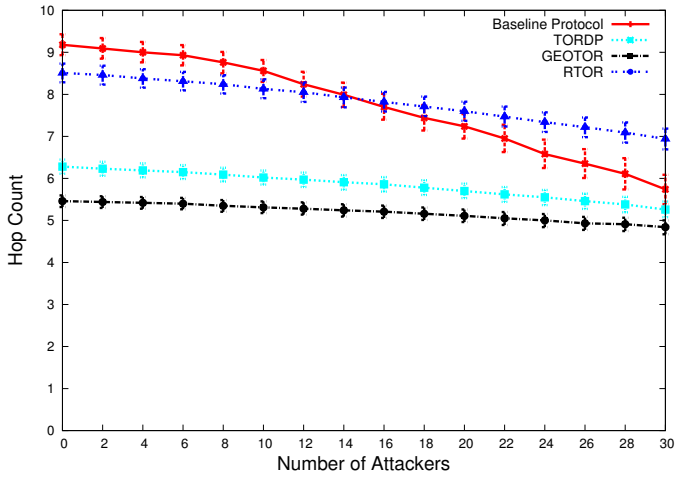

Figure 3.10: Hop count by changing the number of malicious nodes

Fig. 3.11 shows the effect of changing the number of nodes on hop count following the second scenario. Clearly, GEOTOR has the best hop count whereas RTOR has the worst as a result of their various candidate selection methods. The average hop count for GEOTOR and TORDP shows almost the same figures when the number of nodes is smaller than 60. According to the size of the simulation area, this situation resembles the case that the distance between source and destination nodes is not too far, and packets can reach their destination in smaller than four hops. As the number of nodes increases, however, greater packets can reach their long-distance destinations due to the creation of new routes. Obviously, such packets need to travel from more hops to be finally delivered to their destination. Nevertheless, GEOTOR outperforms TORDP in terms of hop count when the number of nodes increases. Similarly, the hop count for RTOR is higher than the Baseline protocol. The reason is that hop count is calculated for packets that are successfully delivered to their destination and RTOR can handle delivery of larger number of packets compared to the Baseline protocol. This conveys that

some of the maliciously dropped packets can be delivered to their destinations in RTOR, after the malicious intermediate node is replaced by a benign one.

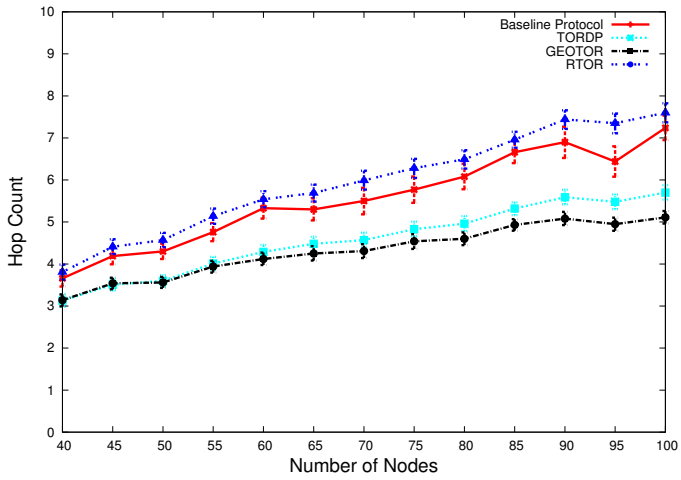

Figure 3.11: Hop count by changing the number of nodes

Finally, Fig. 3.12 demonstrates the effects of changing the number of data connections on hop count. As the figure shows, the performance of GEOTOR algorithm is higher than TORDP and RTOR, while RTOR shows the worst performance. For example, when the number of connections is only one and the distance between the source and the destination is more than 1400 meters, the average hop count for RTOR is about 16.6 whereas this number is around 13, and 11.5 for TORDP and GEOTOR, respectively. In contrary, the packet delivery ratio for RTOR is the highest, while this measure is the lowest for GEOTOR. This represents that sending packets from more reliable routes may result in a higher hop count and delay. The hop count for packet transmissions tends to be almost a constant value as the number of source-destination data flows increases and higher number of randomly-generated connections are created in the network.

To sum up, simulation results show that the proposed trust model can significantly improve the performance of delivering packets to their destination, when malicious nodes exist in the network. Furthermore, amongst the proposed metrics for candidate selection, it is observed that RTOR outperforms GEOTOR and TORDP in terms of packet delivery ratio, whereas it is the worst metric when it comes to the end-to-end delay and hop count. This conveys that there is a trade off between delivering larger number of packets to their

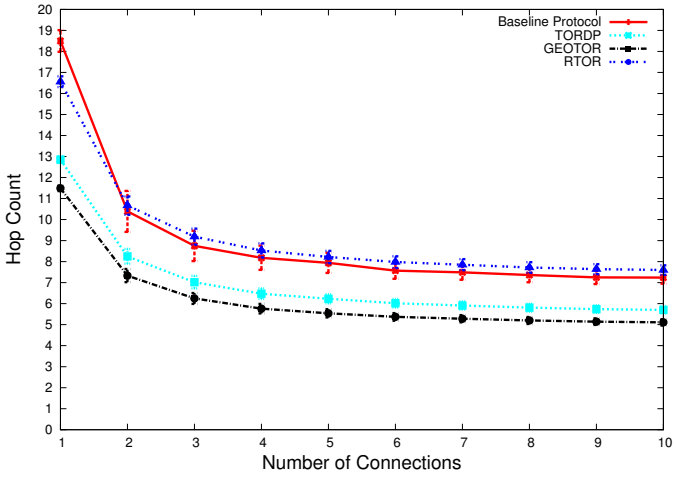

Figure 3.12: Hop count by changing the number of source-destination data connections

destination through more reliable routes, and delivering packets with smaller amounts of delay through less reliable paths. Therefore, it is concluded that selecting the suitable candidate selection metric may be different for different applications depending upon various requirements, characteristics, and priorities.

3.2 Comprehensive Reputation-based OR Protocol

In this section we propose to extend the introduced trust model to utilize not only the direct interactions between nodes, but also recommendations of others. A detailed description of this extension is explained in the following subsections.

3.2.1 Reputation Calculation

As mentioned in Chapter 2 Section 2.2.2, reputation or trust in wireless networks consists of different aspects such as experience-based (direct) and recommendation-based (indirect) reputation. As explained in Section 3.1, in order to calculate experience-based trust, nodes in different time intervals evaluate the behavior of their peers using a monitoring or watchdog mechanism and compute a measure representing the behavior of such nodes in the mentioned interval. This most recent reputation value is then incorporated with the old reputation value and a final experience-based reputation is obtained. The idea of indirect reputation states that nodes should be able to benefit from the experience

of other nodes in the network. This can be done if they periodically propagate their calculated direct reputation value in the network. In this case, third-party nodes can benefit from such recommendations by combining this information with their own direct trust and finally calculating a comprehensive reputation value. This value is then fed into the candidate selection module to be used for choosing candidates.

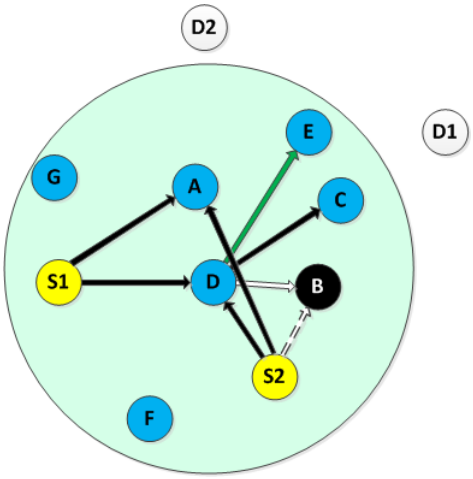

Figure 3.13: A Sample Topology of Network

Figure 3.13 represents how using both direct and indirect observations can help in detecting and isolating malicious nodes using OR protocols. In this scenario, suppose that node $S1$ first starts sending packets towards $D1$. For this, $S1$ and its candidates, say $\{A, B\}$, have to select their own candidates. Assuming that $\{B, C\}$ is the initial candidate set for node D and B is a malicious node, D will be able to detect B using direct interactions, and will tend to replace this node with E. Once D re-evaluates the trust value of its candidates and replaces B, it can then send a reputation report to all of its neighbours and inform them about the behavior of its candidates. As one of the potential nodes in the neighbourhood, $S2$ receives a copy of recommendations coming from D and recalculates the reputation of B. Therefore, $S2$ can benefit from the experience of D, and probably prevent from communicating with the malicious node.

Calculation of Recommendation-based Trust

To benefit from the experience of others, nodes in the proposed model propagate their calculated reputation value to their one-hop neighbours. Upon receiving such packets,

neighbour nodes accumulate such recommendations locally and compute an indirect reputation value for all possible nodes using the following equations.

$$Trust_{i,j(k)}^{indirect} = Trust_{final}^{direct}(k,j) \tag{3.14}$$

$$Trust_{final}^{indirect}(i,j) = \sum_{k=1}^{n} \frac{Trust_{i,j(k)}^{indirect} \times Trust_{final}^{direct}(i,k)}{n} \tag{3.15}$$

In the Equation 3.14, $Trust_{i,j(k)}^{indirect}$ accounts for the average of the reputation values that node k has reported to node i through its own experience with node j. Some nodes, however, may not provide honest reputation reports and try to increase or decrease the reputation value of other nodes. In order to defend against such nodes, provided reputation value from neighbour nodes is discounted by their current direct trust or reputation value. Finally, by combining reported values from all neighbors, the recommendation-based reputation value is calculated using Equation 3.15, where n represents the number of neighbor nodes having provided their recommendations.

Comprehensive Trust

Once nodes calculate the final experience-based as well as recommendation-based trust values, it would be possible for them to compute the final trust value for others. As shown in Equation 3.16, this can be simply done considering the following weighted average equation. In some situations, however, if there is no recommendations from other nodes for some candidates, nodes only use their current direct trust for candidate selection.

$$Trust_{i,j}^{final} = \phi \times Trust_{final}^{direct}(i,j) + (1-\phi) \times Trust_{final}^{indirect}(i,j) \tag{3.16}$$

A General Monitoring Module

The monitoring module or watchdog component plays an important role in the performance of trust and reputation management systems. In fact, all the calculations regarding reputation calculation module depend upon an efficient monitoring or watchdog module, that is responsible to supervise the behavior of candidate nodes, and to properly compute the values of α and β. For this purpose, an extended version of the monitoring module that was introduced in Section 3.1.3 is proposed, which operates

based on the existence of a perfect coordination between candidate nodes in the candidate set. As explained in Chapter 2.1, prefect coordination guarantees that if one of the candidate nodes transmitted a packet, all of the other nodes in the candidate set as well as the sender node will receive an acknowledgement and prevent from forwarding the same packet repeatedly [25]. Considering the existence of perfect coordination between candidates, each forwarder node will be able to calculate the probability that each of its candidates will forward a packet. This has been shown as $Prob^{S}_{c_i}$, in Equation 3.17.

$$Prob^{S}_{c_i} = p_{S,c_i} \times \prod_{j=1}^{i-1}(1 - p_{S,c_j}) \tag{3.17}$$

The above equation demonstrates that, by sending a packet, node S as the sender node expects the highest-priority candidate node to forward the packet. This candidate might receive the mentioned packet with probability p_{S,c_1}, corresponding to the link delivery probability between nodes S and c_1. On the other hand, the next-priority candidate is expected to forward the packet only if the highest-priority candidate does not receive it. A similar explanation can be provided for other lower-priority candidates. It is also worth noting that node S expects all of the candidates to cooperatively forward packets if they receive them. Therefore, node S will be able to calculate the number of packets it expects each candidate to forward towards the destination. This can be simply done by keeping the track of the number of packets sent to the candidates within the last Δt. By comparing the expected number of packets that a candidate should send with the number of packets that it has actually sent, nodes will be able to calculate the values of α and β. Afterwards, such values are used as explained to calculate the reputation of candidate nodes, and to perform the process of candidate selection.

3.2.2 Simulation Study

In order to assess the performance of the proposed comprehensive trust model, an extensive simulation-based analysis has been conducted using Network Simulator (ns-2.35) [68] by changing different parameters. Overall, it is observed that the proposed model is able to properly detect malicious nodes in the network and to considerably cancel the effect of such nodes. In the following sub-sections, the network environment and parameters for this set of experiments are first discussed and then some of the obtained results are investigated.

Table 3.4: Simulation Parameters

Parameter	Value
Propagation Model	Shadowing
MAC	802.11
Number of Nodes	100
Field Size	$(1000s1000)$
Number of Candidates	2
Maximum Retransmissions	
Transmission Rate	4 Packets/Second
Trust Calculation Interval (Δt)	20 Seconds
Default Trust Value	0.8
Direct-reputation's Weight (ϕ)	0.7
Coordination Delay	15 Milliseconds
Simulation Time	600 Seconds

Network Environment

For the purpose of simulating a realistic wireless network where some amount of signal loss exists in the channel, as observed in Table 3.4, the shadowing propagation model has been used with standard parameters for outdoor urban environments following [68]. Furthermore, to simulate the effect of the proposed reputation model, three different parameters have been selected and changed containing the number of malicious nodes, the weight of the previously calculated reputation value (ω), and the reputation ageing factor (θ). When it comes to the simulated data communication traffic, 10 different source-destination CBR traffic patterns have been generated in the network between nodes, having been located randomly in the simulation field. Here too, the link delivery probability between nodes is calculated as a function of distance between them following [60] as discussed in Section 3.1.4. A list of the simulation parameters can be found in Table 3.4.

Apart from the simulation parameters, however, any trust and reputation model should be evaluated in presence of adversary nodes. To keep the attack scenario simple, Black-hole attack [63] has been selected in which malicious nodes, that are unknown at the beginning, might be selected as candidate nodes. Simulation results represent

how such nodes can significantly reduce the efficiency of communications, and how the proposed reputation model is able to significantly resist against such nodes.

Results

Results of the proposed comprehensive trust management protocol are discussed in this subsection. In the represented graphs, three different protocols have been depicted labelled as Optimum, Baseline, and Proposed algorithms. The Optimum protocol accounts for the situation that there is no malicious node in the network and all nodes in the network are cooperative. The Baseline protocol represents the presence of attacker nodes in the network, whereas benign nodes have no defensive strategy against them. Finally, the proposed model shows the effect of the proposed model in a realistic network, where malicious nodes attack the network and benign nodes try to detect such nodes using the proposed reputation system. In all of the represented results, the Optimum protocol reasonably ranks first amongst the three, the Baseline protocol is recognized as the worst, and the Proposed solution ranks second.

- **Effect of Ageing Factor**

  The effect of changing the trust ageing factor on packet delivery ratio has been depicted in Fig. 3.14, where the weight of the old trust (ω) is 0.7 and the number of malicious nodes is 20 representing 20% of malicious nodes in the network. The ageing factor axis represents the discount factor being considered for the formerly calculated reputation value as explained in Section 3.1.2. As observed, the Optimum as well as Baseline protocols are not affected by this parameter as they are not equipped with a reputation management system. Regarding the proposed method, however, the effect of having a forgetting factor can be clearly seen. Since nodes recompute the trust of other nodes in different Δt time intervals and impose the ageing factor on the previously calculated reputation, selecting a reasonable value for this parameter seems to be important.

  It is interesting to see that when the value of θ is low, the effect of the proposed reputation system decreases. This is due to the fact that a considerable proportion of the obtained reputation value is forgotten in each round of reputation recalculation. This means that, since the weight of the old reputation is higher than the most recently calculated one, as time goes the trust value of all nodes is significantly

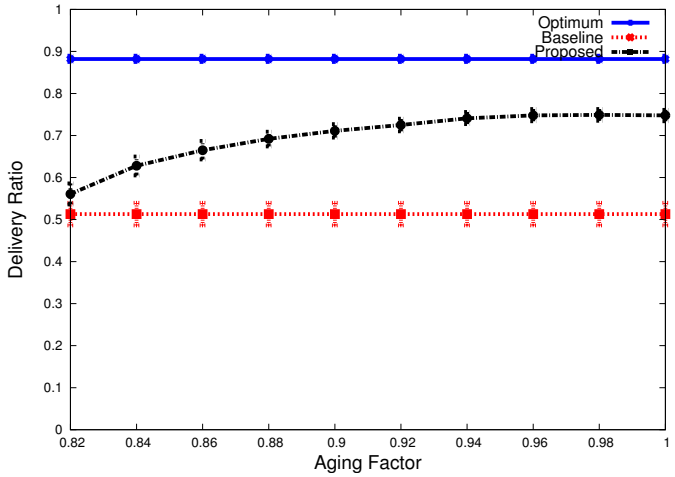

Figure 3.14: Effect of changing reputation ageing factor on delivery ratio

decreased, which results in reducing the effect of the proposed mechanism. As the ageing factor increases, however, the effect of the proposed model becomes more observable. Furthermore, since in the implemented scenarios nodes do not change their behavior in the course of time, higher values of the ageing factor represents a better performance in terms of packet delivery ratio. More specifically, in the simulated scenarios, a malicious node is always punished by other nodes and its reputation level is dramatically decreased. Similarly, a benign node always acts as a trustworthy node and collaborates in packet forwarding. Therefore, in such a scenario, having no discount will be more beneficial for benevolent nodes and this explanation is confirmed by simulation results.

- **Effect of Old Reputation's Weight**

  Fig. 3.15 shows the effect of changing old reputation's weight (ω), when the reputation ageing factor (θ) and the number of malicious nodes are set to 0.98 and 20 respectively. In this plot, when the value of ω is set to zero, it means nodes only use their recently calculated trust and old trust values are given no weight. On the other hand, setting this value to one means nodes do not benefit from recently calculated trust values and only use the initial trust value for candidate selection. The latter case resembles the situation that nodes are not equipped with a reputation system. The reason is that recent trust values are calculated but not utilized properly. Here too, since the behavior of nodes does not change in the course of

time, the more nodes use recently calculated reputation values reasonably leads to a greater packet delivery ratio. However, if nodes change their behavior between collaborative and malicious, the effect of using the old trust weight becomes more clear. In this situation, nodes need to look at the entire history of communications, and assess the reputation of other nodes by incorporating both recent and old reputation values.

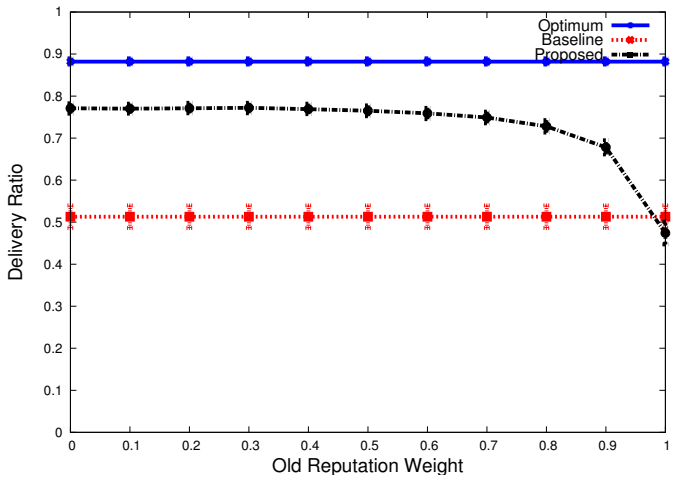

Figure 3.15: Effect of changing old reputation on delivery ratio

- **Effect of Malicious Nodes**

  Fig. 3.16 shows the effect of changing the number of attackers on delivering packets to their destination. As observed, in Baseline protocol by increasing the number of attackers, the packet delivery ratio dramatically decreases so that having 40% of nodes as malicious ones, the delivery ratio of packets falls down to 20%. The situation, however, for the proposed method is completely different and nodes are able to deliver larger number of packets to their destination in presence of malicious nodes. More specifically, having 40% of nodes as attackers, the delivery ratio is still around 57%. This conveys that the proposed protocol is able to successfully detect and significantly cancel the effect of malicious nodes using the suggested comprehensive reputation model.

  End-to-end delay of delivering packets to their destination when changing the number of malicious nodes is shown in Fig. 3.17. It should be emphasized that the end-to-end delay is only computed for packets being successfully delivered to their

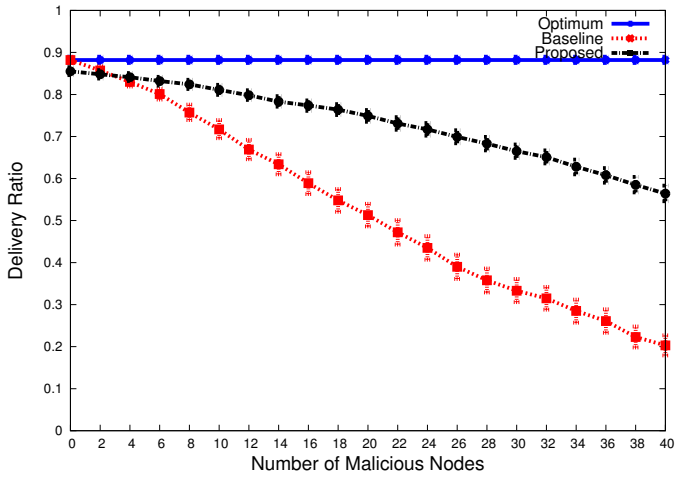

Figure 3.16: Effect of changing the number of attackers on delivery ratio

final destination. As observed, the delay for Baseline protocol increases first by increasing the number of malicious nodes, but starts to decrease as the number of attackers continues to grow. This behavior is normal owing to the fact that by increasing the number of malicious nodes up to a certain level, it is still possible for the Baseline protocol to deliver a considerable number of packets to their destination. This occurs, for instance, in situations where the first candidate is a malicious node and drops packets but the second one is benign and forwards packets after the coordination delay is passed. This situation leads to an increase in the end-to-end delay. By escalating the number of malicious nodes, however, smaller number of packets will be able to reach their destination. Such packets are the ones that need to travel a short distance between the source and the destination. Clearly, the higher the distance between source and destination nodes, the higher the probability of dropping such packets by malicious nodes.

3.3 Summary

In the first section of this chapter, a novel trust model was proposed, designed, and implemented for OR protocols that benefits from both reliability of OR protocols and security of trust management systems. A new watchdog mechanism was also designed and customized that is applicable to OR protocols. Furthermore, according to different important parameters for candidate selection in OR protocols such as trust value

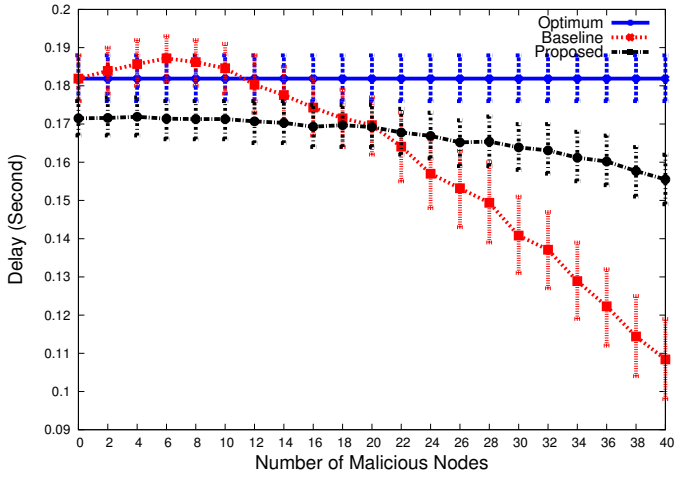

Figure 3.17: Effect of changing the number of attackers on end-to-end delay

of neighbor nodes, link delivery probability of neighbors, and the amount of achievable distance progress through each neighbor, three different candidate selection metrics were proposed known as RTOR, GEOTOR, and TORDP. Finally, the proposed model was implemented and evaluated using network simulation. Results represented that proposed model performs well in a hostile environment, where malicious nodes try to drop data packets and decrease the network performance. Additionally, considering different proposed candidate selection metrics, it is concluded that there is a trade off between delivering larger number of packets to their destination from more reliable paths, and sending packets through shorter but less reliable routes.

In the second section, that is am extension to the proposed model developed in the first section, the effect of indirect interactions between nodes in the calculation of trust was investigated, developed, and added to the proposed trust model. Furthermore, the watchdog mechanism was extended to be a general and comprehensive component that is independent of the number of nodes in the candidate set. Simulation results demonstrated that the proposed comprehensive trust model is efficiently able to defend against malicious nodes, when the number of such nodes increases in the network.

Chapter 4

Modeling and Evaluation of Security Challenges in Opportunistic Routing Protocols

In Chapter 3, we proposed a trust calculation model to defend against malicious nodes in OR-based wireless networks. This chapter, however, focuses on modeling and simulation of routing attacks for OR protocols in greater details. As explained in previous chapters, malicious nodes can significantly downgrade the performance of wireless networks. Therefore, it would be of high significance to analytically model the effects of such nodes. In this chapter, the Discrete-Time Markov Chain (DTMC) is utilized to analytically model the existence of malicious nodes in the network and to evaluate their effects. Section 4.2 demonstrates the proposed model in greater details and elaborates its possible applications. Section 4.3 includes a performance evaluation on the introduced model and, finally, Section 4.4 provides a summary for this chapter.

4.1 Applications of the Proposed Model

This section includes some of the most important scenarios in which the proposed model can be applied. To be more specific, the introduced approach is a general and comprehensive analytical model that can be utilized in order to assess the performance of any wireless sensor or mesh network in realistic situations. In fact, although the focus of this chapter is to investigate the behavior of malicious nodes in OR protocols, the

proposed model can be applied to simulate any hardware and software node failure that can occur in a wireless network. For example, in a WSN [99, 30], nodes have a limited amount of energy, and will crash once their energy is completely consumed [43, 7, 114]. In this scenario, dead nodes will have a similar role to malicious nodes in the proposed model, due to the fact that such nodes will not be cooperating in the packet forwarding process [37, 36]. On the other hand, as stated in [13] and [134], OR protocols outperform traditional routing protocols when it comes to network performance. Furthermore, traditional routing protocols for wireless networks can be considered as a special case of OR protocols, in which each node selects only one node as its next hop forwarder. Therefore, the introduced model can assist in a realistic study of any wireless network, including malicious behaviors or node failures. Some applications of the proposed model can be summarized as follows.

- **Highly Sensitive Networks:** As stated in [5, 112, 24], sensor networks have been widely used in highly sensitive situations such as military applications for detecting noise, light, chemicals, explosions, etc. in the area of interest. Consequently, for such mission critical situations, it is highly important to deploy a network that is effective and operational in hostile environments. For example, there is always a possibility that some of the sensor nodes in the battlefield become compromised or get destroyed. The proposed model in this chapter is able to effectively simulate the existence of compromised or broken sensors in a military sensor network. More precisely, the proposed model can simulate the creation of any number of sensors, when a proportion of them is not functioning properly. This will create the possibility of evaluating the performance of such a network, and in the creation of a network that is tolerant to node failures, or misbehaviors in adversary environments.

- **Underwater Networks:** Underwater sensor networks are a promising subsection of WSNs [39] with the purpose of exploring and investigating the world beneath the water's surface [4]. However, due to the highly dynamic environment and unstable physical conditions, a suite of more reliable routing protocols must be designed and developed for underwater sensor networks. [113] indicated that OR protocols can be appropriate solutions for addressing the unreliability of such networks, as they can boost the reliability of communications in noisy and unstable environments. On the other hand, maintaining and providing technical support for underwater sensor networks is a considerably more complex task compared to regular sensor networks.

Moreover, sensors are at high risk of getting damaged or lost in such environments. Therefore, investigating and studying the effects of malfunctions in such networks before the actual deployment can assist in a more sophisticated and realistic sensor deployment. The proposed model in this chapter can effectively simulate the existence of node failures in underwater sensor networks. It should also be noticed that, using the introduced analytical model, the extraction of network parameters such as packet delivery ratio, drop ratio, hop count, etc. will be less computationally expensive, and more flexible than performing network simulations.

- **Environment Monitoring or Sensing Applications:** As cited in [73, 42] and [9], wireless sensor networks (WSNs) and Internet of Things (IoT) have been or will be widely used in many applications including, but not limited to, intelligent transportation systems, target tracking, manufacturing, wildlife monitoring, data gathering, healthcare systems, and so forth [107]. In each of the mentioned applications, wireless nodes might fail to connect to other peers, or might be compromised. Therefore, considering that the proposed protocol in this chapter is a general-purpose model that is applicable in any wireless sensor or mesh network, it can be used to evaluate the performance of such networks in realistic situations. This can not only help in performing risk management, but also in building a more reliable, secure, and fault tolerant network [45, 67].

4.2 Modeling Routing Attacks using DTMC in OR Protocols

This section describes the proposed model in which an OR-based wireless network containing malicious nodes is modeled using DTMC. As explained in [61], hop-by-hop routing process of OR methods creates the possibility of modeling an OR-based wireless network using DTMC, if the coordination method between candidates is perfect. In the following section, applications of the proposed model are first described. In continue, an overview of modeling OR protocols using DTMC is described in Section 4.2.1. Afterwards, a complementary model is proposed in Section 4.2.2 which includes and models the effect of malicious nodes in OR protocols. Section 4.2.3 includes more details on how to calculate transition probabilities between states of DTMC in the proposed model. Finally, a novel

method of calculating packet drop ratio by malicious nodes is proposed and calculated in 4.2.4.

4.2.1 Modeling OR using DTMC

As explained in Chapter 1, according to the link quality between the transmitting node in OR protocols and each of its candidates, some candidates may receive the mentioned packet; otherwise the sending node tends to retransmit the packet a certain number of times if no candidates receive it. Assuming that a perfect coordination mechanism is applied between candidate nodes, and considering node priority in the candidate set, only one of the candidates will forward the packet, while others discard it. This process continues until the packet reaches its destination successfully, or gets discarded in a node after being retransmitted for the maximum number of times.

Routing operations in OR methods can be modelled accurately using DTMC, according to its characteristics described in the remainder of this section. First of all, DTMC is a memoryless system, meaning that reaching a specific state in DTMC is independent of past states. This situation applies to OR protocols in which packet arrival at any given time is independent of previous hops, and the current hop attempts to forward the packet to the next hop. Second, in DTMC the state of the system changes given probability values between different states. This is also applicable in OR protocols, where a probability value for each node in the candidate set determines if each candidate will act as the relay node and will progress the packet. Finally, a DTMC with two absorbing states is basically similar to an OR protocol with two absorbing states. Such states can be considered as a *Success* state, which resembles successful delivery of the packet to its final destination, and a *Fail* state, which represents the situation where a packet gets discarded after a maximum number of retransmissions. For the purpose of modeling an OR protocol using DTMC, perfect coordination between candidate nodes should be applied. Furthermore, each state in the DTMC is defined using a tuple, which contains the node identifier and the number of occurred retransmissions in that specific node. The proposed model is valid for an OR containing any number of candidates or retransmissions, any topology, or candidate selection algorithm, as described in [61].

4.2.2 A DTMC Model for Black-hole Attack

Although the proposed model in [61] is a general model for assessing the performance of an OR-based wireless network, such a model is not able to represent realistic scenarios in which the network may contain uncooperative or malicious nodes. More specifically, there are numerous situations in which nodes do not participate in routing operations as expected. This may occur due to a hardware or software failure, malicious purposes, and so forth. In this section, we propose a modified model of an OR protocol that uses DTMC in the presence of uncooperative nodes. It should be emphasized that an uncooperative node is considered a node which, for any reason, does not collaborate in routing operations according to the specification of an OR protocol. More specifically, it is possible to assign a probability value to a malicious node indicating its ratio of willingness to cooperate. In this chapter, however, it is assumed that malicious nodes will always behave maliciously if they are supposed to take any action. Table 4.1 contains symbols and notations that are used in the entire chapter.

The main idea of the proposed model in this chapter is the existence of M malicious nodes in the network. In fact, the focus of the proposed model is on modeling black-hole attacks in the network. Similarly, it would be possible to generalize the model, to encompass any uncooperative behavior in which the malicious node is assigned with a probability value regarding its ratio of cooperation. As mentioned in Chapter 2 Section 2.2.1, a black-hole node receives all of the data packets and drops them maliciously. Basically, when a node selects a black-hole node as one of its candidates, it expects the malicious candidate to forward received packets according to its priority in the candidate set. Surprisingly, however, the malicious node drops such packets and, following perfect coordination, sends an acknowledgment message to all other candidates indicating that it has already forwarded every single packet. Therefore, the previous hop and all of the other candidates will prevent the forwarding of such packets, and they will be permanently lost.

In order to simply describe the proposed model, a linear topology is studied in which $N = 5$, $K = 3$, $M = 1$, and $C = 2$. In this simple model, it is also assumed that the distance between all nodes is equal, and nodes with $(ID = 0)$ and $(ID = 4)$ are the source and the destination, respectively. In this scenario, it is also assumed that node with $(ID = 2)$ is the only malicious node that follows black-hole attack. This scenario conveys that the malicious node will drop all of the received packets upon receiving them.

Table 4.1: Notation and Symbols

Symbol	Description
N	Number of nodes in the network
M	Number of malicious nodes
K	Maximum number of allowed retransmissions
C	Maximum number of candidates
$\text{CS}_{i,dest}$	Candidate set of node i for destination $dest$
S	Number of states in DTMC
P	Transition probability matrix
Q	Transition probability matrix between transient states
R	Transition probability matrix between transient and absorbing states
I	Transition probability matrix between absorbing states
Z	Transition probability matrix between absorbing and transient states
V	DTMC's initial state
F	Fundamental matrix of Markov process
ID	Node identifier
ReTx	Number of retransmissions taken place so far
(ID, ReTx)	A state in the DTMC
$p^{(i,j)}_{(i',j')}$	Transition probability between states (i,j) and (i',j')
c_i	The i_{th} priority candidate
$link_{prob}(x,y)$	Link delivery probability between nodes x and y

Such nodes can therefore be modeled as absorbing states in DTMC. More specifically, once the system reaches an absorbing state, it will stay in that state, and no more transitions between states will occur. Therefore, as shown in Figure 4.1, since the node with $(ID = 2)$ is a packet dropper, neither an attempt to forward the packet towards the destination nor a retransmission will take place once the system reaches state $(2, 0)$. Considering this, as well as the existence of M malicious nodes, it would be possible to calculate the number of states in the system, say S, using Equation 4.1.

$$S = (N - M - 1) \times (K + 1) + M + 2 \tag{4.1}$$

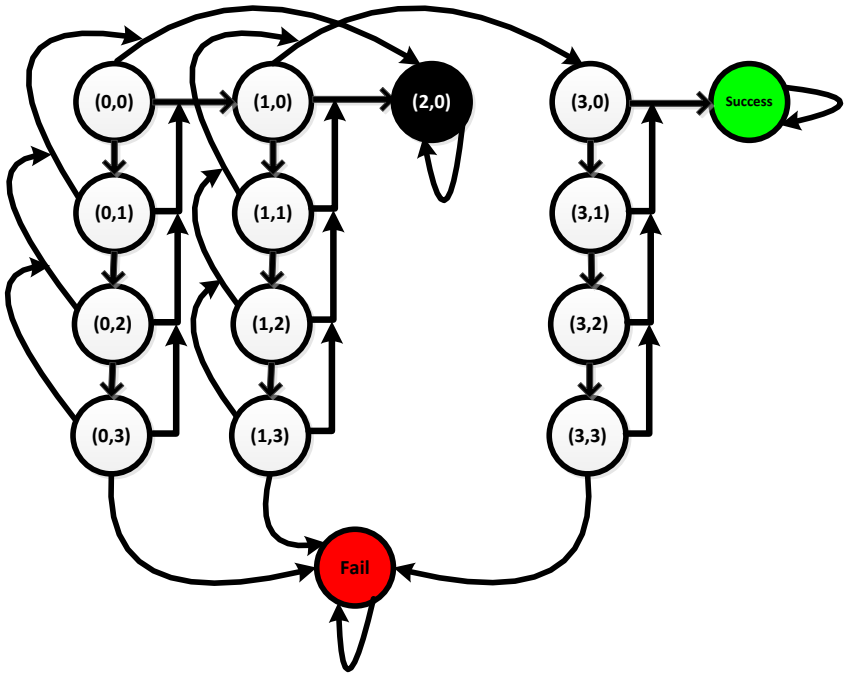

Figure 4.1: An example of a DTMC for a linear topology in the presence of a black-hole node ($N = 5$, $M = 1$, $K = 3$, $C = 2$)

As shown in Equation 4.1, the number of absorbing states will be equal to $M + 2$, corresponding to M number of malicious nodes, one *Fail*, and one *Success* state. Furthermore, the number of transient states in the proposed model would be $(N - M - 1) \times (K+1)$. Finally, once all of the the states in a DTMC are known, it would be possible to create a stochastic matrix containing the transition probabilities between states; using that matrix, we can extract required network parameters such as packet delivery ratio, drop ratio, etc.

The transition probability matrix P is an $S \times S$ matrix which corresponds to the modeled DTMC in the presence of malicious nodes. This matrix is presented in canonical form in Figure 4.2. As observed, P is composed of four different sub-matrices. Q, which demonstrates the probability of transitioning between transient states, is a matrix with $[(N - M - 1) \times (K + 1), (N - M - 1) \times (K + 1)]$ dimensions. Matrix R, which shows the probability of the transition from transient states to absorbing states, is a $[(N - M - 1) \times (K + 1), (M + 2)]$ matrix. The bottom-left corner of P, shown as Z, is a

$$P = \begin{bmatrix} Q = \begin{bmatrix} p_{(0,0)}^{(0,0)} & p_{(1,0)}^{(0,0)} & \cdots & p_{(N-M-1,K)}^{(0,0)} \\ p_{(0,0)}^{(1,0)} & p_{(1,0)}^{(1,0)} & \cdots & p_{(N-M-1,K)}^{(1,0)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(0,0)}^{(N-M-1,K)} & p_{(1,0)}^{(N-M-1,K)} & \cdots & p_{(N-M-1,K)}^{(N-M-1,K)} \end{bmatrix} & R = \begin{bmatrix} p_{BH_1}^{(0,0)} & \cdots & p_{BH_M}^{(0,0)} & p_{Fail}^{(0,0)} & p_{Dest}^{(0,0)} \\ p_{BH_1}^{(1,0)} & \cdots & p_{BH_M}^{(1,0)} & p_{Fail}^{(1,0)} & p_{Dest}^{(1,0)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ p_{BH_1}^{(N-M-1,K)} & \cdots & p_{BH_M}^{(N-M-1,K)} & p_{Fail}^{(N-M-1,K)} & p_{Dest}^{(N-M-1,K)} \end{bmatrix} \\ Z = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} & I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \end{bmatrix}$$

Figure 4.2: The transition probability matrix

$[(M+2),(N-M-1)\times(K+1)$ matrix and represents the probability of transitioning between absorbing and transient states. According to DTMC, logically this matrix is filled with all zero elements; the reason is that once the system is in an absorbing state, it remains in that state permanently, and no more transitions occur. Finally, I, which is an identity matrix of $[(M+2),(M+2)]$ dimensions, demonstrates transition probabilities between absorbing states.

In some network situations particularly when the number of malicious nodes is significant and corresponds with the candidate selection algorithm, it is possible that all of the candidates for a specific node will be selected as malicious nodes. In this case, all of the packets being sent by the sending node will be maliciously dropped by all of the candidates. More specifically, the probability of reaching the destination node, represented as *Success* state in DTMC, will be zero. This is due to the fact that no path will exist between the source and the destination nodes. This scenario has been depicted in Figure 4.3, in which all candidates of node 0, say nodes 1 and 2, are malicious nodes. In this scenario, regardless of the number of retransmissions by node 0, all of the sent packets will be captured and dropped by its candidates, and no packet will have the chance to arrive at its destination.

4.2.3 Calculating Transition Probability Matrix

Once all states of the DTMC are recognized and the dimensions of each matrix are known, it should be possible to calculate the probability values of each element in P.

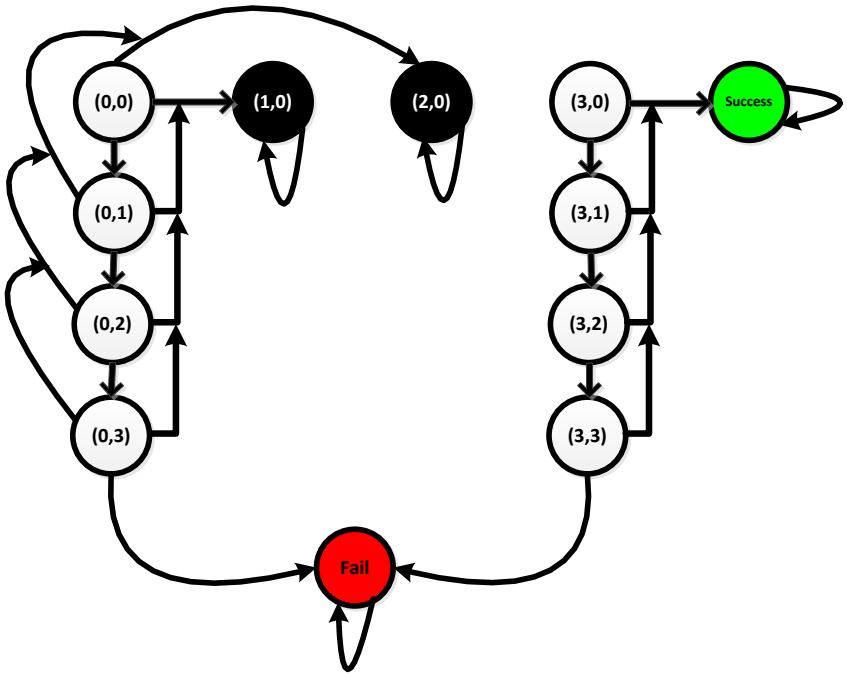

Figure 4.3: The DTMC for a linear topology in the presence of two black-hole nodes ($N = 5$, $M = 2$, $K = 3$, $C = 2$)

As previously stated, the transition probability value between states (i, j) and (i', j') is defined as $p^{(i,j)}_{(i',j')}$ where i and i' represent node identifiers, whereas j and j' stand for the number of occurred retransmissions in each node. To obtain the probability values, different situations should be considered, as inspired by [61]. It should also be mentioned that all of the calculations are independent of any network topology, and are valid for any OR-based wireless network regardless of the number of candidates or retransmissions.

- **Reaching a state corresponding to the highest-priority candidate:** This case demonstrates the probability of transitioning to a state in DTMC which corresponds to the highest-priority candidate in the candidate set. For example, $p^{(0,0)}_{(2,0)}$ or $p^{(1,0)}_{(3,0)}$ in Figure 4.1 is calculated following this rule. This probability value is basically equal to the link delivery probability between node i and its highest-priority candidate, say c_1, in the candidate set, as specified in Equation 4.2.

$$p^{i,j}_{c_1,0} = link_{prob}(i, c_1) \tag{4.2}$$

- **Reaching a state corresponding to other candidates (except for the highest-priority candidate):** In this case, the transition occurs from state (i, j) to state (i', j') where i' is not the highest-priority candidate for node i. This can be calculated using the link delivery probability between nodes i and i', say $link_{prob}(i, i')$, given that none of the higher-priority candidates have already received the packet. The probability of this transition in the state machine can be calculated using Equation 4.3.

$$p^{i,j}_{c_x,0} = link_{prob}(i, c_x) \times \prod_{t=1}^{x-1}(1 - link_{prob}(i, c_t)) \tag{4.3}$$

- **Reaching a state corresponding to a retransmission or the *Fail* state:** As described earlier, if no candidates receive a packet when it is transmitted, the sending node tends to perform a retransmission. This happens for a maximum of K retransmissions. After that, if no candidate receives the packet, it is permanently discarded from the network. Equation 4.4 is designed to address the probability of retransmission or failure of a packet.

$$p^{i,j}_{i',j'} = 1 - \sum_{t=1}^{C} p^{i,j}_{c_t,0} \tag{4.4}$$

- **Reaching absorbing states:** Finally, in the process of different transitions, it would be possible for the system to reach one of the absorbing states. This is simulated as a situation in which the packet is discarded after K retransmissions, reaches the final destination successfully, or is grabbed by one of the malicious nodes in the network. All of the demonstrated cases are shown in the state machine using $Fail$, $Success$ or $(ID, 0)$ where ID is the identifier of a malicious node. In this scenario, the system will stay in the mentioned state, and no other transition takes place between states. This results in creation of an identity matrix, represented as I in the transition probability matrix P. Matrix I is created by setting $p^{i,0}_{i,0}$ to 1 where $(i, 0)$ shows an absorbing state.

4.2.4 Packet Drop Ratio Calculation

After creating states in DTMC and calculating all probability values in transition probability matrix P, it will be possible to obtain different parameters in the network. [61]

shows how to calculate different parameters such as packet delivery ratio, expected number of transmissions, and hop-count. In this chapter, calculation of another important parameter is introduced, known as *packet drop ratio* which is defined as the ratio of packets being received by uncooperative nodes and maliciously dropped. Basically, packet drop ratio can be obtained by calculating the probability of reaching each of the absorbing states that correspond to malicious nodes from an initial state, and combining them. Clearly, the initial state in OR protocols will be related to the source node, which generates data packets. Equation 4.5 shows the initial state of an OR-based DTMC. The probability of reaching any state from the initial state V after multiple (h) number of transitions between states can then be calculated using P^h.

$$V = \begin{bmatrix} 1 & 0 & ... & 0 \end{bmatrix} \tag{4.5}$$

If we assume existence of only one node as a source node in the network, say a node with $ID = 0$, it would be necessary to calculate the probability of reaching each absorbing state related to malicious nodes. For this purpose, considering the initial state of the system in Equation 4.5, calculating $V \times P^h$ will result in the first row of the P^h. This can be used to obtain the probability of reaching any absorbing state from the source node. More specifically, the element $(0, BH_1)$ in matrix $V \times P^h$ will represent the probability for malicious node BH_1 to receive and drop the packet. Finally, Equation 4.6 is used to calculate the overall ratio of packets being dropped by all of the malicious nodes. Similarly, as demonstrated in [61], it becomes possible to determine the probability of reaching $Success$ or $Fail$ states. Such values account for the probability of reaching the destination, or failing a packet, respectively.

$$Drop\ Ratio = \sum_{i=1}^{M} Drop_{BH_i} \tag{4.6}$$

Alternatively, as shown in [61], required parameters can be obtained using the fundamental matrix F of the Markov process. Equation 4.7 shows how to calculate F. In the end, by calculating $F \times R$, it would be possible to calculate the probability of reaching any absorbing state from any transient state.

$$F = (I - Q)^{-1} \tag{4.7}$$

4.3 Analysis

The proposed DTMC model, which takes into account the behavior of malicious nodes, must be evaluated. The model is independent of any OR or algorithm; however, four well-known OR protocols, EXOR [12], POR [127], DPOR [60], and MTS [88] have been considered as case studies. A brief explanation on the behavior of each algorithm is presented in the following subsections. All of the mentioned algorithms are then evaluated under the effect of different network parameters, using both the proposed analytical model as well as the simulation. The proposed model has been implemented using Java programming language while all simulations have been performed through Network Simulator 2 (NS 2.35) [68]. Having done so, it would be possible to perform the same set of experiments using the analytical model and simulation and compare results accordingly.

4.3.1 OR Case Studies

As explained earlier, four famous OR protocols have been selected for performance evaluation. Out of these four algorithms, EXOR and MTS select candidates using link delivery probability between nodes, where MTS is proven to select the optimum candidate set in terms of expected number of transmissions (ETX). The other two algorithms, POR and DPOR, take geographical location of nodes into account for candidate selection. This way, it would be possible to compare both categories of OR protocols.

EXOR Algorithm

EXOR [12] uses the ETX metric for candidate selection. More specifically, ETX between nodes i and j is calculated considering the link delivery probability between such nodes following Equation 4.8. First, EXOR tries to establish the shortest path between each node and its destination, then finds the first neighbor of each node as a potential candidate in this shortest path. Afterwards, if the ETX value from such a candidate to the destination is smaller when compared to the current node, this candidate will be added to the candidate set for the current node. This process is repeated until the maximum number of candidates is selected. Algorithm 1 provides greater detail on the method of candidate selection used by EXOR following [12].

$$ETX_{i,j} = \frac{1}{link_{prob}(i,j)} \tag{4.8}$$

Algorithm 1 EXOR Protocol

```
1: procedure SELECT-CANDIDATES(node, dest, C)
2:     cost_node ← ETX_node,dest
3:     CS_node,dest ← ∅
4:     while (|CS_node,dest| < C) and exists(shortestPath(node, dest)) do
5:         path ← shortestPath(node, dest)
6:         cand ← getNeighgour(node, path)
7:         if equals(cand, dest) then
8:             add(CS_node,dest, dest)
9:             cost(dest) ← 0
10:        else
11:            cost(cand) ← ETX_cand,dest
12:            if cost(cand) < cost(node) then
13:                add(CS_node,dest, cand)
14:            end if
15:        end if
16:        removeEdge(node, cand);
17:    end while
18: end procedure
```

POR Algorithm

POR [127] is a simple OR protocol that selects the candidate set for each node by considering only the geographical location of nodes. To be more specific, each node in the network selects the neighbor, which results in gaining the highest amount of distance progress towards the destination, and adds this neighbor to its set of candidates. This process continues until the maximum number of candidates has been selected, or the number of neighbors is too small to meet demands. The basic concept behind POR algorithm is to decrease the number of required hops to send a packet by selecting the nearest neighbors for each particular node to the destination. Algorithm 2 provides the pseudocode of the POR protocol, as explained in [127].

Algorithm 2 POR Protocol

1: **procedure** SELECT-CANDIDATES(node, dest, C)
2: $CS_{node,dest} \leftarrow \emptyset$
3: **while** $(|CS_{node,dest}| < C)$ *and* $notEmpty(neighbors(node))$ **do**
4: $cand \leftarrow findBestNeighborByDistanceProgress(node)$
5: $add(CS_{node,dest}, cand)$
6: $removeEdge(node, cand)$
7: **end while**
8: $sortByDistanceProgress(CS_{node,dest}, node, dest)$
9: **end procedure**

DPOR Algorithm

Similar to POR, DPOR [60] benefits from the geographical location of nodes for its candidate selection. DPOR introduces a metric known as Expected Distance Progress (EDP), and attempts to establish a balance between the amount of achievable distance progress through each neighbor and the link delivery probability between them. In fact, DPOR suggests candidate selection not only on their ability to progress the packet towards the destination, but which also have an acceptable link quality. More details regarding the candidate selection algorithm for DPOR can be found in Algorithm 3, according to [60].

MTS Algorithm

MTS [88] is an OR algorithm that guarantees selection of the optimum set of candidates when it comes to the expected number of transmissions between source and destination. MTS uses EAX as a metric for candidate selection. [25] provides more information on how to calculate the EAX metric. Basically, EAX is calculated recursively, considering multiple possible paths for reaching the destination in an OR-based wireless network. MTS initiates the process of candidate selection from the destination node's neighbors, adds the destination node to the candidate set of all such neighbors and assigns the cost of each link using the EAX of each candidate to the destination. Afterwards, the algorithm iteratively finds the node with a minimum amount of EAX to the destination, for example bestNode, and adds bestNode and all of its candidates to the initial candidate set of all bestNode's neighbors. Finally, to determine the optimum candidate set, nodes

Algorithm 3 DPOR Protocol

```
1: procedure SELECT-CANDIDATES(node, dest, C)
2:     candSetEDP ← −1
3:     neighbors ← getNeighbors(node)
4:     eligibleNeighbors ← ∅
5:     for each neighbor in neighbors do
6:         if distance_{neighbor,dest} < distance_{node,dest} then
7:             add(eligibleNeighbors, neighbor)
8:         end if
9:     end for
10:    sortByDistance(eligibleNeighbors)
11:    while (|CS_{node,dest}| < C) and notEmpty(eligibleNeighbors) do
12:        cand ← findBestNeighborByEDP(node)
13:        thisSet ← (CS_{node,dest} ⋃ cand)
14:        thisSetEDP ← EDP(thisSet)
15:        if thisSetEDP > candSetEDP then
16:            add(CS_{node,dest}, cand)
17:            candSetEDP ← thisSetEDP
18:            removeEdge(node, cand)
19:        end if
20:    end while
21:    sortByDistanceProgress(CS_{node,dest}, node, dest)
22: end procedure
```

are sorted in an increasing order using EAX value, and an exhaustive search is conducted to obtain the candidate set that includes less than or equal to C candidates. It has been proved in [88] that this set contains an optimum combination of candidates with regards to the expected number of transmissions (ETX). More details on the candidate selection method for MTS protocol is presented in Algorithm 4, following [88].

4.3.2 Customized Black-hole attack

As explained in Chapter 2 Section 2.2.1, black-hole nodes try to decrease network performance by attracting and dropping as many data packets as possible. In this section, a

Algorithm 4 MTS Protocol

1: **procedure** SELECT-CANDIDATES(node, dest, C)
2: $cost_{node} \leftarrow 0$
3: $nodes \leftarrow The\ set\ of\ all\ nodes\ except\ dest$
4: **for each** node in nodes **do**
5: **if** isNeighbor(node, dest) **then**
6: $add(CS_{node,dest}, dest)$
7: $cost(node) \leftarrow \frac{1}{link_{prob}(node,dest)}$
8: **else**
9: $CS_{node,dest} \leftarrow \emptyset$
10: $cost(node) \leftarrow \infty$
11: **end if**
12: **end for**
13: **while** $notEmpty(nodes)$ **do**
14: $currentNode \leftarrow minCost(nodes)$
15: $removeNode(currentNode, nodes)$
16: $neighbors \leftarrow getNeighbors(currentNode)$
17: **for each** neighbor in neighbors **do**
18: $add(CS_{neighbor,dest}, currentNode)$
19: **for each** c in $CS_{currentNode,dest}$ **do**
20: $add(CS_{neighbor,dest}, c)$
21: **end for**
22: $cost(neighbor) \leftarrow EAX(CS_{neighbor,dest}, neighbor, dest)$
23: **end for**
24: **end while**
25: $nodes \leftarrow The\ set\ of\ all\ nodes\ except\ dest$
26: $sortByCost(nodes)$
27: **for each** node in nodes **do**
28: $CS_{node,dest} \leftarrow pickBestSetsByMaxSize(CS_{neighbor,dest}, C)$
29: $cost(node) \leftarrow EAX(CS_{node,dest}, node, dest)$
30: **end for**
31: **end procedure**

variation of the black-hole attack is introduced, after customization for OR protocols. In the proposed version, the malicious node tries not only to drop all of the received packets, but also prevents other candidates from progressing the packet. In fact, in conducted simulations, once the malicious node receives the packet, it informs all other candidates (as well as the previous-hop node) that it has already received and forwarded the packet. Following perfect coordination, other candidates and the previous-hop then assume that the packet has already been transmitted, and they abstain from transmitting or retransmitting packets. The malicious node, however, drops the packet and removes it from the network.

4.3.3 Evaluation Settings

Before presenting and exploring evaluation results, it is of significant importance to study network parameters and settings. For instance, in the modeling and simulation of wireless networks, it is important to select a realistic propagation model. In this thesis, we have selected the shadowing propagation model. This is due to the fact that it is possible to simulate the existence of noise in wireless channels via the shadowing propagation model, as explained in [68]. More details on related parameters for this model are presented in Chapter 3 Section 3.2.

Network Parameters

Table 4.2 shows a list of all parameters being used in both analytical and simulation studies. In order to thoroughly study the effects of various parameters, three different parameters have been chosen for simulation including the number of malicious nodes, node density, and the maximum number of candidates (C). When all parameters are set to their default value, then changed one at a time, four different important network evaluation metrics are calculated and reported. These parameters include drop ratio, packet delivery ratio, expected number of transmissions, and hop count. It should be mentioned that plotted figures represent only the results of analytical studies, although the figures are significantly similar to simulation results. At the end of each subsection, however, a table is presented which includes the mean and the standard deviation of difference for each calculated parameter between analytical and simulation results.

Table 4.2: Simulation Parameters

Parameter	Value
Propagation Model	Shadowing
MAC	802.11
Number of nodes (N)	40
Network field dimension	$500 \times 500\ m^2$
Number of malicious nodes	6
Maximum Number of candidates (C)	3
Maximum Number of retransmissions (K)	3
Data Payload Size	512 bytes
Transmission Rate	5 Packets/Second
Coordination Delay	15 Milliseconds
Simulation Time	1800 Seconds

4.3.4 Results

As mentioned in Section 4.3.3, the results of performance evaluation for three different parameters are reported in this chapter. For each parameter, evaluations for both analytical and simulation results have been conducted 100 times by randomly changing the network topology and reporting the average value of all executions, while considering a confidence interval of %95. Finally, all graphs consist of four curves for EXOR, POR, DPOR, and MTS protocols.

Effect of Malicious Nodes

This section is presented to investigate the effects of the number of malicious nodes on different parameters. For this purpose, the number of malicious nodes changes from 0 to 15, where all other parameters are set to their default values listed in Table 4.2. As will be observed through all figures, malicious nodes can have significant and devastating effects on different network parameters.

- **Drop Ratio**

  Figure 4.4 shows the packet drop ratio as a function of the number of malicious nodes. Drop ratio in analytical results has been calculated using Equation 4.6. For simulation

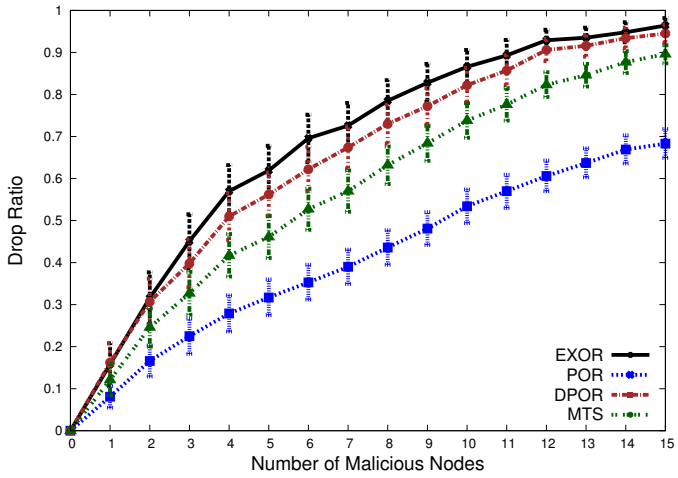

Figure 4.4: Drop Ratio

results, however, the drop ratio is calculated by dividing the overall number of dropped packets through malicious nodes by the total number of generated packets. Clearly, as the number of malicious nodes increases, the drop ratio rises as well. This is due to the fact that having more malicious nodes in the network, the probability of such nodes being selected as candidates in all of the protocols increases and, consequently, more malicious nodes will have the chance to attack the network by capturing data packets and dropping them accordingly. Amongst the four protocols, POR demonstrates the best performance when it comes to exposing packets to malicious nodes. The reason is that in POR, the focus of the algorithm is on minimizing the number of hops for every single packet. This indicates a reasonable decrease in the probability of packet receipt by malicious nodes. MTS, which focuses on optimizing the number of transmissions, ranks second. Here too, fewer overall transmissions indicates a smaller chance of capturing packets by malicious nodes, and consequently, a smaller drop ratio. Finally, DPOR outperforms EXOR through its consideration of nodes' geographical locations for candidate selection.

- **Delivery Ratio** The delivery ratio of packets is calculated via the method explained in [61]. Basically, the delivery ratio in the analytical results is calculated as the probability of receiving the success state from the initial state. In simulations, however, the delivery ratio is the percentage of packets being received to the destination node, divided by the total number of generated packets. As observed in Figure 4.5, increas-

ing the number of malicious nodes will result in a decrease of the the delivery ratio for all protocols. This is reasonable, as an increase in the number of malicious nodes will result in more packets being captured and dropped. This will clearly lead to a lower delivery ratio. When comparing different protocols, it is shown that MTS, which has an optimum candidate selection algorithm, possesses nearly the highest delivery ratio. As shown in previous paragraph, POR seems to be less affected by malicious nodes as compared to other protocols. The reason for its resiliency can be found in its candidate selection algorithm, which attempts to decrease the number of potential hops that can receive packets between the source and the destination. Similarly, DPOR outperforms EXOR in terms of packet delivery ratio.

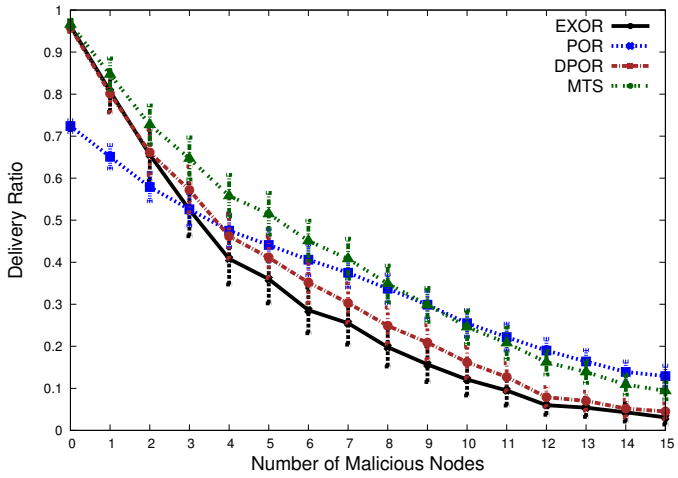

Figure 4.5: Packet Delivery Ratio

- **Expected Number of Transmissions** Figure 4.6 shows how a change in the number of malicious nodes can affect the expected number of transmissions (ETX). As observed, the ETX for MTS is reasonably the lowest value, when compared to other protocols. This behavior is predictable, because MTS has been proven to minimize ETX value. In contrast, EXOR has the worst ETX between all four protocols. POR, with its focus on minimizing hop count, outperforms DPOR. For all protocols, however, ETX value slightly decreases as the number of malicious nodes increases. The reason for this observation is that with the presence of a large number of malicious nodes in the network, a negligible proportion of packets will have an opportunity to arrive at their destination. Such packets demonstrate a rare situation, in that the route between

the source and the destination is most likely clear of malicious nodes and, as shown later in this section, fewer hops are required to route such packets to their destination. A smaller hop count will result in fewer overall transmissions/retransmissions and, as a result, a lower value for ETX.

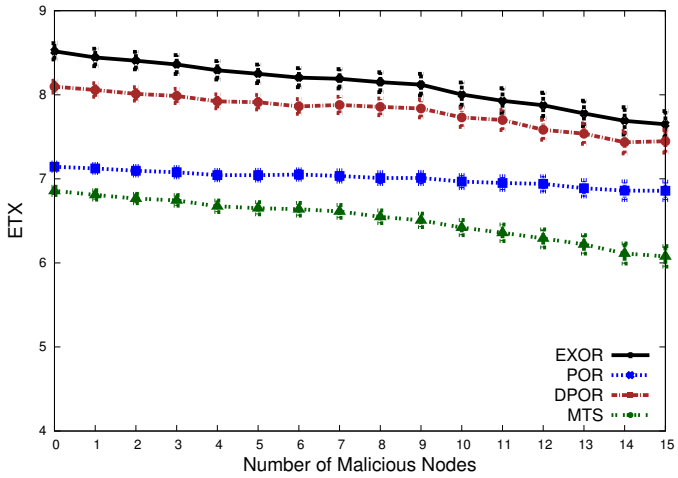

Figure 4.6: Expected Number of Transmissions

- **Hop count** Figure 4.7 shows the effect of changing the number of malicious nodes in the hop count. As observed, by increasing the number of malicious nodes, the hop count decreases slightly for all protocols. As previously explained, this is because an increase in the number of attackers corresponds with a greater probability that such nodes may receive and drop data packets. In this scenario, a smaller number of hops between the source and the destination indicates a higher probability that packets will reach their destination, because the probability of having a malicious node in the path also decreases. When different algorithms are compared, POR displays the best hop count by considering the closest node to the destination to be the best candidate, even though POR is incapable of delivering a large number of packets to their destination. MTS ranks second, with an optimal algorithm for candidate selection. DPOR is the third best algorithm, whereas EXOR ranks as the worst protocol in terms of the number of hops between source and destination.

- **Comparison of Results** This section is allocated for the comparison of conducted analytical results with simulation results. As observed in Table 4.3, two values have been reported for each evaluated parameter, which represent not only the average of the

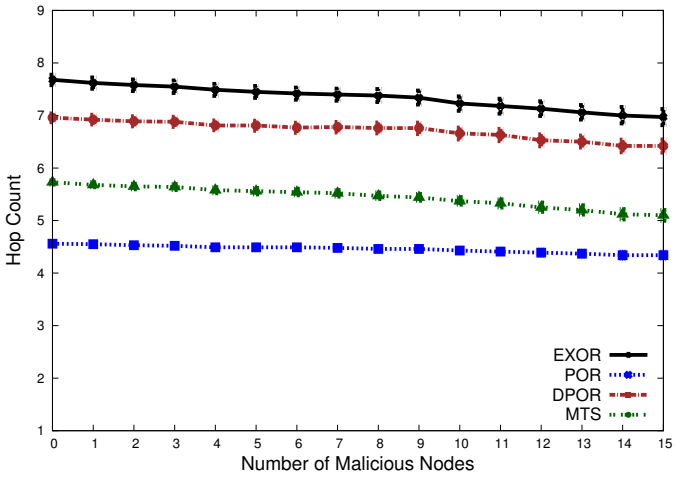

Figure 4.7: Hop Count

Table 4.3: Comparison of Results for Changing Malicious Nodes

Evaluated Parameter	Measurement	EXOR	POR	DPOR	MTS
Drop Ratio (%)	Average	0.0121	0.0175	0.0141	0.0286
	Deviation	0.0086	0.0097	0.0109	0.0125
Delivery Ratio (%)	Average	0.0759	0.0025	0.0710	0.0026
	Deviation	0.0360	0.0021	0.0266	0.0020
ETX	Average	0.1809	0.0233	0.2192	0.0289
	Deviation	0.0323	0.0092	0.0339	0.0140
Hop Count	Average	0.2643	0.0043	0.2512	0.0156
	Deviation	0.0467	0.0051	0.0398	0.0072

difference between all points in analytical and simulation results, but also their standard deviation. In fact, for drop ratio and delivery ratio, reported values demonstrate a percentage value, whereas absolute values are reported for ETX and hop count. As observed in Table 4.3, simulation results are, overall, very close to analytical results, which can be assumed as a verification for provided analytical results.

Effect of Node Density

This subsection studies the effects of changes in node density on different network parameters. For this evaluation, the dimensions of the network area will change from 300×300

to 1000 × 1000 square meters while the number of malicious nodes is set to 6 nodes.

- **Drop Ratio** Figure 4.8 demonstrates the effect of changes in network size on packet drop ratio. As observed, by enlarging the field size, the drop ratio increases for all protocols up to a certain level, and starts to decrease afterwards. This behavior is reasonable, since with a smaller network, say 300 × 300 m^2, the path between source and destination is shorter, and packets are not required to travel from many different hops to reach the destination. This decreases the probability that malicious nodes may receive data packets. In contrast, by enlarging the field size, more nodes will become involved in routing packets towards their destination. This offers greater opportunities for malicious nodes to capture more packets. However, when the network size is too large, say 1000 × 1000 m^2, the average distance between nodes will be reflective of that size; therefore, a great deal of packets will become lost in the network as a result of obstructions in the wireless channel. Therefore, although malicious nodes may still be selected as potential candidates by other nodes, fewer packets will successfully reach them, so they can discard less. A comparison of different protocols shows that POR is the most resistant protocol against malicious nodes, and EXOR ranks as the worst.

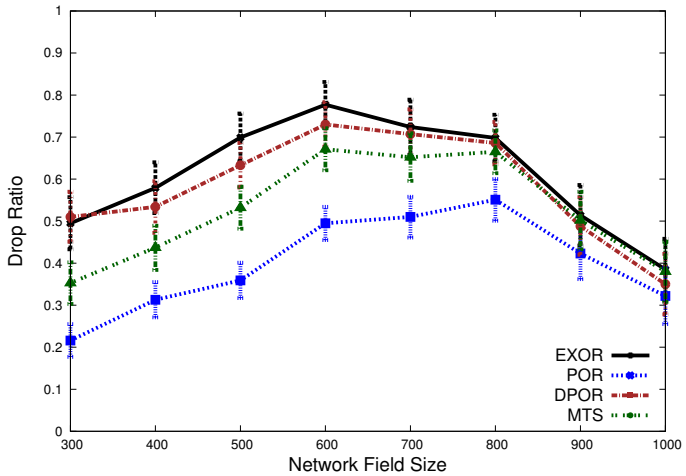

Figure 4.8: Drop Ratio

- **Delivery Ratio** Figure 4.9 shows packet delivery ratio as a function of changing the network field size. Reasonably, by increasing the network size, delivery ratio decreases. This occurs because, first of all, malicious nodes will be able to capture and drop some of the received data packets. Second, as the network size extends, the distance between

nodes increases and the probability of packet loss escalates for all different protocols. Therefore, fewer packets will have the opportunity to successfully reach the destination. A comparison of different protocols indicates that MTS acts as the best algorithm for packet delivery to the destination, whereas EXOR shows the worst delivery ratio. Interestingly, however, POR almost outperforms both EXOR and DPOR. The reason, as stated in the previous subsection is that POR will try to decrease the number of necessary hops between the source and the destination. This results in a smaller probability of receiving packets to malicious nodes compared to DPOR and EXOR and, as a result, a higher delivery ratio. DPOR still outperforms EXOR by incorporating both link delivery probability between nodes, and their geographical information for candidate selection.

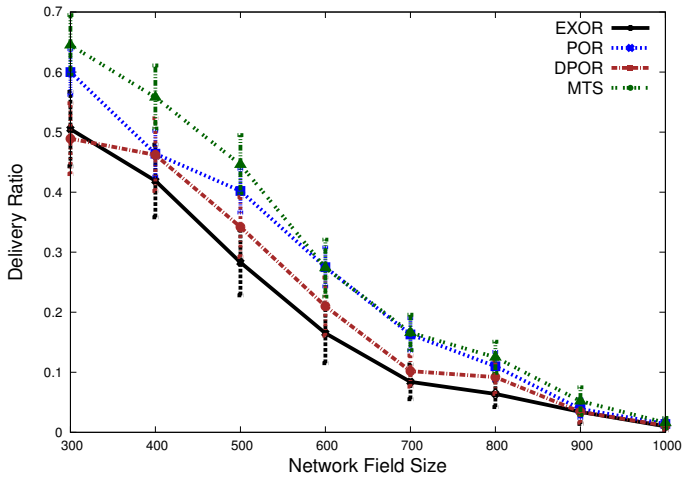

Figure 4.9: Packet Delivery Ratio

- **Expected Number of Transmissions** Figure 4.10 shows the expected number of transmissions for all different protocols as a function of network field size. enlarging the network size in all of the protocols results in an increase in distance between source and destination; as a result, packets will need to travel longer paths to reach their destination. Having longer paths means that packets must be transmitted or retransmitted more frequently in larger networks, when compared to smaller ones. In this scenario, MTS, as expected, performs the best due to its optimum candidate selection scheme, and POR ranks second. DPOR and EXOR rank third and fourth, respectively.

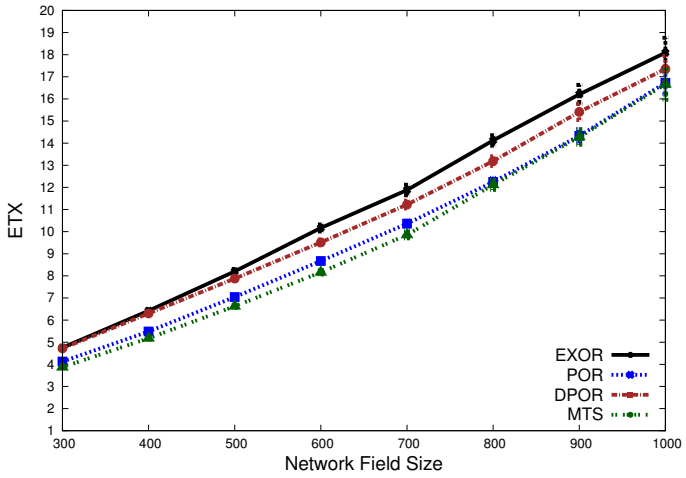

Figure 4.10: Expected Number of Transmissions

- **Hop Count** The hop count of packets sent between the source and the destination is shown in Figure 4.11. As explained earlier, a larger network area requires longer paths. Therefore, it is reasonable to have a higher hop count for larger networks. Taking the POR algorithm as an example, in a 300×300 m^2 field packets must travel for less than 3 nodes to reach their destination, whereas in a 600×600 m^2 field, the hop count is less than 6 and in a network with an area of 1000×1000 m^2, the value is less than 11. Comparing different protocols also represents a reasonable behavior in which POR has the lowest hop count, and EXOR has the highest. Similarly, MTS outperforms DPOR due to its candidate selection algorithm.

- **Comparison of Results** A comparison of simulation and analytical results following changes to node density is shown in Table 4.4. As observed, the average and standard deviation of changes show that simulation results are close to analytical ones. This conveys that changes to different network parameters follow the same trends for both sets of conducted results.

Effect of Candidates

This subsection investigates the effect of the number of candidates on different network parameters. In this scenario, the maximum number of candidates changes from one to six nodes, while other parameters are set to their default values, as shown in Table 4.2.

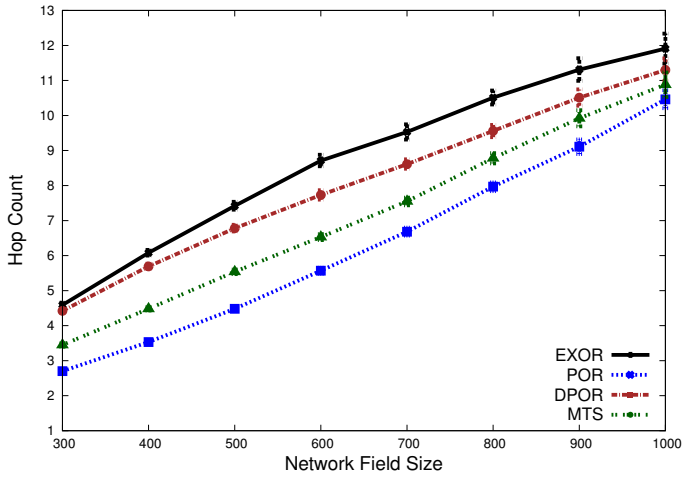

Figure 4.11: Hop Count

Table 4.4: Comparison of Results for Changing Node Density

Evaluated Parameter	Measurement	EXOR	POR	DPOR	MTS
Drop Ratio (%)	Average	0.0732	0.0686	0.0650	0.0808
	Deviation	0.0484	0.0602	0.0632	0.0615
Delivery Ratio (%)	Average	0.0640	0.0018	0.0582	0.0025
	Deviation	0.0504	0.0012	0.0445	0.0017
ETX	Average	0.3535	0.0215	0.3084	0.0245
	Deviation	0.2344	0.0146	0.1550	0.0154
Hop Count	Average	0.3812	0.0187	0.3137	0.0287
	Deviation	0.1872	0.0172	0.1147	0.0195

- **Drop Ratio** Figure 4.12 shows the effect of candidate changes on the packet drop ratio. The change in the drop ratio for EXOR and DPOR appears insignificant when the number of candidates is more than two. More specifically, although the probability of a packet reaching its destination increases with the presence of more nodes in a candidate set, the probability of including a malicious node in the candidate set also escalates. POR, however, shows an interesting behavior; an increase in the number of malicious nodes leads to an increase in packet drop ratio for this protocol. This is because a small number of candidates, say one node, results in losing a large number of packets due to packet loss and fading in the propagation model. In fact, increasing the number of nodes in the candidate set decreases the chance of packet loss, while at

the same time, the probability for more malicious candidates increases the. This will lead to an increase in drop ratio. MTS reasonably shows results identical to EXOR when there is only one node in the candidate set. However, with more candidates, MTS' focus on best candidate selection shows a slight reduction until the number of candidates is 3, and the trend becomes almost constant thereafter. Overall, it can be concluded that EXOR has the highest drop ratio, whereas POR has the lowest value. Similarly, in MTS fewer packets can be captured by malicious nodes.

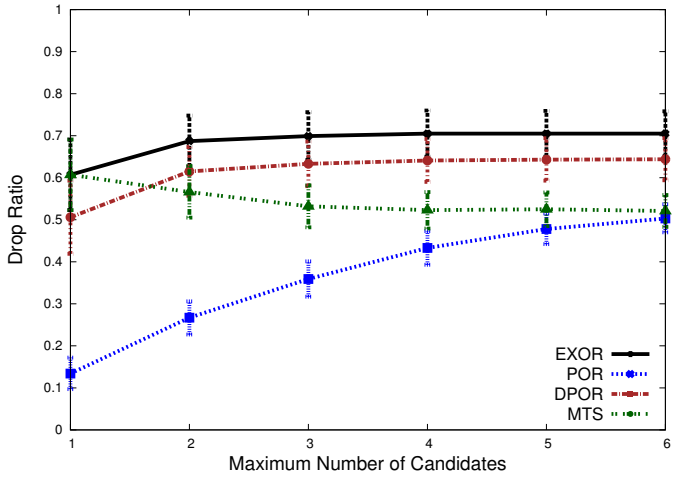

Figure 4.12: Drop Ratio

- **Delivery Ratio** Packet delivery ratio is shown in Figure 4.13. In POR, as discussed earlier, by increasing the number of candidates the probability of packet loss decreases, while the algorithm still attempts to reduce the hop count by selecting nodes closest to the destination. Therefore, the reliability of sending packets to their destination increases, resulting in an increase in packet delivery ratio. MTS also shows a considerable rise in delivery ratio until the number of candidates is 3 nodes. After that, the trend of packet delivery ratio becomes almost constant, similar to EXOR and DPOR algorithms. Finally, effects of the candidate selection algorithm can be clearly compared between all protocols. Overall, MTS has been proven to function as the best algorithm. POR shows a poor delivery ratio when the maximum number of candidates is less than 3 nodes, and its performance increases with more nodes in the candidate set. Finally, DPOR has a higher delivery ratio compared to EXOR by incorporating geographical information with link delivery probability between nodes.

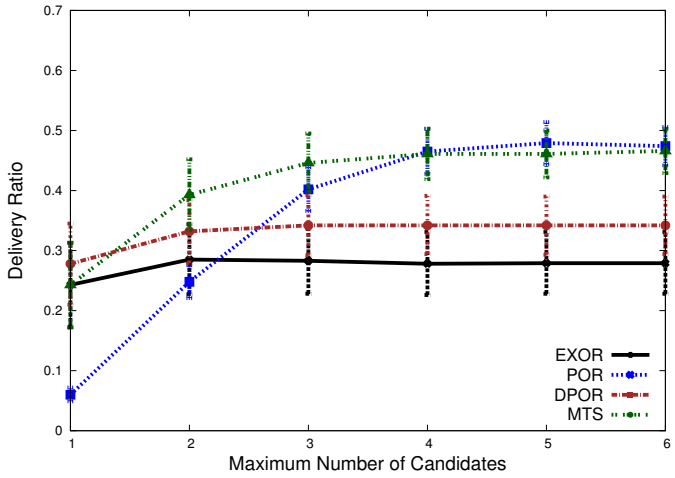

Figure 4.13: Packet Delivery Ratio

- **Expected Number of Transmissions** Figure 4.14 shows the expected number of transmissions when the number of candidates is variable. For all protocols, the expected number of transmissions decreases as the number of candidates increases. This is reasonable because, as explained earlier, having more candidates in the candidate set means that the probability of packet loss will decrease. Thus, packets will most likely either be captured and dropped by malicious nodes, or successfully sent to the next hop. In either case, the probability of retransmission decreases, and packets may experience fewer transmissions. Here too, MTS shows the best ETX when there is more than one node in the candidate set whereas EXOR shows the worst value. Finally, DPOR demonstrates better performance than EXOR.

- **Hop count** The hop count of packets is represented in Figure 4.15. As observed in POR, an increase in the number of candidates results in a slight corresponding increase in hop count. This is due to the fact that a smaller number of candidates leads to greater packet loss in POR; if some packets reach the destination, they have traveled from a very short path. As the number of candidates increases, however, more packets will reach the destination, but packets may need to travel from more hops. Nevertheless, POR still has the lowest hop count compared to other protocols, and EXOR shows the worst hop count. MTS reasonably ranks second, while DPOR operates better than EXOR but worse than MTS.

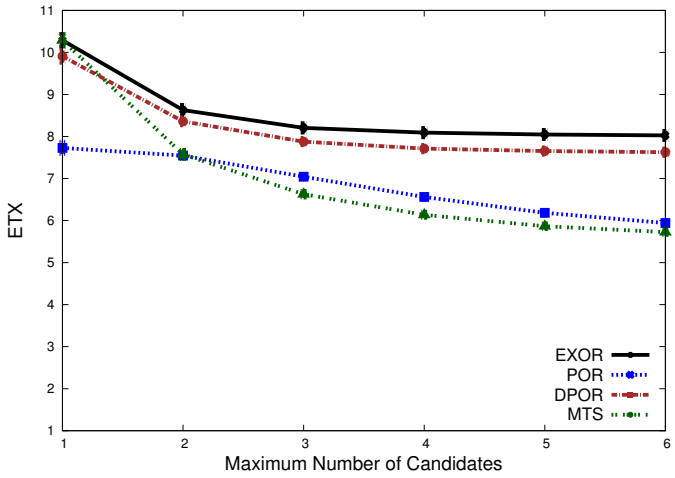

Figure 4.14: Expected Number of Transmissions

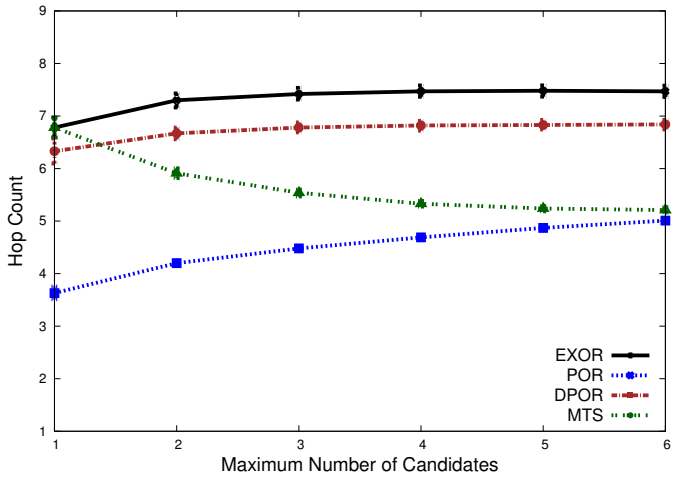

Figure 4.15: Hop Count

- **Comparison of Results:** Finally, Table 4.5 shows the average and standard deviation of gaps between simulations as well as analytical results. The values on the table demonstrate that simulation results are very close to analytical results, which verifies the correctness of the proposed model and reported results.

4.4 Summary

In this chapter, the effects of malicious nodes were studied on opportunistic routing protocols in wireless networks. More specifically, assuming that nodes in the candidate

Table 4.5: Comparison of Results for Changing Candidates

Evaluated Parameter	Measurement	EXOR	POR	DPOR	MTS
Drop Ratio (%)	Average	0.0150	0.0180	0.0181	0.0233
	Deviation	0.0060	0.0133	0.0076	0.0147
Delivery Ratio (%)	Average	0.1011	0.0040	0.0840	0.0051
	Deviation	0.0555	0.0036	0.0416	0.0078
ETX	Average	0.2132	0.0552	0.1819	0.0668
	Deviation	0.1314	0.0995	0.0958	0.1072
Hop Count	Average	0.2833	0.0266	0.2350	0.0516
	Deviation	0.1676	0.0557	0.1275	0.0884

set follow perfect coordination, a new analytical model was designed and implemented using Discrete-Time Markov Chain (DTMC) to demonstrate the existence of malicious nodes. Additionally, in order to measure the effect of malicious nodes on the network, a new method of calculating drop ratio was introduced. As an example of a malicious behavior, an implementation of a black-hole attack was introduced, after customization for opportunistic routing protocols. Finally, a comprehensive set of performance evaluation scenarios was designed and conducted, using both simulation and analytical studies on four well-known opportunistic routing protocols known as EXOR, POR, DPOR, and MTS. To summarize, evaluation results demonstrate that malicious nodes can significantly decrease the performance of wireless networks by preventing packets from reaching their destination. Finally, by comparing simulation results and analytical results, we conclude that the proposed model is capable of demonstrating the effects of malicious nodes on opportunistic routing protocols. For future works, we will extend the proposed analytical model to include a defensive mechanism against malicious nodes, using a variation of trust and reputation systems.

Chapter 5

A Packet Salvaging Model to Improve the Security of Opportunistic Routing Protocols

In Chapter 3, a novel trust management scheme was designed, developed, and evaluated specifically for OR protocols. Chapter 4, however, presented a novel analytical scheme and mathematically modeled the effects of adversary nodes on an OR-based wireless network. In this chapter, the analytical model introduced in Chapter 4 is extended to demonstrate the efficiency of OR protocols in salvaging packets that are maliciously dropped by attacker nodes. Greater details regarding the proposed model can be found in the following subsection. Section 5.2 provides a performance evaluation on the proposed packet salvaging model, and Section 5.3 summarizes this chapter.

5.1 A DTMC-based Packet Salvaging Model for OR

As discussed in Chapter 1 Section 1.1, the main advantage of OR protocols is the redundancy of nodes in the candidate set. In hostile environments, however, including a higher number of nodes in the candidate set will result in a higher risk of selecting malicious nodes in the candidate set. In this situation, as shown in Chapter4, malicious nodes can significantly decrease the performance of OR protocols. To defend against attackers, we propose a packet salvaging mechanism through which nodes in the candidate set can follow an appropriate candidate coordination mechanism, and significantly nullify the

effects of attacker nodes by saving dropped packets.

5.1.1 Preliminaries

In this chapter, similar to Chapter 4, the Discrete-Time Markov Chain (DTMC) is used for modeling an OR-based wireless mesh network. The proposed model operates under the assumption that nodes in the candidate set follow the *perfect coordination* algorithm [61]. As explained in [25], perfect coordination allows the actual relay node in the candidate set to send a notification to all other candidates, as well as the previous-hop node. This will prevent duplicate transmissions of data packets. Regarding the modeling tool, DTMC is a state machine, in which the states are connected to each other via edges weighted by probability values. Such values demonstrate the probability of transitioning from one state to the other. It has been shown in [61] that DTMC is an appropriate tool for modeling OR-based wireless mesh networks, for many reasons. First, DTMC is a memoryless system meaning, that the transition from the current state is independent of previous states. This situation is similar to OR protocols, in which a packet must be forwarded to the destination once it reaches an intermediate node, regardless of the previous hops. Second, transitioning between states of the DTMC is determined using a probability value. To illustrate, we offer an analogy of node packet transmission in OR protocols. According to the priority of nodes in the candidate set, as will be discussed in greater details in Section 5.1.2, each candidate can be assigned a probability to act as the actual relay node. Third, the DTMC is composed of *transient* and *absorbing* states. If the state machine transitions to one of the transient states, it will further shift to other states. However, transitioning to absorbing states will result in the state machine remaining in such states, and no further transitions will take place. Transient states can be mapped to intermediate nodes in OR protocols, whereas absorbing states simulate either the arrival of a packet at its final destination, or the failure of a packet to arrive due to the unreliability of links [61].

Furthermore, as discussed in Chapter 4, it is possible to model the existence of malicious or uncooperative nodes in OR protocols. For example, Fig. 5.1 shows a simple scenario of a wireless network modeled with DTMC, where nodes form a linear topology. In this example, each state in the DTMC is shown using a tuple in the form of $(ID, ReTx)$. ID represents the identifier of a node and $ReTx$ stands for the number of retransmissions that occurred in that node. Furthermore, state $(4, 0)$ is an absorbing

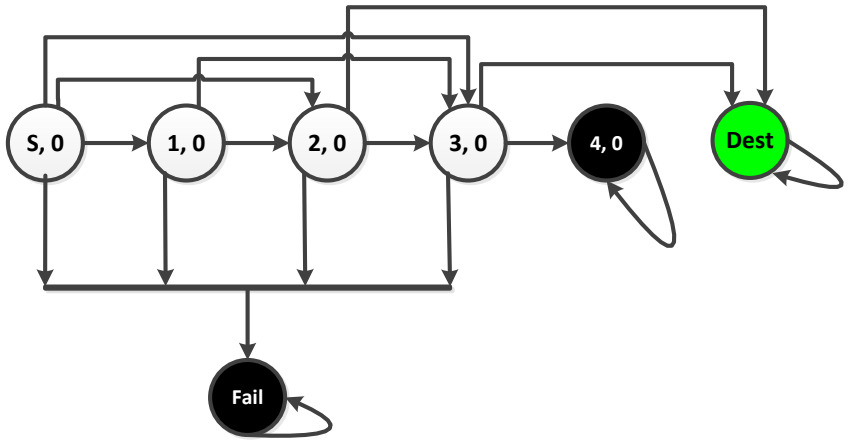

Figure 5.1: An example of modeling OR protocols using DTMC in presence of malicious nodes

state regarding malicious nodes. This means a node with ID=4 does not collaborate in the packet forwarding process. For simplicity, only one transmission of packets has been depicted in Fig. 5.1, and retransmissions have not been shown. Greater details regarding this model and its performance evaluation can be found Chapter 4. In short, the model proposed in Chapter 4 represented that malicious nodes can significantly decrease the performance of OR protocols. Therefore, it would be very interesting to study how OR protocols can be applied in hostile environments, and how such protocols can cancel the effects of malicious nodes. This has motivated us to propose a DTMC-based model, with the purpose of salvaging data packets dropped by malicious nodes.

5.1.2 Proposed Packet Salvaging Model

The proposed model functions under the assumption that nodes follow the perfect coordination mechanism. Upon transmission of a data packet, this assumption conveys that the relay node sends a notification to other candidates to prevent repeated forwarding. However, this also creates a possibility for malicious nodes to discard or corrupt a data packet, and send false acknowledgments. The idea of the proposed analytical model in this chapter is to study how the set of candidates can save packets that are maliciously dropped or manipulated by attacker nodes. More precisely, it is assumed that nodes in

the candidate set will follow perfect coordination, while promiscuously listening to the wireless channel. Therefore, candidates will be able to monitor one anothers' behavior with greater efficiency, and discover any misbehavior accordingly. In fact, in OR protocols, candidate nodes will be given the chance to forward packets according to their priority in the candidate set. If a node prevents or corrupts the forwarding of a packet, all of the lower-priority candidates will notice this misbehavior, and will have a chance to salvage this packet by forwarding it towards the destination. In summary, if benevolent candidates ratify that a higher-priority candidate is acting maliciously, they can form a chain of *backup candidates*, and try to salvage the corrupted data packet. In this chapter, this packet salvaging mechanism and the consequences of utilizing it in a wireless mesh network has been modeled and studied using DTMC.

The State Set in the Proposed DTMC

Considering that DTMC is used as a modeling tool in this chapter, the first step is to define the set of states creating the Markov chain. In the proposed model, three different types of states are defined and created as follows.

- **Regular transient states:** Regular transient states are related to benign intermediate nodes, which receive data packets and collaboratively forward them towards their destination. Similar to the model demonstrated in Chapter 4, Each state in this category is recognized using $(ID, ReTx)$ tuple, which demonstrates the current node's identifier, and the number of retransmissions taken place in this node on the same data packet.

- **Malicious transient states:** Malicious transient states are related to malicious nodes. Each state of this class is represented using a different tuple in the form of $(ID, Selector)$. This tuple demonstrates the identifier of the malicious node, along with the identifier of the node that has selected this malicious node as a candidate.

- **Absorbing states:** The proposed packet salvaging model consists of three different absorbing states known as *Fail*, *Drop*, and *Dest*. The *Fail* state shows a packet that has been transmitted or retransmitted for the number of allowed repetitions. Consequently, this packet is not going to be further considered for routing, and should be eliminated from the network. This is modeled in the proposed DTMC by transitioning from a transient state to *Fail*. *Drop* represents a packet drop by

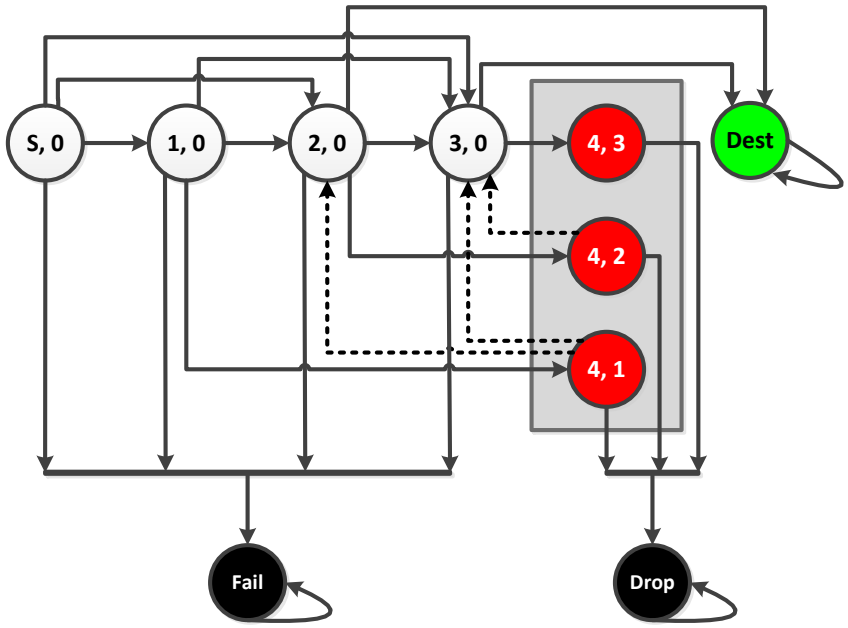

Figure 5.2: An example of the proposed packet salvaging model

one of the malicious nodes. More specifically, if a packet is dropped or corrupted by one of the malicious nodes and no candidate is able to salvage this packet, the DTMC will shift to the *Drop* state. Finally, *Dest* is an absorbing state representing the successful arrival of a packet to its final destination.

Regarding the creation of the proposed DTMC-based analytical model, two different network scenarios are considered and examined. The first scenario demonstrates a network containing a single malicious node, whereas the second one takes into account the existence of multiple malicious nodes in the network. Details of such scenarios and their representation is found following in this subsection.

Scenario 1: Modeling a Single Malicious Candidate

Once the set of states in the Markov chain are known, the next step is to recognize the probability of transitioning between such states. For this purpose, although the proposed model is independent of any network topology or candidate selection algorithm, a simple scenario is created and explained, as shown in Fig. 5.2. In this scenario, 6 nodes exist in

the network located on a linear topology, wherein each node selects a maximum of three candidates, and candidates closer to the destination are assigned a higher priority in the candidate set. Furthermore, state $(S, 0)$ shows the source node, state *Dest* demonstrates the destination, and node with $ID = 4$ is the only malicious node in the network. This malicious node receives data packets from its candidate selectors, for example, nodes with $ID = [1, 2, 3]$, and either corrupts or eliminates all of them from the network. On the other hand, as mentioned earlier in this section, malicious transient states are created according to the number of nodes that select malicious nodes as their candidates. Therefore, three different malicious transient states are created in the proposed DTMC shown as $(4, 1)$, $(4, 2)$, and $(4, 3)$. However, considering node 1 as a candidate selector, it is worth noting that there are two candidates, say nodes 2 and 3, other than the malicious node having been selected as candidates for node 1. In the depicted scenario, such nodes have a lower priority than node 4 in the candidate set. Therefore, they can actively monitor the behavior of node 4, and react once an attack is detected. More precisely, nodes 2 and 3 can act as backup candidates, and can salvage packets dropped or maliciously manipulated by node 4. This process is shown in Fig. 5.2 using dashed arrows, which represent a transition from malicious transient to regular transient states. Such arrows show that lower-priority candidates can rescue packets dropped by higher-priority malicious candidates.

Scenario 2: Modeling Multiple Malicious Candidates

In OR-based wireless networks that are applied in hostile environments, it is possible for multiple malicious nodes to be selected in the candidate set of a specific relay node. This scenario has been depicted in Fig. 5.3, where two malicious nodes, say nodes with $ID = [3, 4]$, are selected in the candidate set of node 1. In this situation, node 3 will be selected as a candidate for nodes $[S, 1, 2]$, and node 4 will be selected as a candidate for nodes $[1, 2, 3]$. Considering node 1 as a relay node, this node selects nodes with $ID = [2, 3, 4]$ as its candidates. Therefore, if a packet transmitted by node 1 is captured by node 4, it tends to eliminate the packet from the network. On the other hand, according to the proposed packet salvaging model, lower-priority candidates will have an opportunity to salvage this packet. However, node 3 is also a malicious node that prevents collaborating in packet forwarding, but also does not contribute to the packet salvaging process. This means that it is not required to create state $(4, 3)$ in the proposed

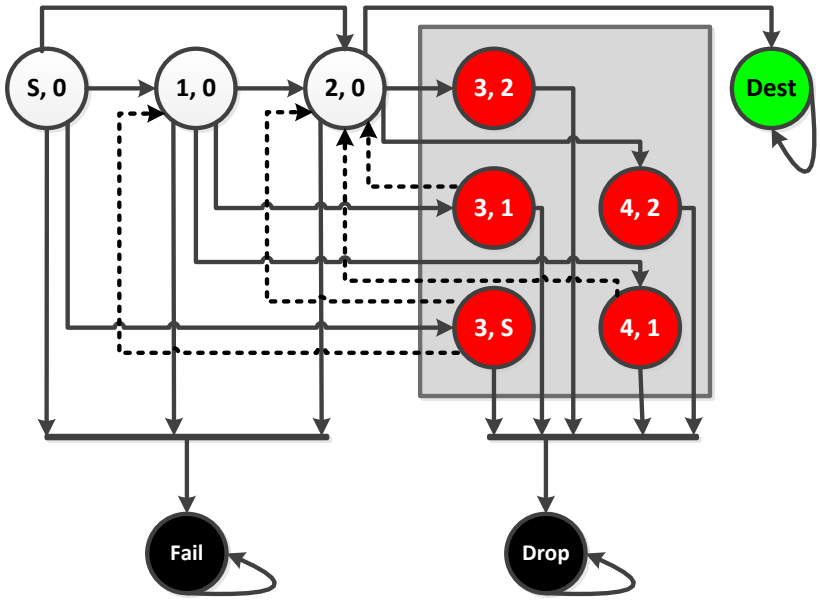

Figure 5.3: An example of the proposed packet salvaging model when multiple malicious nodes are selected in the candidate set

DTMC. Consequently, the only possible savior for packets sent by node 1 will be node 2. In Fig. 5.3, this is shown using a dashed arrow from state $(4, 1)$ to state $(2, 0)$. Similarly, the proportion of packets sent by node 2 that do not reach the destination will be completely lost from the network. This is due to the fact that nodes with $ID = [3, 4]$ are both malicious nodes, which do not transmit or salvage packets for node 2.

In summary, according to the priority of nodes in the candidate set, backup candidates can start the process of packet salvaging until the packet is saved. If no backup candidates exist or such nodes are unable to save the packet, it will be removed from the network unless a retransmission takes place by the previous-hop node. The following subsections include greater details of the proposed model, and demonstrate how the probability of transitioning between states in the Markov chain is calculated.

The General Architecture of the DTMC State Transitions

Fig. 5.4 shows a different representation of transitions that can take place for a single packet in the proposed DTMC until it reaches one of the absorbing states. More precisely, once a packet is transmitted by the source node, according to the priority of nodes in the source's candidate set, and also based upon the probability of link delivery between the source and its candidates, one of the following cases might occur.

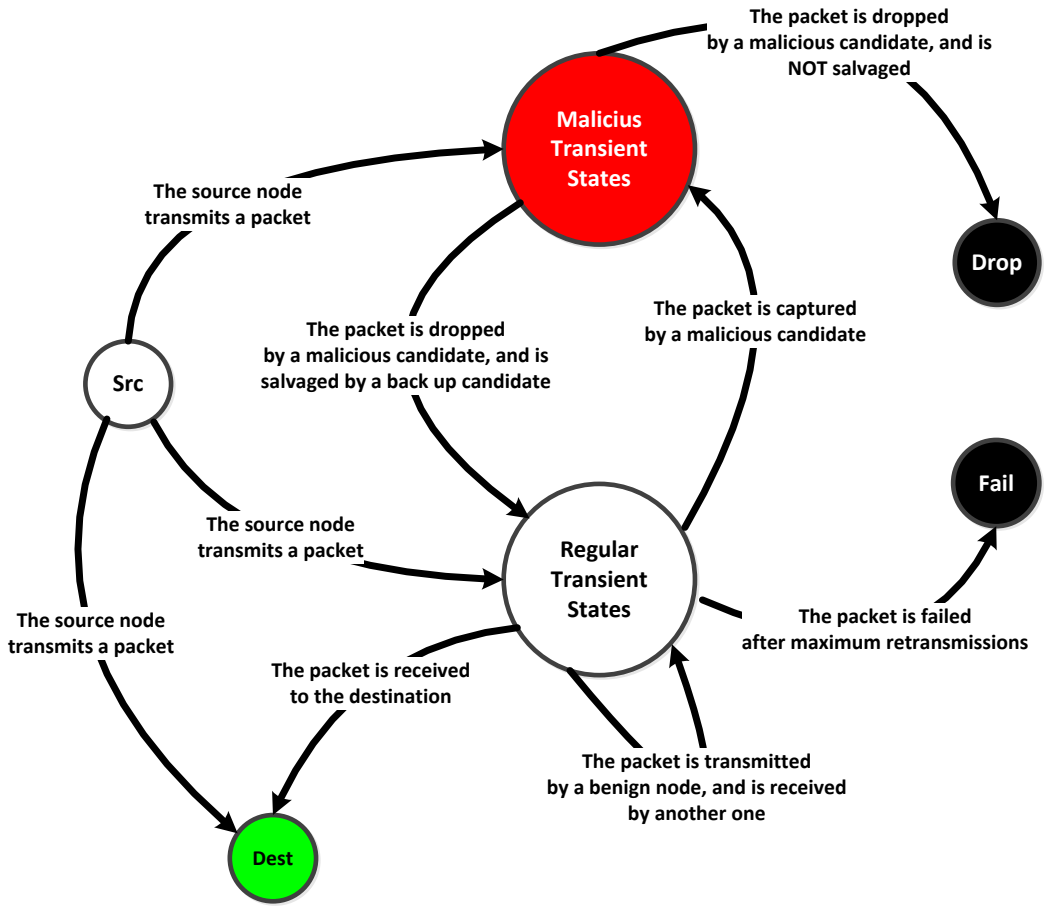

Figure 5.4: The general architecture of transitions between states of the proposed DTMC for a single packet

- **Reaching a malicious candidate:** If a malicious candidate receives the packet, it simply drops or manipulates the packet, and notifies other candidates and the source node that it has forwarded the packet. This has been shown in Fig. 5.4 as a transition

between the source node and a node corresponding to malicious transient states. However, lower-priority candidates might recognize this misbehavior, and tend to salvage the packet by forwarding it one hop closer to the destination. Therefore, a transition can take place between malicious transient and regular transient states. On the other hand, if no back-up candidate exists in the set of candidates, or back-up candidates do not realize this misbehavior, the dropped packet will be permanently eliminated from the network. This fact is shown using a transition between malicious transient states and the *Drop* state.

- **Reaching a benign candidate:** If a benevolent candidate receives the generated packet, according to its priority in the candidate set, it forwards the packet to its candidate set and notifies other nodes of this transmission. In this case, the packet might reach another benevolent candidate, a malicious candidate, the final destination, or it might be failed after performing the maximum number of retransmissions. Such occurrences have been visualized using transitions from regular transient to regular transient, to malicious transient, to the *Dest*, or to the *Fail* states in Fig. 5.4 respectively.

- **Reaching the final destination:** In some situations, the distance between the source and the destination nodes might be insignificant. Therefore, according to the candidate selection algorithm, the destination node might be selected as a candidate for the source node. In this situation, if the destination receives the packet generated by the source, no more interaction between nodes needs to occur; this has been shown using a transition from *Source* state to the *Dest* in Fig. 5.4.

Markov Transition Probability Matrix

After creating the states of the Markov chain and recognizing different transitions between states, the transition probability matrix of Markov process should be created. Transition probability matrix $\mathbb{P}$ is a stochastic matrix containing the probability of transitioning between different states in the state machine. Generally, $\mathbb{P}$ is composed of four sub-matrices as shown in Equation 5.1, and explained in Chapter 4, where $\mathbb{Q}$ includes the probability of transitioning between transient states; $\mathbb{R}$ contains the probability of transitioning between transient and absorbing states; $\varnothing$, that is filled with zero values, encompasses the probability of shifting from absorbing to transient state, and $\mathbb{I}$, which

$$\mathbb{P} = \begin{bmatrix} \mathbb{Q} = \begin{bmatrix} \mathbb{TT} & \mathbb{TM} \\ \\ \mathbb{MT} & \mathbb{MM} \end{bmatrix} & \mathbb{R} = \begin{bmatrix} \mathbb{TA} \\ \\ \mathbb{MA} \end{bmatrix} \\ \\ \varnothing = \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ 0 & \dots & 0 \end{bmatrix} & \mathbb{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

Figure 5.5: The transition probability matrix schema in the proposed model

is an identity matrix, contains the probability of transitioning between absorbing states.

$$\mathbb{P} = \begin{bmatrix} \mathbb{Q} & \mathbb{R} \\ \varnothing & \mathbb{I} \end{bmatrix} \tag{5.1}$$

However, regarding our proposed model in this chapter, the transition probability matrix $\mathbb{P}_{\mathbb{N}_\mathbb{P} \times \mathbb{N}_\mathbb{P}}$ needs to be investigated in greater detail. As shown in Fig. 5.5, matrices $\mathbb{Q}_{\mathbb{N}_\mathbb{Q} \times \mathbb{N}_\mathbb{Q}}$ and $\mathbb{R}_{N_Q \times 3}$ have specific representations in the proposed Markov chain. More precisely, matrix $\mathbb{Q}$ is composed of four sub-matrices known as $\mathbb{TT}_{N_T \times N_T}$, $\mathbb{TM}_{N_T \times N_M}$, $\mathbb{MT}_{N_M \times N_T}$, and $\mathbb{MM}_{N_M \times N_M}$. $\mathbb{TT}$ accommodates values that indicate the transition probability between regular transient states, while $\mathbb{TM}$ shows the probability of shifting from regular transient to malicious transient states. Sub-matrix $\mathbb{MT}$ includes the probability of changing the state from malicious transient to regular transient states, and finally, $\mathbb{MM}$ that is filled with zero values, contains the probability of transitioning between malicious transient states.

Similarly, matrix $\mathbb{R}$ is composed of two sub-matrices represented as $\mathbb{TA}$ and $\mathbb{MA}$. $\mathbb{TA}$ contains the probability of shifting from regular transient to absorbing states, and $\mathbb{MA}$ shows the probability of changing the state from malicious transient to absorbing states. For example, in Fig. 5.2, the probability of transitioning between states $(1, 0)$ and $(2, 0)$ will fit in matrix $\mathbb{TT}$; however, the transition probability between states $(4, 1)$ and $(3, 0)$ will be stored in matrix $\mathbb{MT}$. Table 5.1 contains examples of the distribution of probability values in different sub-matrices, based upon Fig. 5.2. Regarding dimensions of different sub-matrices, N_T demonstrates the number of regular transient states; N_M accounts for the number of malicious transient states; $N_Q = N_T + N_M$ represents all

Table 5.1: Examples of transition probability matrix values

From State	To State	Matrix
$(1,0)$	$(2,0)$	$\mathbb{TT}$
$(1,0)$	$(4,1)$	$\mathbb{TM}$
$(4,1)$	$(3,0)$	$\mathbb{MT}$
$(3,0)$	Dest	$\mathbb{TA}$
$(1,0)$	Fail	$\mathbb{TA}$
$(4,1)$	Drop	$\mathbb{MA}$

transient states, and considering only three absorbing states, $N_P = N_Q + 3$.

In DTMC, upon creating the state machine and recognizing the set of transitions between states, it should be possible to calculate all transition probabilities. More precisely, all elements of $\mathbb{P}$ should be recognized and calculated. Figs. 5.6 and 5.7 show the open form of matrices $\mathbb{Q}$ and $\mathbb{R}$ where $p^{(i,j)}_{(i',j')}$ demonstrates the probability of transitioning from state (i,j) to (i',j'). In order to calculate elements of $\mathbb{P}$, different rules are defined according to various sets of transitions, as follows.

- **Transitioning from a regular transient state to a state representing the highest-priority candidate:** This rule indicates a transition from a regular transient state to another state in which the node is the highest-priority candidate, say c_1, in the candidate set. For example, in Fig. 5.2, the element $p^{(1,0)}_{(4,1)}$ is determined using this rule since node with $ID = 4$ is the highest-priority candidate for node with $ID = 1$. Generally, this value is calculated according to the link delivery probability between a node and its highest-priority candidate as shown in Equation 5.2 [61].

$$p^{i,j}_{c_1,0} = prob^{link}(i, c_1) \tag{5.2}$$

- **Transitioning from a regular transient state to states demonstrating lower-priority candidates:** Similar to the model introduced in Chapter 4, this rule is used to define the probability of passing from a regular transient to a state representing a node other than the highest-priority candidate. For instance, in Fig. 5.2, $p^{(2,0)}_{(4,2)}$ is calculated using this rule. Basically, according to perfect coordination, lower-priority nodes in the candidate set operate on condition that no higher-priority candidate

$$\mathbb{Q} = \begin{bmatrix} \mathbb{TT} = \begin{bmatrix} p_{(0,0)}^{(0,0)} & p_{(1,0)}^{(0,0)} & \cdots & p_{(T-1,K)}^{(0,0)} \\ p_{(0,0)}^{(1,0)} & p_{(1,0)}^{(1,0)} & \cdots & p_{(T-1,K)}^{(1,0)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(0,0)}^{(T-1,K)} & p_{(1,0)}^{(T-1,K)} & \cdots & p_{(T-1,K)}^{(T-1,K)} \end{bmatrix} & \mathbb{TM} = \begin{bmatrix} p_{M_0}^{(0,0)} & p_{M_1}^{(0,0)} & \cdots & p_{M_{N_M}}^{(0,0)} \\ p_{M_0}^{(1,0)} & p_{M_1}^{(1,0)} & \cdots & p_{M_{N_M}}^{(1,0)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M_0}^{(T-1,K)} & p_{M_1}^{(T-1,K)} & \cdots & p_{M_{N_M}}^{(T-1,K)} \end{bmatrix} \\ \mathbb{MT} = \begin{bmatrix} p_{(0,0)}^{M_0} & p_{(1,0)}^{M_0} & \cdots & p_{(T-1,K)}^{M_0} \\ p_{(0,0)}^{M_1} & p_{(1,0)}^{M_1} & \cdots & p_{(T-1,K)}^{M_1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(0,0)}^{M_{N_M}} & p_{(1,0)}^{M_{N_M}} & \cdots & p_{(T-1,K)}^{M_{N_M}} \end{bmatrix} & \mathbb{MM} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix} \end{bmatrix}$$

Figure 5.6: The $\mathbb{Q}$ matrix in the proposed model

$$\mathbb{R}_{N_Q \times 3} = \begin{bmatrix} \mathbb{TA} = \begin{bmatrix} p_{Drop}^{(0,0)} & p_{Fail}^{(0,0)} & p_{Success}^{(0,0)} \\ p_{Drop}^{(1,0)} & p_{Fail}^{(1,0)} & p_{Success}^{(1,0)} \\ \vdots & \vdots & \vdots \\ p_{Drop}^{(T-1,K)} & p_{Fail}^{(T-1,K)} & p_{Success}^{(T-1,K)} \end{bmatrix} \\ \mathbb{MA} = \begin{bmatrix} p_{Drop}^{M_0} & p_{Fail}^{M_0} & p_{Success}^{M_0} \\ p_{Drop}^{M_1} & p_{Fail}^{M_1} & p_{Success}^{M_1} \\ \vdots & \vdots & \vdots \\ p_{Drop}^{M_{N_M}} & p_{Fail}^{M_{N_M}} & p_{Success}^{M_{N_M}} \end{bmatrix} \end{bmatrix}$$

Figure 5.7: The $\mathbb{R}$ matrix in the proposed model

receives the packet. Therefore, such elements are calculated using Equation 5.3.

$$p^{i,j}_{c_x,0} = prob^{link}(i, c_x) \times \prod_{t=1}^{x-1}(1 - prob^{link}(i, c_t)) \tag{5.3}$$

- **Transitioning to a state representing a retransmission or to the fail state:** In OR protocols, as explained in Chapter 2 Section 2.1, when a packet is transmitted, there is a chance that none of the candidates receive it. In this case, the sender node tends to retransmit the packet for a number of predetermined repetitions; this is when the state machine shifts to a retransmission state. If no candidate receives the packet after the maximum number of retransmissions, the packet is discarded from the network. This demonstrates a transition to the *Fail* state. Equation 5.4 shows how the probability value related to such transitions are calculated [61].

$$p^{i,j}_{i',j'} = 1 - \sum_{t=1}^{ncand} p^{i,j}_{c_t,0} \tag{5.4}$$

- **Transitioning from a malicious transient to a regular transient or *Drop* state:** This rule is used to calculate the probability of transitioning from a malicious transient to a regular transient or the *Drop* state. More precisely, it is used for salvaging data packets by lower-priority nodes in the candidate set. In fact, when a packet is transmitted, if a malicious candidate drops or maliciously manipulates it and sends a false acknowledgment, all lower-priority nodes in the candidate set will have the possibility of recognizing this misbehavior, and will be able to salvage the packet accordingly. Similar to packet forwarding, however, packet salvaging is also conducted according to the priority of nodes in the candidate set. In the proposed model, the highest-priority packet survivor will be the first benevolent node after the malicious one in the set of candidates. Furthermore, the probability of salvaging a packet will be equivalent to the link delivery probability between the malicious node and the highest-priority survivor; this value is calculated using an equation similar to Equation 5.2. For example, the probability of transitioning from state $(4, 1)$ to state $(3, 0)$ will be calculated according to this rule since node with $ID = 3$ is the first benevolent candidate for node with $ID = 1$ after the attacker node with $ID = 4$.

  Similar to the packet forwarding process, other lower-priority benign candidates can contribute to the packet salvaging process according to a probability that is calculated

using an equation similar to Equation 5.3. In the proposed model, all elements of matrix $\mathbb{MT}$ are calculated following these two rules.

In the end, if a malicious node drops a packet and no benign node can salvage it, the packet will be discarded from the network. In our proposed DTMC, this is represented by shifting from a malicious transient to the *Drop* state. The probability of this transition is calculated using an equation similar to Equation 5.4. This final rule is used to populate the elements of sub-matrix $\mathbb{MA}$.

- **Transitioning to absorbing states:** In a DTMC, the final transition occurs between a transient and one of the absorbing states. Once such a transition takes place, the state machine will remain in that state and no more transition occurs. This results in the creation of an identity matrix represented by $\mathbb{I}$ in the transition probability matrix $\mathbb{P}$.

5.1.3 Extracting Network Parameters from the Proposed Model

The main purpose for developing the proposed analytical model is to extract related network parameters. For instance, it would be of interest to determine how different parameters such as packet delivery ratio, drop ratio, hop count, and so forth will be calculated using the introduced model. In this chapter, a novel method of calculating delivery ratio, drop ratio, expected number of packet transmissions, and hop count is introduced. Furthermore, two novel network parameters, known as salvage ratio and direct delivery ratio, are proposed and calculated accordingly. The following subsections include greater details regarding the calculation of each parameter.

Packet Delivery, Drop, and Fail Ratio

The initial state of the DTMC in our proposed model is defined in the form of $V = [1, 0, 0, ..., 0]$. This means the state machine will initially be in a state corresponding to the source node [61]. Assuming this initial state, the probability of being in any other state in the Markov chain after exactly one transmission can be obtained from $V \times \mathbb{P}$. Similarly, the probability of being in any state after 2 transmissions can be obtained from $V \times \mathbb{P}^2$. Finally, $V \times \mathbb{P}^h$ indicates the status of the DTMC after h transitions between states. Considering that h is a large number, it would be possible to obtain the probability of a packet reaching its final destination, or the packet delivery ratio, using

element $(0, Dest)$ of $V \times \mathbb{P}^h$ [61]. Alternatively, it is possible to utilize the fundamental matrix of the Markov process, as shown in Equation 5.5. In this equation, $\mathbb{F}$ represents the probability of reaching any transient state after an infinite number of transitions. A full analysis and proof of this feature of absorbing Markov chains can be found in [79]. Finally, it would be possible to calculate the probability of reaching any absorbing state such as *Dest*, *Drop*, and *Fail* using $\mathbb{F} \times \mathbb{R}$. Elements of matrix $\mathbb{F} \times \mathbb{R}$ show the probability of successful packet delivery, the probability of packet failure due to unreliable network links, and the probability of packet drop due to node misbehaviors.

$$\mathbb{F} = (I - \mathbb{Q})^{-1} \tag{5.5}$$

However, apart from the explained approach, another method of calculating the packet delivery probability is to specifically consider different cases in which a packet can reach its destination. More precisely, as explained in Section 5.1.3, a packet might be successfully delivered without experiencing a drop by malicious nodes, and subsequently a salvage by backup candidates. Moreover, the packet may reach the destination after being dropped and salvaged exactly once, twice, and so forth. As shown in Equation 5.6, the process of salvaging a packet is modeled by transitioning from regular transient to malicious transient, and subsequently from malicious transient to regular transient states.

$$\mathbb{S} = \mathbb{TM} \times \mathbb{MT} \tag{5.6}$$

Before or after each occurrence of the packet salvaging process, however, multiple transitions might take place between regular transient states, as shown in Fig. 5.4. These transitions are shown with matrix $\mathbb{F}'$, as represented in Equation 5.7. More specifically, considering only matrix $\mathbb{TT}$, a second version of the Markov fundamental matrix can be assumed and calculated using Equation 5.7. Therefore, it would be possible to demonstrate each step of the packet salvaging, and subsequent transitions between regular transient states using matrix $\mathbb{Y}$, as shown in Equation 5.8.

$$\mathbb{F}' = (I - \mathbb{TT})^{-1} \tag{5.7}$$

$$\mathbb{Y} = \mathbb{TM} \times \mathbb{MT} \times \mathbb{F}' \tag{5.8}$$

In the end, Equation 5.9 formulates the packet delivery process for the proposed model. The element $(0, Dest)$ of matrix $\mathbb{X}$ results in the probability of packet delivery, or packet delivery ratio.

$$\begin{aligned}\mathbb{X} = \; & \mathbb{F}' \times \mathbb{TA} + \\ & \mathbb{F}' \times \mathbb{Y} \times \mathbb{TA} + \\ & \mathbb{F}' \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{TA} + \\ & \mathbb{F}' \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{TA} + ... \\ = \; & \mathbb{F}' \times (I + \mathbb{Y} + \mathbb{Y}^2 + \mathbb{Y}^3 + ...) \times \mathbb{TA} \\ = \; & \mathbb{F}' \times (I - \mathbb{Y})^{-1} \times \mathbb{TA}\end{aligned} \tag{5.9}$$

Direct-delivery Ratio

Direct delivery ratio, which is a novel parameter introduced in this chapter, represents the ratio of packets that have reached the destination without being dropped by malicious nodes, and subsequently salvaged by backup candidates. In the proposed DTMC, it would be possible to calculate the direct delivery ratio by considering multiple transitions specifically between regular transient states, on the condition that the packet finally reaches its final destination. More precisely, by calculating $\mathbb{F}' \times \mathbb{TA}$, it would be possible to obtain the probability of reaching any absorbing state after multiple transitions between regular transient states. Element $(0, Dest)$ in $\mathbb{F}' \times \mathbb{TA}$ demonstrates a successful packet delivery in this situation or direct delivery ratio. It is worth noting that the direct delivery of packets in the proposed model will be equal to the actual delivery ratio of packets, if no salvaging mechanism exists.

Salvage Ratio

Salvage ratio, which is another parameter proposed in this chapter, demonstrates how effectively the proposed packet salvaging model can improve the communication performance in hostile environments. More precisely, it represents the proportion of packets that, first, have been dropped by malicious nodes, then salvaged by backup candidates, and finally reached their destination. Salvage ratio can be calculated using two different approaches, as will be discussed in this section, in greater detail. The first method is to simply calculate the difference between the actual packet delivery ratio, as explained in Section 5.1.3, and the direct delivery ratio as discussed in 5.1.3. This will result in the probability of a packet delivery, on the condition that the packet has not been directly delivered to its destination. In other words, the difference between the actual delivery

ratio and the direct delivery ratio creates the probability of a successful packet delivery, given that the packet is definitely dropped, and then salvaged by backup candidates.

The second approach to calculating salvage ratio is to examine several transitions between regular and malicious transient states in the proposed DTMC using matrices $\mathbb{TT}$, $\mathbb{TM}$, $\mathbb{MT}$, and $\mathbb{TA}$. More precisely, as shown in Equation 5.10, the probability of salvaging a packet can be obtained from matrix $\mathbb{X}'$ considering one or multiple transitions between regular transient and malicious transient states, on the condition that the packet is finally delivered to the destination.

$$\begin{aligned} \mathbb{X}' = \; & \mathbb{F}' \times \mathbb{Y} \times \mathbb{TA} + \\ & \mathbb{F}' \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{TA} + ... \\ & \mathbb{F}' \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{Y} \times \mathbb{TA} + ... \\ = \; & \mathbb{F}' \times (\mathbb{Y} + \mathbb{Y}^2 + \mathbb{Y}^3 + ...) \times \mathbb{TA} \\ = \; & \mathbb{F}' \times ((I - \mathbb{Y})^{-1} - I) \times \mathbb{TA} \end{aligned} \tag{5.10}$$

Expected Number of Transmissions (ETX)

ETX demonstrates the expected number of required transmissions for successfully delivering a packet to the destination. The focus of OR protocols is to reduce ETX in order to optimize the consumption of network resources [61]. In a regular DTMC similar to Fig. 5.1, this can be obtained by calculating the expected value of the random variable X, where X shows the number of required transitions to reach the *Dest* state from one of the transient states. Basically, the probability of reaching an absorbing state in exactly h transitions can be calculated using Equation 5.11. This represents $h - 1$ transitions between transient states, and one transition from transient to absorbing states.

$$P[X = h] = \mathbb{Q}^{h-1} \times \mathbb{R} \tag{5.11}$$

The expected value of the random variable X, or ETX, can be calculated using Equation 5.12, given that matrices in the nominator and denominator are divided element by element [61].

$$
\begin{aligned}
E_{total}[X] = & \frac{\sum_{h=1}^{\infty}(h \times p[X=h])}{p[X=1]+p[X=2]+P[X=3]+...} \\
= & \frac{1\times\mathbb{R}+2\mathbb{Q}\times\mathbb{R}+3\mathbb{Q}^2\times\mathbb{R}+...}{\mathbb{R}+\mathbb{Q}\times\mathbb{R}+\mathbb{Q}^2\times\mathbb{R}+...} \\
= & \frac{(I-\mathbb{Q})^{-2}\times\mathbb{R}}{(I-\mathbb{Q})^{-1}\times\mathbb{R}} = \frac{\mathbb{F}^2\times\mathbb{R}}{\mathbb{F}\times\mathbb{R}}
\end{aligned}
\tag{5.12}
$$

In the proposed model, however, this method will not result in an accurate value for ETX. For packets that are dropped and then salvaged, the transition between malicious transient and regular transient states is counted as one transmission in Equation 5.12. However, in reality, such transitions in the proposed DTMC do not demonstrate a data packet transmission. Instead, they are simulated as part of the coordination method between candidates in OR protocols. In Fig. 5.2, for example, the transition between states $(4,1)$ and $(3,0)$ should not be considered as a separate transmission when calculating the expected number of transmissions. In order to address this issue, we propose to calculate the expected number of such extra transmissions using Equation 5.13, and deduct this value from the previously calculated ETX in Equation 5.12. Basically, if a packet is directly delivered to the destination, there will be no extra transmissions ($h=0$). If it is salvaged only once, there will be one extra transmission ($h=1$) related to the shift from malicious transient to regular transient states, and so forth.

$$
\begin{aligned}
E_{extra}[X] = & \frac{\sum_{h=0}^{\infty}(h \times p[X=h])}{p[X=0]+p[X=1]+p[X=2]+...} \\
= & \frac{[0\times\mathbb{F}'\times\mathbb{TA}]+[1\times\mathbb{F}'\times\mathbb{Y}\times\mathbb{TA}]+[2\times\mathbb{F}'\times\mathbb{Y}\times\mathbb{Y}\times\mathbb{TA}]+...}{[\mathbb{F}'\times\mathbb{TA}]+[\mathbb{F}'\times\mathbb{Y}\times\mathbb{TA}]+[\mathbb{F}'\times\mathbb{Y}\times\mathbb{Y}\times\mathbb{TA}]+...} \\
= & \frac{\mathbb{F}'\times\mathbb{Y}\times(I+2\times\mathbb{Y}+3\times\mathbb{Y}^2+...)\times\mathbb{TA}}{\mathbb{F}'\times(I+\mathbb{Y}+\mathbb{Y}^2+...)\times\mathbb{TA}} \\
= & \frac{\mathbb{F}'\times\mathbb{Y}\times(I-\mathbb{Y})^{-2}\times\mathbb{TA}}{\mathbb{F}'\times(I-\mathbb{Y})^{-1}\times\mathbb{TA}}
\end{aligned}
\tag{5.13}
$$

Regarding Equation 5.13, it is worth noting that the expected number of extra transmissions is conditioned for delivering the packet to its destination. Consequently, the denominator in this equation will equal to the probability of the packet reaching its final destination, as shown in Equation 5.9. In the end, the final ETX value will be calculated using the difference between $E_{total}[X]$ and $E_{extra}[X]$ as shown in Equation 5.14.

$$E[X] = E_{total}[X] - E_{extra}[X] \tag{5.14}$$

Hop Count

The hop count is defined as the expected number of hops required for a packet to reach its destination. As explained in [61], the hop count is calculated considering transitions between transient states that are related to different nodes. More accurately, transitions related to packet retransmissions should be excluded when calculating hop count. Therefore, it would be possible to combine all of the states related to packet retransmissions into one single state, and create a new version of the Markov chain. This requires the recalculation of the transition probability matrix for the new Markov chain. In this version, each transition takes place between states related to different nodes, with the only difference that each transition also includes retransmissions. The detailed procedure of creating this Markov chain, and calculating the transition probability matrix, is thoroughly discussed in [61].

$$\begin{aligned} p^i_{c_1} &= p^{i,j}_{c_1,0} + p^i_{ReTx} \times p^{i,j}_{c_1,0} + (p^i_{ReTx})^2 \times p^{i,j}_{c_1,0} + ... \\ &= \sum_{t=0}^{K} (p^i_{ReTx})^t \times p^{i,j}_{c_1,0} \\ p^i_{c_x} &= p^{i,j}_{c_x,0} + p^i_{ReTx} \times p^{i,j}_{c_x,0} + (p^i_{ReTx})^2 \times p^{i,j}_{c_x,0} + ... \\ &= \sum_{t=0}^{K} (p^i_{ReTx})^t \times p^{i,j}_{c_x,0} \end{aligned} \tag{5.15}$$

Equation 5.15 demonstrates how the probability of transitioning between different states is calculated in the new Markov chain. It is worth noting that calculation of transition probabilities depends upon the number of nodes in the candidate set, as shown in Equation 5.15. In this equation, p^i_{retx} is the probability of a packet retransmission by node i, and is calculated using Equation 5.4. Similarly, $p^{i,j}_{c_1,0}$ and $p^{i,j}_{c_x,0}$ are extracted

from Equations 5.2 and 5.3 respectively [61]. Finally, $p^i_{c_1}$ and $p^i_{c_x}$ are the probability of shifting from node i to its highest-priority candidate, say c_1, and its other candidates, shown as c_x, in the new Markov chain. Upon creation of the new Markov chain and calculating all probability values, the hop count can be calculated using the expected number of transmissions between the source and the destination, say ETX, as explained in Section 5.1.3.

5.2 Performance Analysis

This section includes a performance evaluation on the proposed model. For evaluation purposes, the introduced analytical model has been implemented using Java programming language. Furthermore, experiments have also been performed using network simulation. Network simulator (ns-2.35) [68] has been used as the simulation tool. Performing experiments, both analytically and using simulation, provides the possibility of comparing and verifying obtained results. In addition, all results are extracted from an average of 100 executions, considering 95% confidence intervals. In each execution, the distribution of nodes in the network field is randomly changed, while other parameters are kept invariant. Section 5.2.1 describes network setting parameters, whereas experimental results are reported in Section 5.2.2.

5.2.1 Network Scenario

In this section, different network parameters are reported by changing important variables such as the number of malicious nodes in the network, the maximum number of candidates in the candidate set, and the maximum number of permitted retransmissions. Moreover, although the proposed packet salvaging model is a general and comprehensive model applicable in any OR algorithm, EXOR protocol [12] has been selected as the case study for performing experiments. In order to demonstrate the effectiveness of the proposed model, all of the results are compared with a baseline protocol. The baseline model is basically similar to the proposed model, with the only difference that no packet salvaging takes place for this algorithm. A schema of the DTMC for the baseline protocol is represented in Fig. 5.1. This protocol has been discussed in greater details in Chapter 4.

As discussed in Section 5.1.2, however, the proposed model is created considering

Table 5.2: Simulation parameters

Parameter	Value
Propagation Model	Shadowing
MAC	802.11
Number of Nodes	40
Field Size	$500 \times 500\ m^2$
Data Payload Size	512 bytes
Maximum Number of Candidates	1 to 6
Number of malicious nodes	0 to 15
Maximum number of retransmissions	0 to 6
Transmission Rate	5 Packets/Second
Coordination Delay	15 Milliseconds
Simulation Time	1800 Seconds

the quality of links between nodes. Therefore, following [60], in order to model the existence of channel obstructions and path loss, the shadowing propagation model has been utilized in the experiments. More accurately, the quality of links between nodes in the shadowing propagation model is determined as a function of the distance between nodes, the path loss exponent, and so forth. More details of calculating the link delivery probability between nodes can be found in [68] and [60].

Table 5.2 shows the list of parameters used in experiments. In all of the evaluations, one stream of source-destination data flow has been modeled in which the source node is located at the bottom-left corner of the network field, whereas the destination is located at the top-right corner. Other nodes are randomly distributed in the environment. This provides the possibility of creating the longest possible distance between the source and the destination nodes in the simulated scenario. Performance evaluations are done in a network environment of size $500 \times 500\ m^2$ that contains 40 nodes. IEEE 802.11 has been selected as the MAC layer, and the time of simulations is 1800 seconds. Regarding the rate of generating data packets, the source node generates packets of size 512 bytes at the rate of 5 packets per second. As explained in the following subsections, the maximum number of candidates, the number of malicious nodes in the network, and the maximum number of retransmissions are variables in various scenarios.

5.2.2 Evaluation Results

This subsection includes an extensive set of experimental results for the proposed model. In fact, Section 5.2.2 compares the results of the analytical model against simulation results, whereas Sections 5.2.2 and 5.2.2 contain three-dimensional graphs representing results of baseline and proposed protocols using the analytical approach.

Analytical versus Simulation Results

Fig. 5.8 visually compares analytical versus simulation results for two different scenarios, considering the effects of changing the number of malicious nodes and the number of candidates. Furthermore, three different parameters including packet delivery ratio, drop ratio, and salvage ratio have been selected and examined. The set of figures on the left demonstrates the effects of changing the number of malicious nodes when the maximum number of candidates is 3, whereas the right-side plots show the effects of the number of candidates when 8 malicious nodes exist in the network. As observed, in all of the plotted figures, simulation results prove to be considerably close to analytical results.

The two figures at the top show the packet delivery ratios. As observed, by increasing the number of malicious nodes, the packet delivery ratio decreases. However, the trend of this decrease for the baseline protocol is considerably higher than the proposed salvaging model, such that the delivery ratio is almost 40% higher in the proposed model, when the number of malicious nodes is 8. This conveys how effectively backup candidates are able to nullify the effects of the attacker nodes. On the other side, as represented in the top right plot, increasing the number of nodes in the candidate set leads to the higher efficiency of the proposed model. For instance, when the maximum number of candidates is 6, the delivery ratio of packets for the proposed model is about 80%. This is reasonable, because including a higher number of nodes in the candidate set will result in a higher probability of packet salvaging, which subsequently leads to a higher delivery ratio. In the baseline protocol, however, no significant change is observed in the delivery ratio when the number of candidates increases.

Similarly, the two figures in the middle represent the drop ratio of packets for both scenarios. As shown, the drop ratios in both protocols increase by increasing the number of malicious nodes. However, the slope of this increase for the proposed model is considerably smaller than the baseline algorithm. Moreover, increasing the number of nodes in the candidate set will reasonably result in a smaller drop ratio in the proposed

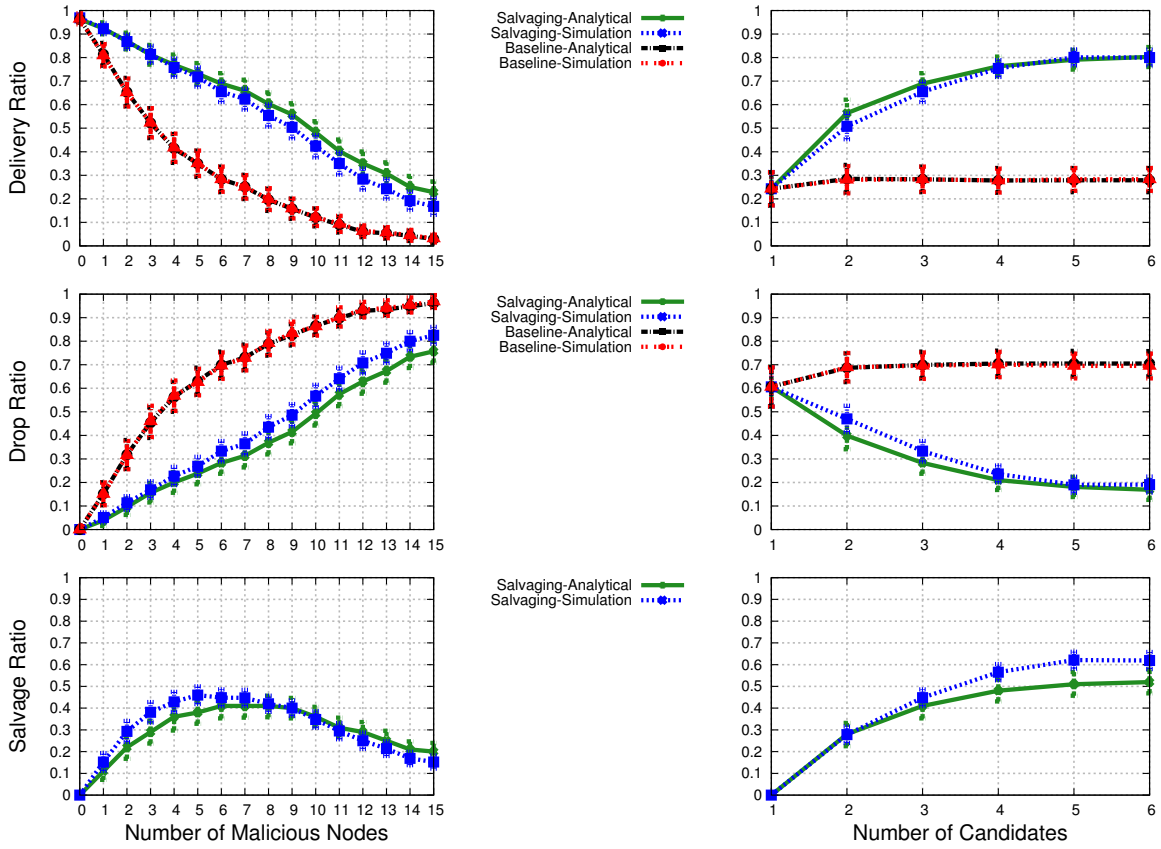

Figure 5.8: Analytical versus simulation results

model. As mentioned earlier, this is due to the fact that a higher number of candidates will result in the salvage of a greater number of packets.

Finally, the salvage ratio for the proposed model is depicted in the last two figures. By increasing the number of malicious nodes, it is interesting to see that the salvage ratio of packets for the salvaging model increases up to a certain level, and decreases afterwards. The reason is that the salvaging model can resist against malicious nodes up to a certain level, when there is a sufficient number of backup nodes in the candidate set. However, increasing the number of malicious nodes will result in a smaller number of backup candidates, and finally, a smaller delivery ratio. On the other hand, as shown in the bottom right figure, increasing the number of candidates results in a greater salvage ratio in the proposed model.

Effect of Candidates

This subsection includes three-dimensional figures representing the effects of changing both the number of malicious nodes, and the number of candidates using the proposed analytical model. Fig. 5.9 shows the trend of changes in the delivery ratio. As shown in Fig. 5.9 and mentioned in Section 5.2.2, increasing the number of malicious nodes will result in a higher decrease in the delivery ratio of the baseline protocol compared to the salvaging protocol. Increasing the number of candidates, however, will assist in salvaging a significant number of dropped packets and, as a result, lead to a greater packet delivery ratio.

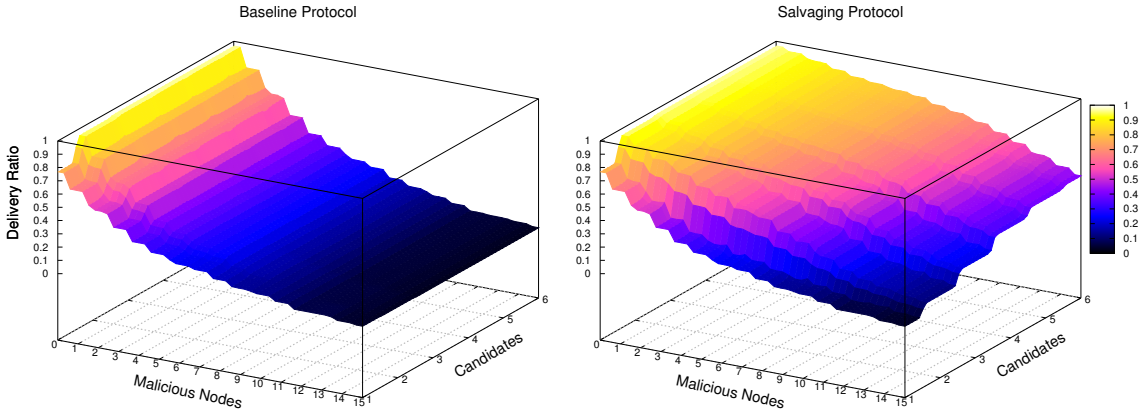

Figure 5.9: Delivery ratio for baseline and proposed protocols when the maximum number of retransmissions is 3

Furthermore, Fig. 5.10 compares the drop ratio of the proposed salvaging model and the baseline protocol. As observed, the salvaging model can effectively control the negative effects of malicious nodes. More precisely, the drop ratio of packets for the baseline protocol rises by increasing the number of malicious nodes, regardless of the number of candidates. The reason is that although increasing the number of candidates assists in a more reliable OR protocol, the probability of selecting a malicious candidate also increases. Regarding the proposed salvaging model, however, the increase in the drop ratio is smaller compared to the baseline algorithm; in addition, an increase in the number of candidates helps decrease the drop ratio significantly.

Similarly, Fig. 5.11 shows the expected number of transmissions (ETX) for both baseline and proposed protocols. As observed, the ETX value for both protocols slightly

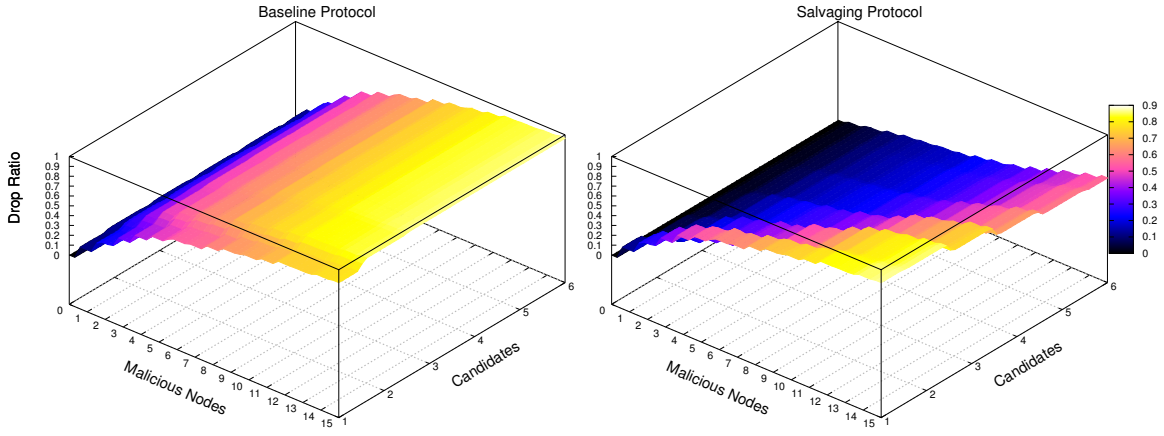

Figure 5.10: Drop ratio for baseline and proposed protocols when the maximum number of retransmissions is 3

decreases, as the number of malicious nodes increases. The reason is that with a higher number of malicious nodes, such nodes will not collaborate in packet forwarding, and the delivery ratio will decrease. In this situation, if a packet reaches the destination, it should have travelled from a secure and reliable path; also, using reliable paths results in a smaller number of transmissions and retransmissions between nodes. Furthermore, the ETX value for the proposed model is slightly lower than the baseline protocol. The reason is that in the baseline protocol, most of the packets are dropped by malicious nodes. Packets that reach their destination, however, must be transmitted or retransmitted by benevolent candidates when it is necessary to do so. In the salvaging protocol, on the other hand, a considerable ratio of packets are salvaged and reach their destination. Salvaging a packet will not only improve the delivery ratio but will also progress packets one hop closer to their destination. This results in a smaller overall ETX of the salvaging protocol. Increasing the number of candidates, on the other hand, will decrease the chance of packet loss, and subsequently the need for a higher number of retransmissions. Therefore, the ETX value decreases while the number of candidates increase in both protocols.

In addition, as shown in the lowest two graphs in Fig. 5.12, the hop count in sending packets increases in accordance with the number of candidates. This is reasonable because a higher number of candidates will increase the flexibility of creating longer but more reliable routes towards the destination. Therefore, a smaller number of packets

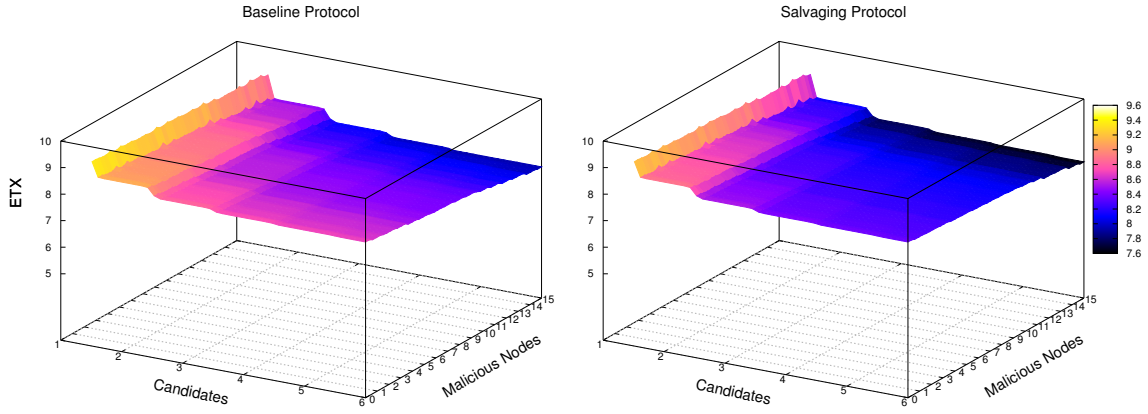

Figure 5.11: ETX for baseline and proposed protocols when the maximum number of retransmissions is 3

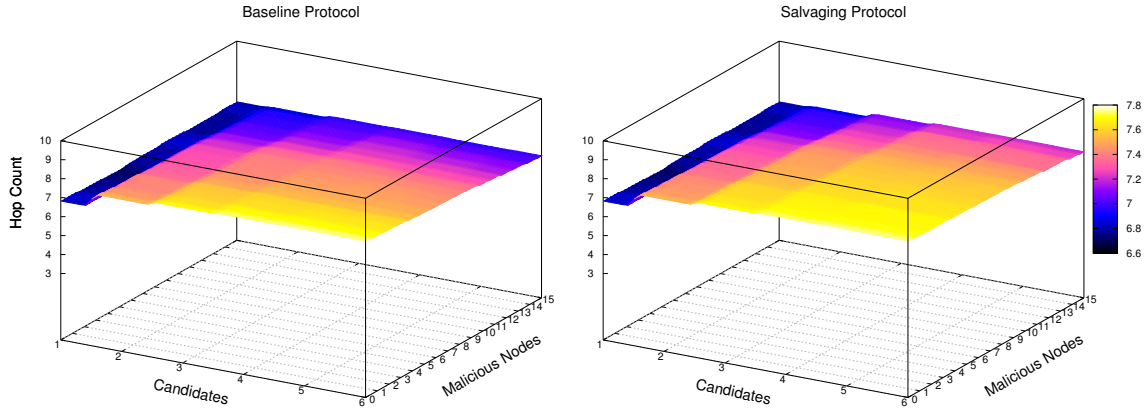

Figure 5.12: Hop count for baseline and proposed protocols when the maximum number of retransmissions is 3

will get lost, due to path loss and channel obstructions. This, however, will lead to an increase in the hop count value in both protocols.

Fig. 5.13 shows the salvage ratio of packets for the proposed model. As observed, the salvage ratio escalates up to 60% by including a higher number of nodes in the candidate set. Similarly, by increasing the number of malicious nodes, a higher number of packets are dropped, and consequently, a larger number of packets are salvaged. However, if the number of malicious nodes is too high, the number of backup candidates decreases in the candidate set, which leads to a decrease in the salvage ratio. As explained in

Section 5.1.3, the direct delivery ratio of the proposed model reasonably shows the same results as the delivery ratio of the baseline protocol.

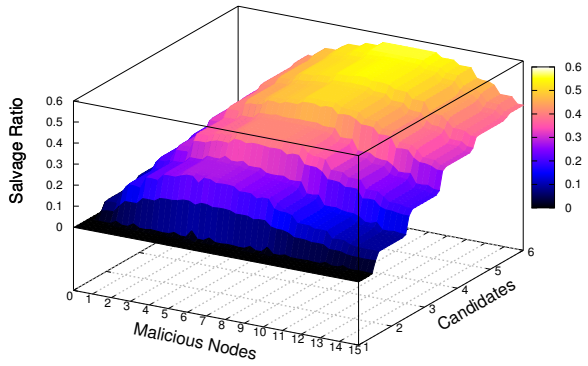

Figure 5.13: Salvage ratio for the proposed model when the maximum number of retransmissions is 3

Effect of Retransmissions

This subsection studies the effects of changing the maximum number of retransmissions on different network parameters, when the number of candidates is limited to 4. As observed in Fig. 5.14, increasing the number of attempts to retransmit a packet has a significant effect in the delivery of packets to their destination for the proposed salvaging protocol. Regarding the baseline algorithm, this increase is effective only when the number of malicious nodes is negligible. For example, with 5 attackers or more, the delivery ratio of the baseline protocol does not significantly change by increasing the number of retransmissions. Furthermore, it is obvious that the salvaging protocol has a higher delivery ratio compared to the baseline algorithm.

Similarly, the trend of changes in packet drop ratio for both protocols is shown in Fig. 5.15. As observed, the negative effects of malicious nodes on the baseline protocol is much more noticeable than the proposed salvaging mechanism. This represents that the proposed model is proportionally capable of defending against malevolent entities. In addition, it is interesting to see that increasing the number of retransmissions does not have a considerable effect on the drop ratio of either protocols.

Fig. 5.16 visualizes the effects of the number of retransmissions on ETX. Logically, increasing the number of retransmissions results in an increase in the expected number

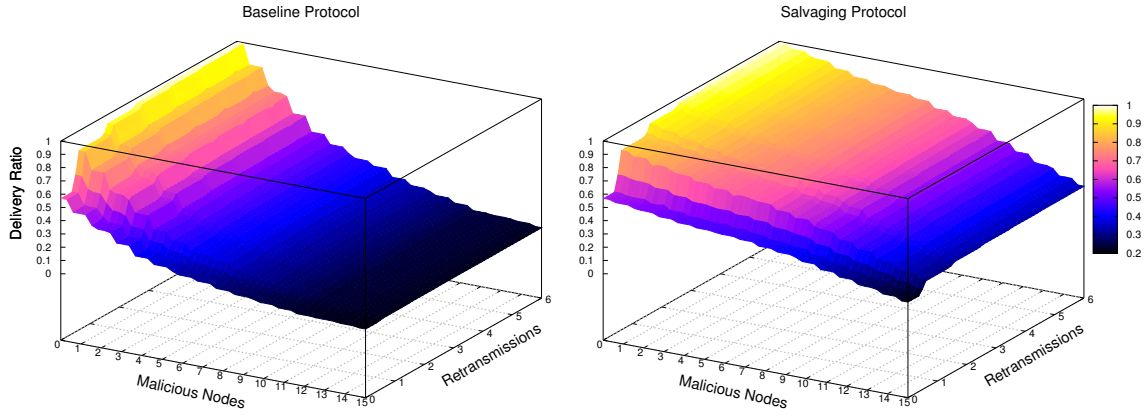

Figure 5.14: Delivery ratio for baseline and proposed protocols when the maximum number of candidates is 4

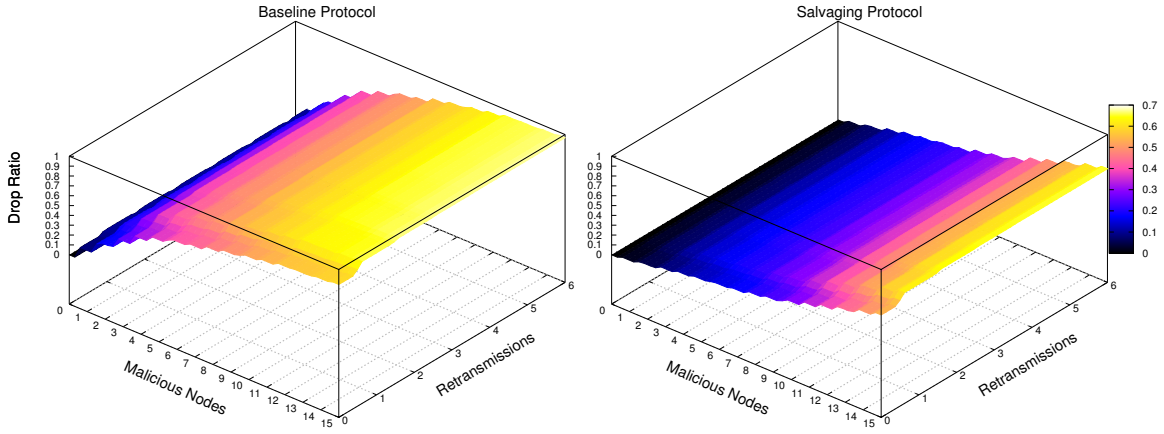

Figure 5.15: Drop ratio for baseline and proposed protocols when the maximum number of candidates is 4

of required transmissions to send packets to the destination. This occurs because packets will have a higher chance of reaching their destination, but in a higher number of transmissions. This is shown in Fig. 5.16 for both baseline and proposed protocols. In addition, as explained in Section 5.2.2, including a higher number of malicious nodes in the network will result in a slightly lower ETX for both protocols.

Fig. 5.17 demonstrate the effects of changing the number of retransmissions on hop count. As observed in both baseline and the proposed protocols, a higher number of retransmissions does not have a noticeable effect on hop count. This is normal in that

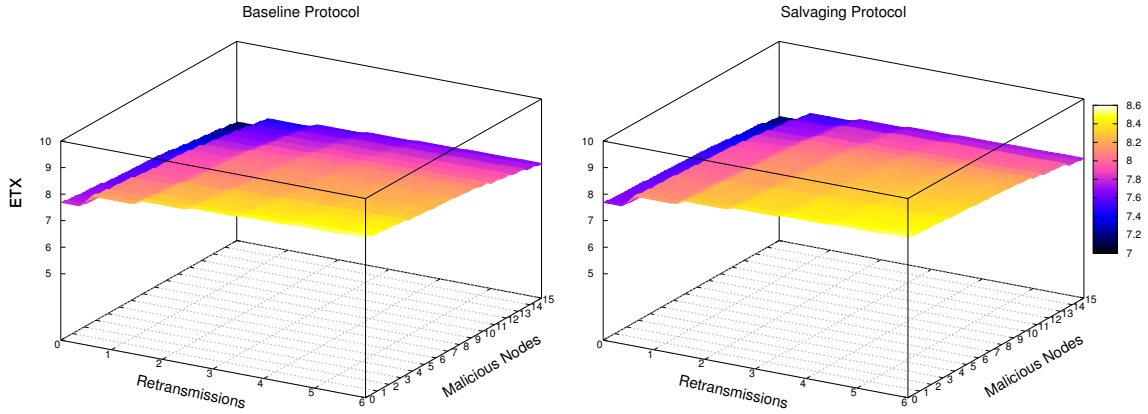

Figure 5.16: ETX for baseline and proposed protocols when the maximum number of candidates is 4

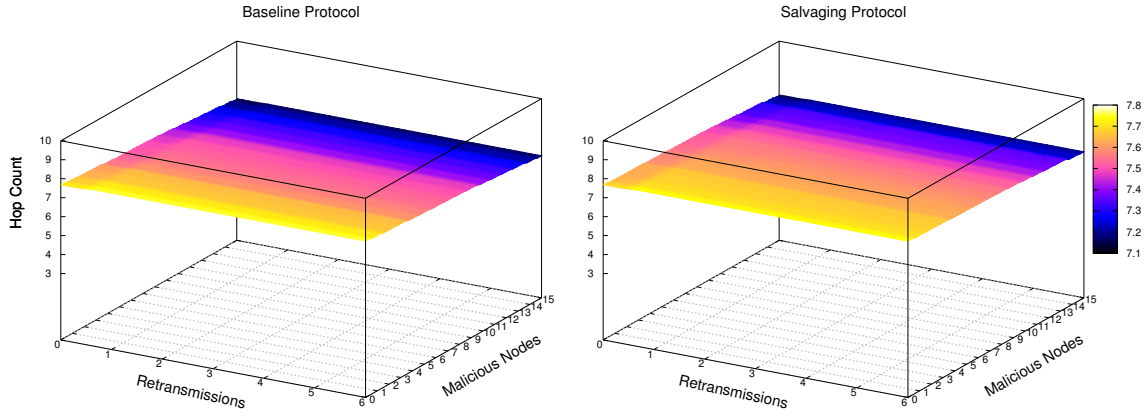

Figure 5.17: Hop count for baseline and proposed protocols when the maximum number of candidates is 4

increasing the number of retransmissions escalates the probability that a packet will reach its destination. However, since retransmissions take place in the same node, the hop count does not significantly change with more retransmissions. In contrast, including a greater number of malicious nodes in the network will result in a decrease in hop count. The reason for this decrease is that by increasing the number of attackers, a large number of packets are grabbed and discarded by such nodes. On the other hand, packets that travel from longer paths will have a higher chance of being dropped by malicious nodes. This conveys that successfully delivered packets should have travelled from shorter paths

to be safe from being captured by attackers.

Finally, the salvage ratio of packets for the proposed model is presented in Fig. 5.18. As explained earlier in this section, increasing the number of retransmissions leads to an increase in the salvage ratio. On the other hand, increasing the number of malicious nodes initially rises the salvage ratio, and results in a decrease of this parameter afterwards.

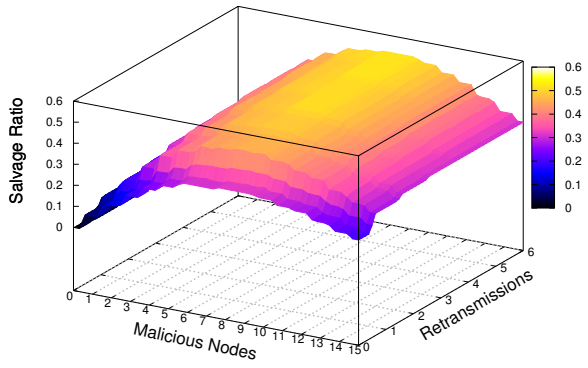

Figure 5.18: Salvage ratio for the proposed model when the maximum number of candidates is 4

Communication Overhead

In this subsection, we provide an analysis on the overhead of the proposed protocol. The evaluation is performed in terms of the number of generated coordination messages between candidates. In fact, when a node in the candidate set forwards a message, it broadcasts an acknowledgment control packet to notify other candidates of this transmission. By receiving such a coordination message, other candidates prevent forwarding the corresponding data packet. In our simulations, we divide the total number of transmitted control packets in the network by the number of generated data packets to obtain the coordination overhead. In fact, the coordination overhead is calculated for all generated packets, although some might be dropped by malicious nodes or discarded due to the unreliability of links.

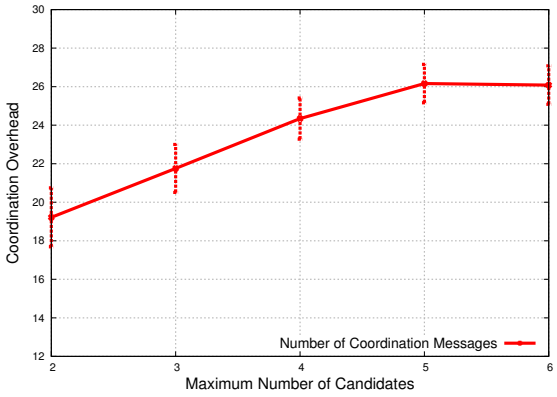

Figure 5.19: Communication overhead between candidate nodes

Fig. 5.19 visualizes the overhead caused by candidates for the network scenario explained in Section 5.2.1, when 8 malicious nodes exist in the network. As observed, increasing the number of candidates results in a higher number of coordination messages. However, it should be noted that this increase is acceptable considering the hop count and number of retransmissions taking place. For example, when the number of candidates is 5, the average number of coordination messages is about 26 for the entire path between the source and the destination. Furthermore, as shown in Fig 5.12, the hop count of delivering packets to their destination is around 7.5 and a maximum of three retransmissions are allowed to take place. Therefore, on average, about 3.5 coordination messages are generated in each hop, considering all retransmissions. As a result, although coordination packets are small in size and do not carry data, delivering a higher number of data packets to their destination in a more reliable network is achieved by transmitting a higher number of such messages. This trade-off can be justified considering various requirements for different applications.

5.3 Summary

In this chapter, we introduced a novel packet salvaging mechanism that benefits from the redundancy of nodes involved in each hop of OR protocols. The proposed model represented how benign nodes in the set of candidates can collaborate with each other, and salvage data packets that are maliciously dropped or manipulated by attacker nodes. The Discrete-Time Markov Chain (DTMC) was adopted as the modeling tool in the proposed model. Using DTMC, a model was proposed that shows how backup nodes in

the candidate set can effectively collaborate with each other and salvage data packets. Furthermore, we introduced a novel method of calculating different network parameters including delivery ratio, drop ratio, ETX, and hop count, as well as two new parameters known as salvage ratio and direct delivery ratio. In addition, although the proposed model is a general approach independent of any OR protocol, the introduced model was implemented as an extension to EXOR [12] protocol, and extracted analytical results were verified by network simulation. Performance evaluation results represented that the proposed model is effectively able to handle the transmission of packets to their destination. In fact, comparing the results with a baseline algorithm demonstrated that the proposed model is able to more efficiently resist against malicious nodes. More precisely, a larger number of packets are salvaged, and a smaller number are dropped by malicious nodes in the introduced protocol. Furthermore, it was observed that increasing the number of nodes in the candidate set, and the number of allowed retransmissions, can considerably boost the performance of the proposed model when delivering packets to their destination. This improvement is achieved with the cost of generating a greater number of coordination messages between candidates.

Chapter 6

Conclusion and Future Work

Dealing with security obstacles of wireless networks is highly significant. In this thesis, we focused on trust management and security analysis protocols for OR-based wireless networks. In fact, after providing an introduction and defining the problem statement in Chapter 1, we studied the state of the arts in the areas of OR protocols, security challenges in wireless networks, and trust management schemes in Chapter 2. Furthermore, a classification and comparison of different security solutions provided specifically for OR protocols was provided in Chapter 2. A summary of major contributions of this thesis, that are all focused on trust and security analysis of OR protocols, is presented in the following paragraphs.

- Due to the high importance of trust management protocols, we proposed a novel trust model to deal with malicious nodes in OR-based wireless networks. More specifically, the introduced model is customized to benefit from both the reliability of OR protocols and the security of trust managements schemes. In this protocol, three different candidate selection metrics were proposed, considering different important aspects such as trust value of nodes, reliability of links, and geographical location of neighboring nodes. These metrics, that are called RTOR, TORDP, and GEOTOR, enable nodes to select their candidates based upon different applications requirements and environments. Furthermore, A novel watchdog mechanism was proposed and applied in the proposed trust model, that is specifically developed for OR protocols. The results of conducted simulations demonstrated that the proposed trust model can effectively act against malicious nodes by recognizing and excluding them from the network.

- As an extension to the developed trust model, that uses only direct interactions between nodes for trust calculation, a variation of the introduced scheme was designed in Chapter 3, that utilized both direct and indirect interactions. In fact, when nodes calculate or recalculate trust value of their candidates, they send recommendation packets to their neighbors and share their experience with them. Furthermore, considering perfect coordination between nodes, a variation of the proposed watchdog mechanism was developed that can be applied in any OR protocol regardless of the maximum size of candidate set. Results of performance evaluations demonstrated that this protocol can efficiently nullify negative effects of malicious nodes.

- Apart from the introduced trust management system, we developed an analytical model in Chapter 4, that assists in evaluating the performance of a wireless network more accurately. In fact, we utilized Discrete-Time-Markov-Chain (DTMC) to show the existence of malicious or uncooperative nodes in an OR-based wireless network. Moreover, a new network parameter was introduced and extracted from the developed DTMC, creating the possibility of calculating the ratio of dropped packets by malicious nodes more accurately. Since the introduced model is a comprehensive and general approach that is independent of any specific OR protocol, it was implemented and applied to four different well-known OR protocols including EXOR, POR, DPOR, and MTS. In continue, various related network parameters were extracted and calculated from the proposed model for all protocols using both analytical approach and network simulation. A comparison of analytical versus simulation results demonstrated that the introduced algorithm is able to efficiently represent the behavior of OR protocols, when malicious or uncooperative nodes exist in the network.

- In Chapter 5, we proposed a packet salvaging model that is an extended version of the proposed model in Chapter 4. In fact, the idea of the proposed packet salvaging model is to utilize the redundancy of nodes in the candidate set, and save packets that are dropped by malicious nodes through the help of backup nodes in the candidate set. Here too, the DTMC was used as the modeling tool to demonstrate an OR-based wireless network. In continue, different network parameters were extracted from the proposed model, and two novel parameters were introduced known as packet salvage ratio and direct-delivery ratio. Finally, an extensive set

of both simulation and analytical tests were conducted to assess the efficiency of the proposed model. Evaluation results demonstrated that the proposed model is capable of salvaging a considerable proportion dropped packets, when malicious nodes and packet droppers exist in the network.

6.1 Future Works

Some of the possible future directions of this research are as follows.

- Regarding the proposed analytical model, it is still not possible to defend against malicious nodes by excluding them from the network. Therefore, it would be interesting to integrate the concept of trust calculation and management into the introduced analytical model. As a result, it would be possible to create different Markov chains in each iteration of the trust calculation process and calculate related network parameters accordingly.

- The current version of all proposed protocols has been applied to wireless mesh networks. It would be interesting to customize proposed protocols for other applications and network environments such as wireless sensor networks. In this case, energy consumption of nodes should also be involved as a major parameter when selecting candidates. In fact, one possible future work is to design and evaluate energy efficient, reliable, and secure OR-based wireless sensor networks.

- In this thesis, all of the proposed protocols and analytical models have been implemented in a static network that lacks node mobility. In fact, introducing the mobility of nodes will add another level of complexity especially to analytical models. As part of the future works, it would be interesting to design and develop analytical approaches that are applicable in mobile ad-hoc and vehicular networks.

- Scalability of network solutions is always an important aspect. For example, considering the scope of vehicular ad-hoc networks, the system should properly function in presence of hundreds or even thousands of nodes. As part of the future work, we aim to perform a more comprehensive set of scalability tests on the proposed protocols, and design more scalable algorithms in the area of security analysis for OR protocols.

- The current version of proposed protocols considers a limited category of security attacks in the network. In real-world applications, however, a wide variety of different attacks might take place in the network. As part of the future work, we would like to introduce different types of security attacks in the network, and analyse the behavior of network accordingly. In continue, we are planning to make proposed protocols capable of defending against such misbehaviors.

Bibliography

[1] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61(1):275–285, 2012.

[2] Asmaa Adnane, Christophe Bidan, and Rafael Timoteo De Sousa. Trust-based countermeasures for securing olsr protocol. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 2, pages 745–752. IEEE, 2009.

[3] Sudhir Agrawal, Sanjeev Jain, and Sanjeev Sharma. A survey of routing attacks and security measures in mobile ad-hoc networks. *arXiv preprint arXiv:1105.5623*, 2011.

[4] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.

[5] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[6] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37:380–392, 2014.

[7] Thanasis Antoniou, Ioannis Chatzigiannakis, George Mylonas, Sotiris Nikoletseas, and Azzedine Boukerche. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. In *Proceedings of the 37th annual symposium on Simulation*, page 43. IEEE Computer Society, 2004.

[8] Alireza Attar, Helen Tang, Athanasios V Vasilakos, F Richard Yu, and Victor CM Leung. A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, 100(12):3172–3186, 2012.

[9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[10] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[11] N Bhalaji and A Shanmugam. Reliable routing against selective packet drop attack in dsr based manet. *Journal of Software*, 4(6):536–543, 2009.

[12] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[13] Sanjit Biswas and Robert Morris. Exor: opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 133–144. ACM, 2005.

[14] Wang Bo, Huang Chuanhe, Li Layuan, and Yang Wenzhong. Trust-based minimum cost opportunistic routing for ad hoc networks. *Journal of Systems and Software*, 84(12):2107–2122, 2011.

[15] Wang Bo, Huang Chuanhe, Li Layuan, and Yang Wenzhong. Trust-based minimum cost opportunistic routing for ad hoc networks. *Journal of Systems and Software*, 84(12):2107–2122, 2011.

[16] A. Boukerch, L. Xu, and K. EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30:2413–2427, 2007.

[17] A Boukerch, Li Xu, and Khalil El-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30(11):2413–2427, 2007.

[18] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *computer communications*, 28(10):1193–1203, 2005.

[19] Azzedine Boukerche. Performance comparison and analysis of ad hoc routing algorithms. In *Performance, Computing, and Communications, 2001. IEEE International Conference on.*, pages 171–178. IEEE, 2001.

[20] Azzedine Boukerche. A simulation based study of on-demand routing protocols for ad hoc wireless networks. In *Simulation Symposium, 2001. Proceedings. 34th Annual*, pages 85–92. IEEE, 2001.

[21] Azzedine Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mobile Networks and Applications*, 9(4):333–342, 2004.

[22] Azzedine Boukerche and Luciano Bononi. Simulation and modeling of wireless, mobile, and ad hoc networks. *Mobile ad hoc networking*, pages 373–409, 2004.

[23] Azzedine Boukerche, Xiuzhen Cheng, and Joseph Linus. Energy-aware data-centric routing in microsensor networks. In *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 42–49. ACM, 2003.

[24] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[25] Azzedine Boukerche and Amir Darehshoorzadeh. Opportunistic routing in wireless networks: Models, algorithms, and classifications. *ACM Computing Surveys (CSUR)*, 47(2):22, 2014.

[26] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Analysis of a randomized congestion control scheme with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995, 2001.

[27] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[28] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. A novel solution for achieving anonymity in wireless ad hoc networks. In *Proceedings of the 1st ACM*

international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pages 30–38. ACM, 2004.

[29] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 618–624. IEEE, 2004.

[30] Azzedine Boukerche and Xin Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, pages 5190–5194. IEEE, 2007.

[31] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. A distributed algorithm for dynamic channel allocation. *Mobile Networks and Applications*, 7(2):115–126, 2002.

[32] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. An efficient synchronization scheme of multimedia streams in wireless and mobile systems. *IEEE transactions on Parallel and Distributed Systems*, 13(9):911–923, 2002.

[33] Azzedine Boukerche, Kathia Regina Lemos Jucá, João Bosco Sobral, and Mirela Sechi Moretti Annoni Notare. An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing*, 30(5):629–646, 2004.

[34] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, volume 3, pages 5–pp. IEEE, 2005.

[35] Azzedine Boukerche, Renato B Machado, Kathia RL Jucá, João Bosco M Sobral, and Mirela SMA Notare. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, 30(13):2649–2660, 2007.

[36] Azzedine Boukerche, Anahit Martirosyan, and Richard Pazzi. An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. *Mobile Networks and Applications*, 13(6):614–626, 2008.

[37] Azzedine Boukerche and Sotiris Nikoletseas. Protocols for data propagation in wireless sensor networks. In *Wireless Communications Systems and Networks*, pages 23–51. Springer US, 2004.

[38] Azzedine Boukerche and Mirela Sechi M Annoni Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[39] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Localization systems for wireless sensor networks. *IEEE wireless Communications*, 14(6):6–12, 2007.

[40] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Secure localization algorithms for wireless sensor networks. *IEEE Communications Magazine*, 46(4):96–101, 2008.

[41] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[42] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo Freire Nakamura, and Antonio AF Loureiro. Dv-loc: a scalable localization protocol using voronoi diagrams for wireless sensor networks. *IEEE Wireless Communications*, 16(2):50–55, 2009.

[43] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, pages 560–567. IEEE, 2005.

[44] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 157–164. ACM, 2004.

[45] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. Fault-tolerant wireless sensor network routing protocols for the supervision of

context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4):586–599, 2006.

[46] Azzedine Boukerche and Yonglin Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer Communications*, 31(18):4343–4351, 2008.

[47] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4):387–399, 2009.

[48] Azzedine Boukerche, Yonglin Ren, and Richard Werner Nelem Pazzi. An adaptive computational trust model for mobile ad hoc networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 191–195. ACM, 2009.

[49] Azzedine Boukerche and Samer Samarah. A novel algorithm for mining association rules in wireless ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(7):865–877, 2008.

[50] Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad Z Ahmad, Ladislau Bölöni, and Damla Turgut. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13):3032–3080, 2011.

[51] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236. ACM, 2002.

[52] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi. *Trading structure for randomness in wireless opportunistic routing*, volume 37. ACM, 2007.

[53] Chen Chen, Jie Zhang, Robin Cohen, and Pin-Han Ho. A trust modeling framework for message propagation and evaluation in vanets. In *Information Technology Convergence and Services (ITCS), 2010 2nd International Conference on*, pages 1–8. IEEE, 2010.

[54] Tingting Chen and Sheng Zhong. An enforceable scheme for packet forwarding cooperation in network-coding wireless networks with opportunistic routing. *IEEE transactions on vehicular technology*, 63(9):4476–4491, 2014.

[55] Long Cheng, Jianwei Niu, Jiannong Cao, Sajal K Das, and Yu Gu. Qos aware geographic opportunistic routing in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 25(7):1864–1875, 2014.

[56] Jin-Hee Cho, Ananthram Swami, and Ray Chen. A survey on trust management for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 13(4):562–583, 2011.

[57] Jin-Hee Cho, Ananthram Swami, and Ray Chen. A survey on trust management for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 13(4):562–583, 2011.

[58] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel AF Mini, and Antonio AF Loureiro. Data communication in vanets: Protocols, applications and challenges. *Ad Hoc Networks*, 44:90–103, 2016.

[59] Amir Darehshoorzadeh, Mohammed Almulla, Azzedine Boukerche, and Sonny Chaiwala. On the number of candidates in opportunistic routing for multi-hop wireless networks. In *Proceedings of the 11th ACM international symposium on Mobility management and wireless access*, pages 9–16. ACM, 2013.

[60] Amir Darehshoorzadeh and Llorenç Cerda-Alabern. Distance progress based opportunistic routing for wireless mesh networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 179–184. IEEE, 2012.

[61] Amir Darehshoorzadeh, Robson De Grande, and Azzedine Boukerche. Towards a comprehensive model for performance analysis of opportunistic routing in wireless mesh networks. 2015.

[62] Horacio Antonio Braga Fernandes De Oliveira, Azzedine Boukerche, Eduardo Freire Nakamura, and Antonio Alfredo Ferreira Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *IEEE Transactions on Computers*, 58(5):677–691, 2009.

[63] Hongmei Deng, Wei Li, and Dharma P Agrawal. Routing security in wireless ad hoc networks. *Communications Magazine, IEEE*, 40(10):70–75, 2002.

[64] Henri Dubois-Ferrière, Matthias Grossglauser, and Martin Vetterli. Valuable detours: Least-cost anypath routing. *Networking, IEEE/ACM Transactions on*, 19(2):333–346, 2011.

[65] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 18–27. ACM, 2006.

[66] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Performance analysis of a distributed comparison-based self-diagnosis protocol for wireless ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 165–172. ACM, 2006.

[67] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, 2008.

[68] Kevin Fall and Kannan Varadhan. The ns manual (formerly ns notes and documentation). *The VINT project*, 47, 2005.

[69] Holger Füßler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad hoc networks. *Ad Hoc Networks*, 1(4):351–369, 2003.

[70] Saurabh Ganeriwal, Laura K Balzano, and Mani B Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):15, 2008.

[71] Euhanna Ghadimi, Olaf Landsiedel, Pablo Soldati, Simon Duquennoy, and Mikael Johansson. Opportunistic routing in low duty-cycle wireless sensor networks. *ACM Trans. Sen. Netw.*, 10(4):67:1–67:39, June 2014.

[72] Tirthankar Ghosh, Niki Pissinou, and Kami Makki. Towards designing a trusted routing solution in mobile ad hoc networks. *Mobile Networks and Applications*, 10(6):985–995, 2005.

[73] Jane K Hart and Kirk Martinez. Environmental sensor networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3):177–191, 2006.

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

[75] Dijiang Huang, Xiaoyan Hong, and Mario Gerla. Situation-aware trust architecture for vehicular networks. *Communications Magazine, IEEE*, 48(11):128–135, 2010.

[76] Vittorio P Illiano and Emil C Lupu. Detecting malicious data injections in wireless sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 48(2):24, 2015.

[77] Shiyu Ji, Tingting Chen, and Sheng Zhong. Wormhole attack detection algorithms in wireless network coding systems. *Mobile Computing, IEEE Transactions on*, 14(3):660–674, 2015.

[78] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks*, 1(2):293–315, 2003.

[79] John G Kemeny, James Laurie Snell, et al. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.

[80] Kashif Kifayat, Madjid Merabti, Qi Shi, and David Llewellyn-Jones. Security in wireless sensor networks. In *Handbook of Information and Communication Security*, pages 513–552. Springer, 2010.

[81] Rafael Laufer, Henri Dubois-Ferriere, and Leonard Kleinrock. Multirate anypath routing in wireless mesh networks. In *INFOCOM 2009, IEEE*, pages 37–45. IEEE, 2009.

[82] Rafael Laufer, Henri Dubois-Ferrière, and Leonard Kleinrock. Polynomial-time algorithms for multirate anypath routing in wireless multihop networks. *IEEE/ACM Transactions on Networking*, 20(3):742–755, 2012.

[83] Tan Le and Yong Liu. Opportunistic overlay multicast in wireless networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[84] Feng Li and Jie Wu. Mobility reduces uncertainty in manets. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1946–1954. IEEE, 2007.

[85] Jie Li, Ruidong Li, and Jien Kato. Future trust management framework for mobile ad hoc networks. *Communications Magazine, IEEE*, 46(4):108–114, 2008.

[86] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 120–130, New York, NY, USA, 2000. ACM.

[87] Na Li and Sajal K Das. A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks*, 11(4):1497–1509, 2013.

[88] Yanhua Li, Wei Chen, and Zhi-Li Zhang. Optimal forwarder list selection in opportunistic routing. In *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*, pages 670–675. IEEE, 2009.

[89] Yunfeng Lin, Baochun Li, and Ben Liang. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In *Network Protocols, 2008. ICNP 2008. IEEE International Conference on*, pages 13–22. IEEE, 2008.

[90] Haitao Liu, Baoxian Zhang, Hussein T Mouftah, Xiaojun Shen, and Jian Ma. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *Communications Magazine, IEEE*, 47(12):103–109, 2009.

[91] Chen Lyu, Dawu Gu, Xiaomei Zhang, Shifeng Sun, Yuanyuan Zhang, and Amit Pande. Sgor: Secure and scalable geographic opportunistic routing with received signal strength in wsns. *Computer Communications*, 59:37–51, 2015.

[92] Félix Gómez Mármol and Gregorio Martínez Pérez. Trip, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks. *Journal of Network and Computer Applications*, 35(3):934–941, 2012.

[93] Sergio Marti, Thomas J Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265. ACM, 2000.

[94] Anahit Martirosyan, Azzedine Boukerche, and Richard Pazzi. A taxonomy of cluster-based routing protocols for wireless sensor networks. In *2008 International Symposium on Parallel Architectures, Algorithms, and Networks (i-span 2008)*, pages 247–253. IEEE, 2008.

[95] Nick McKeown. Software-defined networking. *INFOCOM keynote talk*, 17(2):30–32, 2009.

[96] Pietro Michiardi and Refik Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Advanced Communications and Multimedia Security*, pages 107–121. Springer, 2002.

[97] Hoang Anh Nguyen and Silvia Giordano. Routing in opportunistic networks. *International Journal of Ambient Computing and Intelligence (IJACI)*, 1(3):19–38, 2009.

[98] Horacio ABF Oliveira, Azzedine Boukerche, Eduardo F Nakamura, and Antonio AF Loureiro. Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm. *Performance Evaluation*, 66(3):209–222, 2009.

[99] Richard WN Pazzi and Azzedine Boukerche. Mobile data collector strategy for delay-sensitive applications over wireless sensor networks. *Computer Communications*, 31(5):1028–1039, 2008.

[100] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, 44(11):134–141, 2006.

[101] Juan M Marín Pérez, Antonio Moragón Juan, Jaime Arrazola Pérez, Javier Monge Rabadán, and Antonio F Skarmeta Gómez. Security and privacy in vehicular communications with inter-trust. In *Cyber Security and Privacy*, pages 53–64. Springer, 2015.

[102] Niki Pissinou, Tirthankar Ghosh, and Kia Makki. Collaborative trust-based secure routing in multihop ad hoc networks. In *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, pages 1446–1451. Springer, 2004.

[103] Maxim Raya, Panos Papadimitratos, Virgil D Gligor, and Jean-Pierre Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[104] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *2008 IEEE International Conference on Communications*, pages 2129–2133. IEEE, 2008.

[105] Eric Rozner, Jayesh Seshadri, Yogita Mehta, and Lili Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. *Mobile Computing, IEEE Transactions on*, 8(12):1622–1635, 2009.

[106] Mahmood Salehi and Hamed Samavati. Dsr vs olsr: Simulation based comparison of ad hoc reactive and proactive algorithms under the effect of new routing attacks. In *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2012 6th International Conference on*, pages 100–105. IEEE, 2012.

[107] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology*, 60(2):656–663, 2011.

[108] Glenn Shafer et al. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[109] Nirav Shah and Dijiang Huang. A-weor: Communication privacy protection for wireless mesh networks using encoded opportunistic routing. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6. IEEE, 2010.

[110] Saeed Soltanali, Sajjad Pirahesh, Salman Niksefat, and Masoud Sabaei. An efficient scheme to motivate cooperation in mobile ad hoc networks. In *Networking and Services, 2007. ICNS. Third International Conference on*, pages 98–98. IEEE, 2007.

[111] Fan-Hsun Tseng, Li-Der Chou, and Han-Chieh Chao. A survey of black hole attacks in wireless mobile ad hoc networks. *Human-centric Computing and Information Sciences*, 1(1):1–16, 2011.

[112] Milica Pejanović Đurišić, Zhilbert Tafa, Goran Dimić, and Veljko Milutinović. A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199. IEEE, 2012.

[113] Luiz Filipe M Vieira. Performance and trade-offs of opportunistic routing in underwater networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2911–2915. IEEE, 2012.

[114] Leandro A Villas, Azzedine Boukerche, Horacio ABF De Oliveira, Regina B De Araujo, and Antonio AF Loureiro. A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks. *Ad Hoc Networks*, 12:69–85, 2014.

[115] Leandro Aparecido Villas, Azzedine Boukerche, Heitor Soares Ramos, Horacio AB Fernandes de Oliveira, Regina Borges de Araujo, and Antonio Alfredo Ferreira Loureiro. Drina: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *IEEE Transactions on Computers*, 62(4):676–689, 2013.

[116] Zehua Wang, Yuanzhu Chen, and Cheng Li. Corman: A novel cooperative opportunistic routing scheme in mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 30(2):289–296, 2012.

[117] Zhexiong Wei, Hongying Tang, F Richard Yu, Maoyu Wang, and Peter Mason. Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. *Vehicular Technology, IEEE Transactions on*, 63(9):4647–4658, 2014.

[118] Zhexiong Wei, F Richard Yu, and Azzedine Boukerche. Cooperative spectrum sensing with trust assistance for cognitive radio vehicular ad hoc networks. In *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, pages 27–33. ACM, 2015.

[119] Zhexiong Wei, Fei Richard Yu, and Azzedine Boukerche. Trust based security enhancements for vehicular ad hocnetworks. In *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications*, pages 103–109. ACM, 2014.

[120] Yang WenZhong, Huang ChuanHe, Wang Bo, Zhang ZhenYu, and Wang Tong. A reliable multicast for manets based on opportunistic routing and network coding. In *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, pages 540–545. IEEE, 2010.

[121] Bing Wu, Jianmin Chen, Jie Wu, and Mihaela Cardei. A survey of attacks and countermeasures in mobile ad hoc networks. In *Wireless Network Security*, pages 103–135. Springer, 2007.

[122] Fan Wu, Kai Gong, Tianrong Zhang, Guihai Chen, and Chunming Qiao. Como: a game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks. *IEEE Transactions on Wireless Communications*, 14(2):948–959, 2015.

[123] Fan Wu, Kai Gong, Tianrong Zhang, Guihai Chen, and Chunming Qiao. Como: a game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks. *IEEE Transactions on Wireless Communications*, 14(2):948–959, 2015.

[124] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. A survey on trust management for internet of things. *Journal of network and computer applications*, 42:120–134, 2014.

[125] Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. A security and trust framework for virtualized networks and software-defined networking. *Security and Communication Networks*, 2015.

[126] Zheng Yan, Peng Zhang, and Teemupekka Virtanen. Trust evaluation based security solution in ad hoc networks. In *Proceedings of the Seventh Nordic Workshop on Secure IT Systems*, volume 14, 2003.

[127] Shengbo Yang, Feng Zhong, Chai Kiat Yeo, Bu Sung Lee, and Jeff Boleng. Position based opportunistic routing for robust data delivery in manets. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE, 2009.

[128] Han Yu, Zhiqi Shen, Chunyan Miao, Cyril Leung, and Dusit Niyato. A survey of trust and reputation management systems in wireless communications. *Proceedings of the IEEE*, 98(10):1755–1772, 2010.

[129] Kai Zeng, Wenjing Lou, and Yanchao Zhang. Multi-rate geographic opportunistic routing in wireless ad hoc networks. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7. IEEE, 2007.

[130] Jie Zhang. A survey on trust management for vanets. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 105–112. IEEE, 2011.

[131] Kun Zhang, Rui Wang, and Depei Qian. Aim: An auction incentive mechanism in wireless networks with opportunistic routing. In *Computational Science and Engineering (CSE), 2010 IEEE 13th International Conference on*, pages 28–33. IEEE, 2010.

[132] Zhenxia Zhang, Azzedine Boukerche, and Richard Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, pages 19–26. ACM, 2011.

[133] Zhenxia Zhang, Richard W Pazzi, and Azzedine Boukerche. A mobility management scheme for wireless mesh networks based on a hybrid routing protocol. *Computer Networks*, 54(4):558–572, 2010.

[134] Zhongliang Zhao, Björn Mosler, and Torsten Braun. Performance evaluation of opportunistic routing protocols: A framework-based approach using omnet++. In *Proceedings of the 7th Latin American Networking Conference*, pages 28–35. ACM, 2012.

[135] Zifei Zhong, Junling Wang, Srihari Nelakuditi, and Guor-Huar Lu. On selection of candidates for opportunistic anypath forwarding. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(4):1–2, 2006.