A reliable trust-aware reinforcement learning based routing protocol for wireless medical sensor networks.

HAJAR, M.S.

2022

A Reliable Trust-Aware Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks

Muhammad Shadi Hajar

A thesis submitted in partial fulfilment of the requirements of the
School of Computing
Robert Gordon University
for the degree of
Doctor of Philosophy

December 2022

Abstract

Interest in the Wireless Medical Sensor Network (WMSN) is rapidly gaining attention thanks to recent advances in semiconductors and wireless communication. However, by virtue of the sensitive medical applications and the stringent resource constraints, there is a need to develop a routing protocol to fulfill WMSN requirements in terms of delivery reliability, attack resiliency, computational overhead, and energy efficiency. Therefore, this doctoral research aims to advance the state of the art in routing by proposing a lightweight, reliable routing protocol for WMSN.

Ensuring a reliable path between the source and the destination requires making trust-aware routing decisions to avoid untrustworthy paths. A lightweight and effective Trust Management System (TMS) has been developed to evaluate the trust relationship between the sensor nodes with a view to differentiating between trustworthy nodes and untrustworthy ones. Moreover, a resource-conservative Reinforcement Learning (RL) model has been proposed to reduce the computational overhead, along with two updating methods to speed up the algorithm convergence. The reward function is re-defined as a punishment, combining the proposed trust management system to defend against well-known dropping attacks. Furthermore, with a view to addressing the inborn overestimation problem in Q-learning-based routing protocols, we adopted double Q-learning to overcome the positive bias of using a single estimator. An energy model is integrated with the reward function to enhance the network lifetime and balance energy consumption across the network. The proposed energy model only uses local information to avoid the resource burdens and the security concerns of exchanging energy information.

Finally, a realistic trust management testbed has been developed to overcome the limitations of using numerical analysis to evaluate proposed trust management schemes, particularly in the context of WMSN. The proposed testbed has been developed as an additional module to the NS-3 simulator to fulfill usability, generalisability, flexibility, scalability, and high-performance requirements.

Keywords: Wireless Medical Sensor Networks (WMSN), Trust Management System (TMS), Reinforcement Learning (RL), Q-learning, Double Q-learning, NS-3, Blackhole attacks, Selective forwarding attacks, Sinkhole attacks, On-off attacks.

Acknowledgements

First and foremost, all praise and thanks are due to Allah Almighty for giving me the opportunity, strength, patience, and endurance to achieve this doctoral research.

I am deeply grateful to Robert Gordon University, School of Computing for their studentship as it would not have been possible to pursue my PhD without it. I am also thankful to all my supervisory team for their support and motivation throughout my PhD journey. I also would like to thank all the staff and colleagues at RGU for all the support and inspiration.

Special thanks must go to my principal supervisor and mentor Dr. Harsha Kalutarage for his invaluable supervision, insightful guidance, continuous support, and motivation. I would not have made it halfway without his persistent help.

Last but not least, my appreciation also goes out to my whole beloved family, my father, mother, wife, children, brothers, and sisters for their persistent support and encouragement. No words would ever express the true extent of my gratitude to my beloved family who shared this journey with me and bore all my stressful moments. I owe all my success to my beloved family.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Researchers are in no doubt that Wireless Medical Sensor Network (WMSN) will play an imperative role in our future [2]. Thanks to technological advancements, physicians can monitor their patients' physiological signs regardless of their geographical locations. These patient-friendly devices are not just mobile monitor devices but also telemedicine actuators. However, despite these advantages, security concerns and routing limitations are still hampering the widespread adoption of this promising technology. WMSN is prone to different types of threats that affect its operation and consequently endanger the patient's life.

Due to the ad hoc nature of the WMSN, cooperation between the network's nodes is essential to ensure the network's operation, as malicious or even selfish nodes may pose serious threats to service availability. WMSN is intolerant to packet-dropping attacks due to its real-time and sensitive information. Sensor Nodes (SNs) always have to choose the most reliable path to avoid dropping attacks. Moreover, the proposed routing candidate must be computational and energy efficient to fit the stringent resource constraints of WMSN.

1.1 Research Motivation

The percentage of people with chronic diseases is increasing dramatically, and it is expected to rise more over time as the population of the group of people aged over 60 is growing significantly [3]. This will increase the number of overloaded medical staff who are already working under pressure. Consequently, it increases the number of medical

errors and endangers patients' lives. Considering that caregivers have already been suffering from understaffed teams, the near future does not seem promising without adopting a revolutionized solution. WMSN has attracted much attention because of its various potential applications, in addition to the ability to relieve the burden of overloaded medical staff.

These tiny bio-sensor nodes with wireless communication capability are able to provide continuous monitoring data. For instance, a wrist-mounted pulse oximeter device is able to monitor the oxygen saturation of the blood. In contrast, a Continuous Glucose Monitor (CGM) device is able to send on-time readings of the levels of glucose in the blood. When it is necessary, the augmented insulin pump injects the required dosage of insulin into bloodstream.

WMSN will be part of our future life not just because of the diverse medical applications it provides, but also because it is able to enhance the efficiency of medical staff and reduce the long operational hours [4]. Unfortunately, however, WMSN is susceptible to different kinds of internal attacks that disrupt data delivery and may endanger the patient's life.

1.2 Challenges

WMSN is regarded as a branch of Wireless Sensor Network (WSN). Routing in WSN is still a challenging task despite the fact that abundant research is being put forward. The inherited WSN's security concerns, in addition to WMSN resource constraints and critical applications, make developing a routing protocol for WMSN more challenging for several reasons.

- Reliable delivery is a must in WMSN as packets carry sensitive data. Routing decisions are made based on calculating path cost using different parameters, such as hop counts and link bandwidth, to ensure a high delivery ratio. However, choosing the lowest cost path does not guarantee reliable data delivery as relaying nodes may drop the received packets for different reasons.
- Trust Management System (TMS) is introduced to deal with non-cooperative nodes within the network and enhance the overall security. However, this security countermeasure is vulnerable to on-off attacks, where a smart adversary can launch this attack leveraging his trust value.

- Evaluating novel proposals for Trust Management (TM) schemes remains a challenging task due to the lack of an adequate testbed. Researchers usually use numerical analysis to evaluate their proposed methods despite their limitations and inability to reflect realistic network scenarios.
- WMSN suffers from a resource scarcity, which places additional constraints on the deployment of any potential routing protocol; moreover, any protocol proposed for WSN may not necessarily fit WMSN.
- Reinforcement Learning (RL) algorithms, especially Q-learning, have been widely used in the literature to design routing candidates. However, the traditional RL model is resource-consuming and may not fit the resource constraints of WMSN.

1.3 Research Objectives

This thesis aims to enhance the overall security of WMSN by proposing a lightweight, efficient, and reliable routing protocol. To achieve this goal and address the aforementioned challenges, five objectives have been identified in this thesis as follows:

1. Review the WMSN security threats and identify the potential security countermeasures to protect the network from internal and external adversaries. This includes identifying the desired requirements for each security countermeasure to be met and highlighting the potential research areas and the promising enhancing directions.
2. Develop an effective, lightweight, and attack-resistant trust management scheme that is able to evaluate the behavior of nodes within the networks so as to differentiate between trustworthy and untrustworthy nodes.
3. Develop a novel RL model for routing applications to reduce the computational overhead of the traditional RL model. This method should be effective and able to reflect any change in the environment swiftly.
4. Develop an energy-efficient reliable routing protocol for WMSN. Enhancing the data delivery reliability of WMSN even in a hostile environment is essential because of the critical application of WMSN. SNs must always choose the most lowest cost reliable path to the destination.
5. Develop a trust management testbed to evaluate proposed trust schemes under different conditions. This testbed should be generic to evaluate the trust relationship

for different networks and protocols. Moreover, it should be easy to use and has a minimum resource footprint.

1.4 Contributions

The main contribution of this thesis is to expand the state of the art by proposing a reliable, efficient, and lightweight routing protocol for WMSN. The list below summarizes the key outcomes of this research:

- The first contribution of this thesis is the novel TMS presented in Chapter 3. It is a lightweight and attack-resistant TMS with two methods to evaluate the trust relationship between SNs. It provides a further line of defense against internal attacks.
- The second contribution is developing a novel RL model for routing applications to fit WMSN resource restrictions in Chapter 4. The proposed model, along with the proposed asynchronous updating method, are able to perform efficiently in routing applications and converge swiftly.
- The third contribution is proposing an energy-aware routing protocol based on double Q-learning for WMSN in Chapter 5. Q-learning algorithm has an inborn overestimation problem, which may affect the routing decisions negatively; hence, double Q-learning is used to fix this positive bias in the Q-learning algorithm.
- The fourth contribution is developing a generic trust testbed to evaluate proposed trust schemes for different networks and protocols. To bridge one of the critical gaps in this problem area, the testbed is developed as a module for the NS-3 simulator. The proposed module design meets several requirements, such as being lightweight, usable, and scalable.

1.4.1 List of Publications

Following is the list of publications produced during the work presented in this thesis:

- **Journals**
  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "3R: A Reliable Multi Agent Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks." IEEE Internet of Things Journal. Under

review.

  - Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kumara Kalutarage. "A survey on wireless body area networks: Architecture, security challenges and research opportunities." Computers & Security 104 (2021): 102211.

- **Conferences**

  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "RRP: A Reliable Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks." IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023.

  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "DQR: A Double Q Learning Multi Agent Routing Protocol for Wireless Medical Sensor Network." 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm). 2022.

  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "A Robust Exploration Strategy in Reinforcement Learning Based on Temporal Difference Error." 35th Australasian Joint Conference on Artificial Intelligence (AI). 2022.

  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "TrustMod: A Trust Management Module For NS-3 Simulator." 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2021.

  - Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kalutarage. "LTMS: A lightweight trust management system for wireless medical sensor networks." 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2020.

  - Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kalutarage. "ETAREE: An effective trend-aware reputation evaluation engine for wireless medical sensor networks." 2020 IEEE Conference on Communications and Network Security (CNS). IEEE, 2020.

- **Book Chapters**

  - Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "Security Challenges in Wireless Body Area Networks for Smart Healthcare." CRC Press, 2022.

- **Code Repository**
  - The author's GitHhub is available at https://github.com/mshsyr/

1.5 Thesis Structure

The remainder of the thesis is structured as follows.

Chapter 2 provides a background and literature review about WMSN that would be useful for readers who are not in the WMSN domain to understand the rest of this thesis. In this chapter, WMSN security threats and vulnerabilities have been investigated thoroughly. Moreover, potential security countermeasures have been discussed at different levels of defense, in addition to proposing a relevant taxonomy.

Chapter 3 proposes an effective and lightweight trust management system for WMSN. Two methods have been proposed to evaluate the trust relationship, one for in-body SNs, which suffer from tough resource constraints, and the other for on- and off-body SNs. The latter has been further improved by integrating an on-off attack detection module to enhance the overall protection.

Chapter 4 presents the proposed RL model for routing applications to reduce the computational overhead of the traditional RL model. Moreover, it presents the proposed method to produce a lightweight, reliable routing protocol using Q-learning.

Chapter 5 extends the work in Chapter 4 by proposing an energy-aware double Q-learning routing protocol. This chapter addresses the overestimation problem of Q-learning to ensure that the agents make unbiased routing decisions.

Chapter 6 presents the proposed trust testbed for trust evaluation, including a detailed description of the design requirements and implementation details. The proposed

simulation testbed has been developed as an additional module for the NS-3 simulator to allow researchers to use it under different conditions and for different networks.

Chapter 7 concludes this thesis. It summarizes the contributions and the key outcomes of this research. Moreover, future research directions are discussed in order to highlight potential improvements.

In this doctoral research, we developed an effective trust management system in Chapter 3, which has been used later in Chapters 4 and 5 to ensure reliable data delivery in WMSN. Furthermore, the same trust management system has been used to validate the developed trust testbed for NS-3 in Chapter 6.

Chapter 2

Research Background

In the era of communication technologies, wireless healthcare networks enable innovative applications to enhance the quality of patients' lives, provide helpful monitoring tools for caregivers, and allow timely intervention. However, due to the sensitive information within WMSNs, insecure data can violate the patients' privacy, while unreliable packet forwarding may disrupt the network operation and endangers the patient's life. Achieving a high level of security and packet delivery reliability involves various challenges due to its resource limitations and critical applications. This chapter provides the necessary background and literature review of WMSN technology, with a particular focus on security and privacy concerns along with their countermeasures, followed by highlighting research directions and open issues.

The main findings of this chapter were published in Computers & Security journal in 2021 [5].

2.1 Introduction

A WMSN is a wireless network that consists of a set of tiny bio-medical sensor nodes distributed on the body surface, underneath the skin, inside the body, or in the vicinity of the body. These extremely low-power sensor nodes have the ability to monitor the vital physiological signs of the body, and some nodes are also able to inject a medicine dosage directly into the body. WMSN, also known as Wireless Body Area Network (WBAN), Body Sensor Network (BSN), or a Medical Body Area Network (MBAN), has only one international standard released in 2012, which is IEEE 802.15.6 standard [1]. However,

this standard is vulnerable, as it will be discussed later, and thus many manufacturers use different protocols, such as IEEE 802.15.4 [6]. In the literature, there is some conflicting information regarding the characteristics of WMSN. For instance, some papers suggest that WMSN is scalable up to 256 nodes [7, 8], whereas the IEEE 802.15.6 standard defines it as a maximum of 64 nodes. Therefore, the information of the IEEE 802.15.6 standard is adopted to avoid such conflicts.

IEEE 802.15.6 provides a reliable, short range of communications and a wide range of data rates to fit different types of applications. Both wearable devices and Implantable Medical Devices (IDMs) are designed to send real-time readings of the body bio-signals to a remote server or a sink node. The monitored signals vary considerably depending on the node type and could be body temperature, blood pressure, respiration measurement, heart rate, blood glucose level, Electrocardiogram (ECG), or Electromyogram (EMG). Monitoring these physiological parameters of the elderly or those with chronic diseases provides more flexibility and freedom to patients and allows quick intervention when necessary. At present, the aging population is increasing dramatically across the globe [9]. For instance, the number of people aged over 85 in the United Kingdom is heading to double by mid-2041 [9]. Additionally, according to World Health Organization (WHO) [10], diabetes will be among the leading causes of death by 2030, where up to 15% of the overall national healthcare budget is dedicated to diabetes care. Consequently, the total expenditure on health systems and the percentage of overloaded medical staff are expected to increase significantly. This incentivizes taking advantage of the latest advancements of WMSN to enhance the quality of patients' lives, improve monitoring procedures, make timely intervention decisions, and reduce the overall cost of health systems and the long operational hours of medical staff.

Security and privacy concerns and limitations of routing protocols are the major challenges facing the widespread adoption of WMSN. Sensor nodes send very critical and sensitive data. Any compromise would not only violate the patients' privacy but may also endanger their lives. For instance, when an ECG sensor reading is dropped or a false one is provided to physicians, it may lead to incorrect interventions that could be harmful to patients. Similarly, when an automated insulin pump does not receive an injection command or receive a wrong or compromised command, it may inject an insulin overdose into the patient's bloodstream. In addition, WMSN is prone to various types of attacks ranging from internal to external and passive to active attacks. On the other hand, security countermeasures vary from traditional security solutions like authentication and encryption to Intrusion Detection Systems (IDSs) and Trust Management

Systems (TMSs) in order to address potential security vulnerabilities [5].

2.2 Wireless Medical Sensor Network

The importance of WMSN and its critical role in healthcare systems motivate the standardization process to enable the interoperability of different products from different vendors. IEEE 802.15.6 defines the Physical (PHY), and the Medium Access Control (MAC) layers [1]. The extremely rigorous requirements of WMSN transceivers, such as power efficiency, force IEEE Task Group 6 (TG6) to adopt three types of physical layers in order to satisfy different types of applications [11]. These physical layers could be summarized as follows [12]:

- Narrowband (NB) PHY: supports seven frequency bands with different data rates [1].
- Ultra-wideband (UWB) PHY: supports two different frequency bands, low and high, with a different number of channels while having the same bandwidth. The design of UWB PHY provides durable implementation with lower complexity and power consumption.
- Human body communication (HBC) PHY: supports one low-frequency band centered at 21 MHz, where the data transmission is conducted through the patient's body using Electric Field Communication (EFC) technology.

2.2.1 Design Characteristics

The overall design characteristics of the IEEE 802.15.6 standard are as follows [1, 8, 7]:

- Recoverable in case of any link or node failure.
- The ability to support a vast range of data rates starting from tens of Kbps and up to around 10 Mbps in order to meet all potential applications.
- Provides efficient power consumption mechanisms that allow the power source to last for several years.
- Provides reliable communication with acceptable jitter and latency values for both medical and non-medical applications.
- Supports the coexistence of both in-body and on-body sensor nodes.

- Able to support authentication, encryption, and integrity security mechanisms.
- Able to address node adding and removing within a relatively short time.
- Supports operation in a heterogeneous wireless environment.
- Complies with Specific Absorption Rate (SAR) regulations.
- Supports scalability up to 64 nodes.

These design requirements are taken into account in the proposed routing protocol in this thesis.

2.2.2 WMSN Topology

The IEEE 802.15.6 defines the network as a logical set consisting of sensor nodes and a single hub. It adopts the star topology with two different types of communications, simple one-hop and extended two-hop star topology. In simple one-hop star topology, nodes exchange frames directly with the hub. In contrast, in the extended two-hop topology, a relay node is introduced, and nodes are able to communicate directly with the hub or via a relay node, as illustrated in Fig. 2.1. The total number of nodes is specified by the MAC sublayer parameter mMaxBANSize, which has been set to 64. Sensor nodes (SNs) could be classified based on their role into:

- *Hub*: The hub node, the sink node, or the coordinator are different names for the same node type. The hub acts as a gateway to external networks. It controls the WMSN, and all the external communications go through it. It has better resources compared to normal nodes inside the network.
- *Relay node*: Some nodes have the relay capability to relay messages from end nodes to the hub. They are located in the hub's direct communication range.
- *End node*: Other nodes are considered end nodes. They are designed to perform specific tasks and exchange messages with the hub directly if they are in the direct communication range or via relay nodes if they are out of the direct communication range.

In addition to the aforementioned classification of nodes, different types of classifications are available in the literature, such as node deploying location (in-body or on-body, etc.) and node functionality (sensor nodes or actuator nodes, etc.) [13].

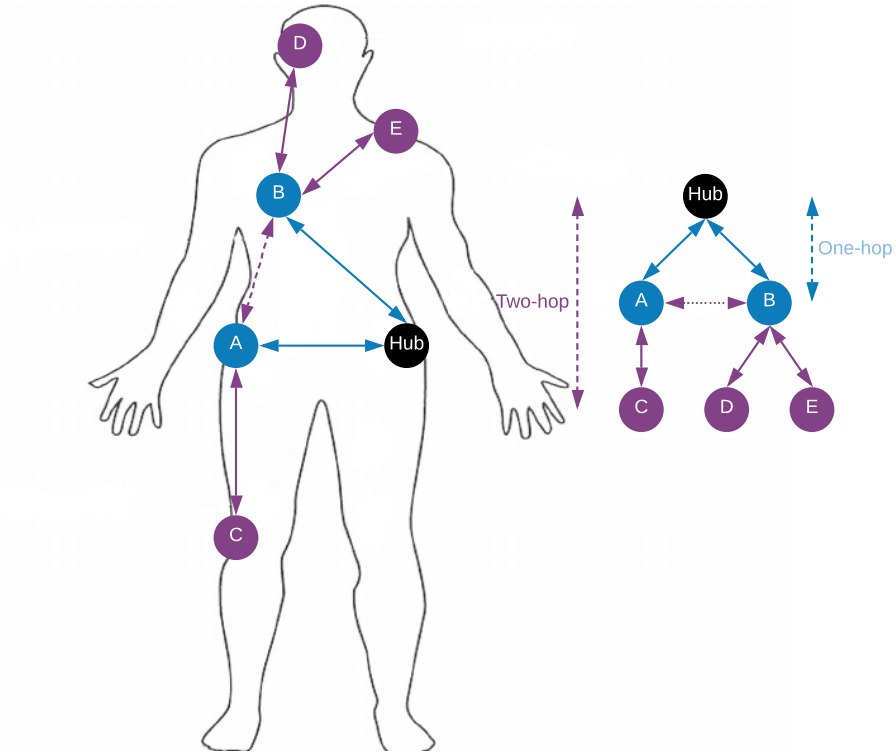


Figure 2.1: WMSN Topology

2.3 Security in WMSN

Security and privacy issues are critical concerns in all types of networks. However, WMSN, which processes critical data that, if compromised, may affect patients' health or endanger their lives, requires more effective security mechanisms to protect patients from all types of malicious activities. Although security in WMSN is crucial and has a high priority, there are still many open areas to research due to its strict resource restrictions, in addition to a wide range of security and privacy vulnerabilities inherited from WSN.

In order to ensure a high level of security and privacy, secure and reliable data delivery must be guaranteed. The basic security requirements could be outlined as follows:

- *Confidentiality*: Data must be protected from being disclosed to any unauthorized parties during data transmission as well as during the storage phase [14]. Data in WMSN contains very sensitive information about the health of the patient. It could be disclosed during transmission in an open channel by eavesdropping or could be disclosed when it is stored in a plain format when a node gets compromised.

- *Integrity*: When data is received, the receiver party has to ensure that the received information is original and has not been altered during the transmission phase [15]. Confidentiality measures cannot protect data from modification, which can be easily done by intercepting data in the transmission phase to inject, delete, or modify the sent message.

- *Availability*: Adversary can breach the availability and prevent authorized entities from accessing the required data [16]. Considering the critical applications of WMSN, disrupting the communication between the caregivers and sensor nodes may threaten the patient's life. Therefore, maintaining the ability to access the required data under any circumstances is a crucial requirement for this kind of applications.

- *Data Authentication*: While data integrity aims to save data from being modified during the transmission, data authentication aims to ensure that the received message came from the origin node, which is believed to be [17]. IEEE 802.15.6 defines the Message Authentication Code (MAC) to verify that the received message is sent by the original sender.

- *Data Freshness*: Adversary may intend to capture the transmitted messages and replay them afterward, which causes confusion and instability in WMSN [18]. Therefore, a mechanism to ensure that the received message is recent and that no adversary replays old messages is a must. Ensuring that the received messages are in order and on time is referred to as strong freshness, whereas there is no latency guarantee in weak freshness.

- *Secure Management*: Many security mechanisms such as encryption, decryption, and data authentication require keys, which must be distributed in a secure manner [19].

IEEE 802.15.6 defines three levels of security, where the hub and the sensor nodes can choose from. Each one of these security levels has different security characteristics as follows:

- *Level-0 Unsecured Communication*: No security measures are used at this level of security. Messages are exchanged in unsecured frames without confidentiality, authentication, integrity validation, or replay defense.

- *Level-1 Authentication*: Messages, at this security level, are exchanged in secured authenticated frames that ensure message authenticity, replay defense, and integrity validation. However, no measures are applied to provide confidentiality and privacy protection.

- *Level-2 Authentication and Encryption*: The highest level of security proposed in the standard. Messages are exchanged in secured, authenticated, and encrypted frames. Therefore, confidentiality, message authenticity, integrity, and replay defense are all provided at this security level.

Nodes and the hub are to choose the suitable security level during the association process based on their security requirements. Fig. 2.2 shows the security structure to generate security keys and provide security services. A preshared Master Key (MK) has to be activated or established between the hub and every node during the association process to achieve secured unicast communication. A Pairwise Temporal Key (PTK) is then created and shared between the two parties to be used per communication session. On the other hand, for secured multicast communication, a Group Temporal Key (GTK) is generated in the hub and subsequently shared with corresponding multicast group members by the hub.

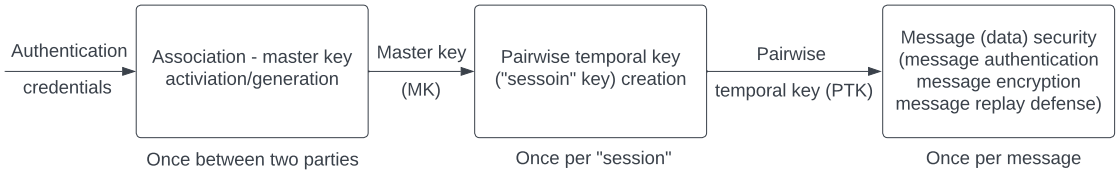


Figure 2.2: Security Structure

Although the IEEE 802.15.6 standard provides three security levels, recent research shows that the security mechanisms introduced in the standard are still vulnerable to different types of attacks. The author in [20, 21] analyses the key agreement protocols of the standard for MK establishment. He states that four protocols for key establishment defined in the standard are vulnerable to Key Compromise Impersonation (KCI) attacks and do not meet the forward secrecy requirement. Additionally, one of the protocols is vulnerable to offline dictionary attacks.

In this thesis, we assume that all secure communication requirements discussed in Section 2.6.1 are fulfilled and all data is exchanged in secure frames.

2.4 Routing in WMSN

WMSN is an infrastructure-less wireless network where cooperation between the network nodes is essential for network operation. The IEEE 802.15.6 standard defines only the physical and medium access control (MAC) layers. Thus, routing is still an open area of research. Furthermore, ensuring a lightweight, reliable data forwarding in WMSN is still challenging due to stringent resource requirements and the sensitive applications it provides. Therefore, WSN routing proposals do not necessarily fit WMSN. Generally speaking, routing in WMSN could be classified into non-learning routing protocols and learning-based routing protocols.

2.4.1 Non-learning routing protocols

Non-learning routing protocols involve using various methods to find the optimal path between the source and destination. Different classifications are proposed in the literature for this kind of routing methods [22, 23]. Generally speaking, those methods could be classified into posture-based, thermal-aware, and cluster-based routing.

Posture-based routing

The posture-based routing is built on the body movements regularity assumption to analyze the network topology [24]. If the SN is able to predict it is neighbors in a given time slot, efficient routing decision could be made to improve the data transmission rate and reduce the end-to-end latency [22]. In [25], the authors proposed a store-and-forward routing protocol for on-body SNs in WBAN. It is a distance vector routing protocol with a stochastic link cost. Although results showed an enhanced end-to-end delay, it is still a delay tolerant protocol that does not fit the critical application of WBAN. Moreover, the protocol is only proposed for on-body SNs. In [26], the authors proposed Network Management Cost Minimization for Dynamic Connectivity and Data Dissemination (NCMD) routing protocol to deal with high network dynamicity and reduce the network management cost. NCMD is proposed for on-body SNs to reduce the topology management overhead due to postural disconnections. However, the protocol complexity is high [22] and still regarded as a scope-specific protocol.

Thermal-aware routing

In thermal-aware routing, the nodes' temperatures are mainly used to evaluate paths with a view to reducing nodes' temperature by avoiding high-temperature nodes. Authors in [27] proposed TARA, one of the early thermal aware routing protocols for implanted SNs to balance temperature rise caused by relaying activities. Another example is Reliability Enhanced-Adaptive Threshold based Thermal-unaware Energy-efficient Multi-hop ProTocol (RE-ATTEMPT) [28] where the authors propose a single-hop and multi-hop routing protocol to reduce the delay and energy consumption. RE-ATTEMPT is designed to address the main shortcomings of ATTEMPT routing protocol [29], such as unbalanced energy consumption and the inability to avoid dead nodes from the routing path. Although there are many routing proposals adopted this approach [30, 31], the interest in this routing approach has decreased recently [22]. It is worth mentioning that both posture and thermal-based routing are regarded as scope-specific routing protocols.

Cluster-based routing

Cluster-based routing is another routing approach to tailor a routing protocol for WMSN. This method has been mainly proposed for WSN, where hundreds of nodes may exist in the network to reduce communication overhead. The Low Energy Adaptive Clustering Hierarchy (LEACH) [32] is the benchmark for this approach of routing with abundant

proposed variants that have been surveyed in [33]. In this routing approach, the network is divided into clusters of nodes. Each cluster elects a cluster head to integrate and forward the information. For instance, the authors in [34] proposed a Clustering based Routing Protocol for wireless Body Area network (CRPBA) to enhance the energy consumption rate and prolong the network lifetime. CRPBA uses two methods to forward frames to the sink, direct forwarding and cluster based forwarding depending on the distance from the sink node and the frame data type. However, the energy of cluster head, which are close to the sink, is depleted fast causing network disruption. Although cluster-based routing approach is widely investigated in WSN for large networks, it may not fit WMSN where the maximum number of SNs is set to 64 [1].

Other routing approaches

In addition to the aforementioned approaches, there are some routing methods that do not fall within the previous categorization. For instance, in [35], the authors proposed independent multi-path routing protocol for WBAN. Another example is mobile sink routing protocols [22]. However, these approaches still do not meet the tough resource constraints and do not ensure reliable data delivery.

2.4.2 Learning-based routing protocols

Machine learning (ML) is the science that enables computer systems to learn and adapt on their own. It has been widely used in every aspect of our life. As routing is regarded as an optimization problem where the agent always try to find the optimal path, much research has been put forward to optimize routing decisions using machine learning algorithms [36, 37, 38, 39, 40]. There are three main types of learning in ML, supervised learning, unsupervised learning and reinforcement learning. In supervised learning, the machine learning model is trained using labelled dataset while in unsupervised learning, unlabelled dataset is used to identify hidden patterns. On the other hand, in reinforcement learning, no dataset is required for training. The learning agent interacts with the environment to maximize the long term rewards. As in routing applications there is no use of training data, Reinforcement Learning (RL) methods, especially Q-learning, are widely used to learn routing paths in WSN [36, 37, 41, 42]; however, a few has targeted WMSN [43, 44, 45]. The reason could be attributed to the computational overhead incurred when adopting the traditional RL model to learn the network environment, which will be discussed further in Chapter 4 when proposing our RL model for routing applications. To the best of our knowledge, all RL-based routing protocols used the same RL model

which is a resource consuming process as it calls for updating the action value table for each sent or forwarded packet. Different kinds of metrics have been used to propose an efficient routing protocol, such as hop counts [46], delay [47] and energy [48]. However, this kind of metrics does not necessarily guarantee reliable delivery, which is a critical requirement for WMSN applications. Therefore, there is a need to develop a routing protocol that ensure high delivery ratio with minimum resource footprint and energy consumption, which have been discussed further in Chapters 4 and 5.

2.5 WMSN Threats

Considering WMSN as a branch of WSN, it inherits all security threats from WSN. Moreover, the special type of applications that WMSN provides and the stringent resource constraints increase the threats and security concerns to WMSN deployment. The security concerns in WMSN could be divided into attacks and misbehavior activities. While attacks involve any malicious activities that intend to damage the network or affect its operation, misbehavior threats occur when an internal node acts in an improper way that affects the operation or the network's performance. Fig. 2.3 shows the WMSN threats taxonomy discussed throughout this chapter.

2.5.1 WMSN Attacks

WMSN is susceptible to different types of attacks, which could be classified based on the origin of the attack into internal and external attacks as follows [49],

- *Internal attacks:* This kind of attack is sourced from inside the network by a malicious or compromised node, such selective forwarding attacks. The main challenge in this type of attack is that traditional security measures, which are considered the first line of defense, are not able to protect the network from this type of attack.
- *External attacks:* This kind of attack is launched by outsiders, which may involve external nodes or any other types of adversaries, such as eavesdropping attack.

Another method proposes classifying attacks based on the nature of the attack into passive and active attacks as follows [49],

- *Passive attacks:* The main aim of passive attacks is to gather data rather than threaten the network, such as eavesdropping attacks. This type of attack violates the confidentiality and privacy requirements. Furthermore, adversaries could utilize

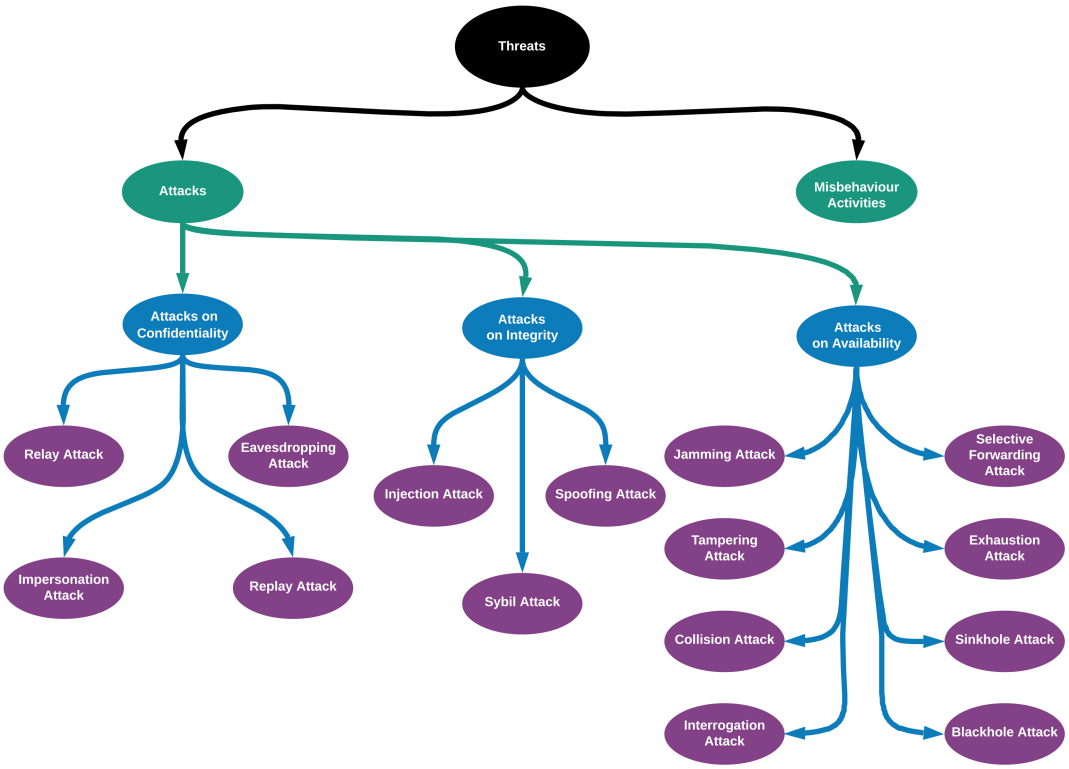


Figure 2.3: WMSN Threats Taxonomy

the gathered information to launch active attacks later.

- *Active attacks*: This kind of attack contains a various range of malicious activities such as data alteration and route poisoning. DoS attacks, for instance, target the operation of the network in order to degrade the performance of the network and deplete the resources as well.

To provide a logical sequence across this chapter, we classify the attacks based on the CIA (Confidentiality, Integrity, Availability) security requirements model. Note that the below list of attacks is not exhaustive. It covers the widely-known types of attacks.

Attacks on Confidentiality

The attacks on confidentiality are listed as follows:

- *Eavesdropping Attack:* The eavesdropping attack is a monitoring attack where the adversary snoops on the medium to capture the transmitted frames in order to extract sensitive information in the absence of a victim's awareness [50]. In security level 0 (unsecured communication) and security level 1 (authentication but not encryption), where no confidentiality or privacy protection is available [1], the adversary can easily gather sensitive and private information. On the other hand, in security level 2, where authentication and encryption are provided, the adversary still has the chance to eavesdrop and get the secret keys during the key exchange phase.

- *Replay Attack:* Attackers may capture and store messages and then replay them into the network [51]. This may lead to confusion and in some circumstances, lead to a significant problem when an action is taken based on these replayed messages because the late received messages are still valid ones. IEEE 802.15.6 provides a replay defense mechanism to detect replay attacks in both security levels 1 and 2, where the first octet of the MAC frame body, which is the "Low-Order Security Sequence Number", is used for data freshness and replay detection [1].

- *Relay Attack:* It is a sort of the Man-in-the-middle attack where the malicious node intercepts the communication between two nodes [18]. The two victimized nodes believe they are in direct communication; however, the malicious node can intercept all exchanged frames.

- *Impersonation Attack:* The attacker can take advantage of the eavesdropped messages to impersonate a legitimate node in order to receive more private information, which causes more harmful effects [52].

Attacks on Integrity

All integrity attacks are active attacks where the adversary injects or manipulates transmitted messages over the network.

- *Spoofing Attack:* There are different types of spoofing attacks where adversaries in all of them aim to alter or modify messages to get legitimate access or even to interrupt the operation of the network [51]. The communication in security levels 1 and 2 of IEEE 802.15.6 transfers authenticated messages between nodes by setting the MIC field of the Media Access Control (MAC) frame body to the Message Authentication Code (MAC) in order to ensure the integrity and authenticity of the received messages [1].

- *Modification/Injection Attack:* It is a sort of Man-In-The-Middle (MITM) attack where the adversary not just relays messages between the victims but also injects new messages or modifies the exchanged ones [50]. What makes it worse is that the exchanged messages may convey critical or urgent biometric signals. For example, the message may reflect an emergency case or contain a command from the healthcare center to a respective node to increase the insulin dosage that has to be injected into the patient's body. In both cases, this attack may affect the patient's life.

- *Sybil Attack:* In the Sybil attack, the adversary impersonates fake identities illegitimately. He can use the fake identity to attack the network until he is got detected and then generates a new identity and continues his malicious activities [53].

Attacks on Service Availability

The main attack against service availability is the Denial of Service (DoS) attacks. DoS attacks can be performed at any stack layer. Below are the most common DoS attacks:

- *Jamming Attack:* Since it has been explored in the literature in 1982 [54] till this date, transmitting data wirelessly is always liable to jamming attacks. The attacker intentionally interferes with the frequencies used by other nodes in the networks;

consequently, the Signal-to-Interference-plus-Noise ratio (SINR) decreases significantly. As the jamming attack is very detrimental, WMSN has to address this serious threat in order to save computational and energy resources.

- *Tampering Attack:* A tampering attack is a physical attack where the adversary is able to access the node physically and cause damage to the hardware components of the node or acquire critical information like cryptographic keys [51]. Although nodes in WMSN are either implanted underneath the skin or in direct touch with the human body, the patient still has to be aware of who is authorized to handle the nodes physically to defeat this type of attack.

- *Collision Attack:* The collision occurs when two or more nodes transmit at the same time [55]. This overlapped transmission degrades network performance and depletes the nodes' energy. Adversary intentionally overlaps other nodes' transmission to create a collision. This signal interference leads to receiving a collided frame, and consequently, Cyclic Redundancy Check (CRC) fails to verify the received frame. Hence, it will be discarded. The performance degrades dramatically by continuing the adversary to collide with the transmitted frames. For example, colliding with acknowledgment frames like I-Ack or B-Ack may double the Contention Window (CW) up to reach CWmax[UP] [1].

- *Interrogation Attack:* In this kind of attack, the adversary or the compromised node takes advantage of the RTS/CTS (Request To Send/ Clear To Send) mechanism, which is usually used with the CSMA/CA protocol to overcome the hidden terminal problem [56]. The attacker frequently sends an RTS message in order to obtain a CTS response from the targeted nodes, and consequently, all nodes abstain from using the network [57, 58, 59]. To the best of our knowledge, RTS/CTS mechanism is not used in IEEE 802.15.4, nor IEEE 802.15.6 [1, 6]; however, authors in [60, 61] evaluate RTS/CTS mechanism in conjunction with the CSMA/CA protocol for both IEEE 802.15.4 and IEEE 802.15.6, respectively. The simulation results show a tangible enhancement in the overall performance. Therefore, the interrogation attack still has to be considered.

- *Exhaustion Attack:* It is a kind of attack in which the adversary attempts to deplete the victim's resources, such as the denial of sleep attack where the adversary depletes the battery of the victim [62].

2.5.2 Misbehavior Activities

The second type of WMSN threats is misbehaving activities, which are unexpected behavior of an internal node. It is usually a selfish activity to gain extra resources unfairly or to save power. For instance, a relay node could launch a packet-dropping attack to disrupt the network operation with a view to saving power. Packet-dropping attacks have different patterns. Below is a list of the most common dropping attacks:

- *Selective Forwarding Attack:* Selective forwarding (or greyhole) attack is a well-known attack in WSN where a compromised node drops some packets or selectively forwards some of them [63]. For instance, forwarding packets to a specific destination and dropping others. According to the WMSN topology, the sink node is not necessarily in a direct communication range with all nodes [1]; hence relay nodes' cooperation is mandatory to ensure WMSN operation [64].

- *Blackhole Attack:* This kind of internal attack is a particular type of selective forwarding attack, where malicious nodes drop all incoming frames instead of forwarding them [65].

- *Sinkhole Attack:* The malicious or the compromised node attempts to attract all the traffic inside the WMSN and then drop it [66]. The adversary can run this kind of attack by sending fake routing updates showing itself as the shortest path to the medical server.

- *On-off Attack:* One of the smart packet dropping attacks where the compromised node alternately changes its behavior between benign and malicious manners in order to keep itself undetected [67]. Hence, a malicious node is regarded as a trusted one while it continues to disrupt the network operation.

Misbehaving is very harmful because traditional security mechanisms usually fail to detect or prevent this kind of internal threat. TMS, discussed later in Section 2.6.3, is an effective measure to assess nodes' behavior to identify untrustworthy and misbehaving nodes.

2.6 Security Countermeasures

Security and privacy in WMSN are essential due to the detrimental effects of its vulnerabilities. However, achieving a high level of security and privacy is challenging. This is because of the specific nature of WMSN, notably, the resource limitations and the vital

applications it provides. In this section, we comprehensively investigate the available security countermeasures and highlight their vital requirements to ensure that our routing proposal is deployed in a secure WMSN ecosystem. Moreover, this allows identifying available research gaps, which have been reported in Section 2.7. In order to ensure an end-to-end high level of security, three different types of security countermeasures need to be considered. It is worth mentioning that due to the scarcity of WMSN-specific schemes in the literature, some security schemes presented in this section are generally proposed for WSN, which have a high potential of deployment on WMSN; however, these potentials are worthy of further investigations before direct adoption. Fig. 2.4 shows a high-level perspective of the security countermeasures discussed throughout this chapter.

2.6.1 Secure Communication

More attention needs to be paid to security and privacy in military and medical applications due to the dangerous consequences of their vulnerabilities. This section states the security requirements related to WMSN communications. This is followed by a review of the security countermeasures, which are authentication and key establishment, integrity validation, and encryption.

Secure Communication Requirements

In order to defend against well-known attacks and achieve a high level of security, the following security requirements have to be met in any proposed scheme:

- *Lightweight:* Any proposed security solution must be computationally lightweight to fulfill the constraints of resource limitations [68].
- *Anonymity:* It ensures that no outsider will be able to know the identity of the two parties during the authentication process, which enhances the privacy [69].
- *Mutual authentication:* It means that the involved parties are able to authenticate each other; therefore, the authentication process is secured from any impersonation attack [70].
- *Unlinkability:* It ensures that the hidden identity of the nodes is still maintained even if the adversary could capture two transmitted messages belonging to the same node because it is impossible to link or associate the captured messages to find out the identity of the sender [71].

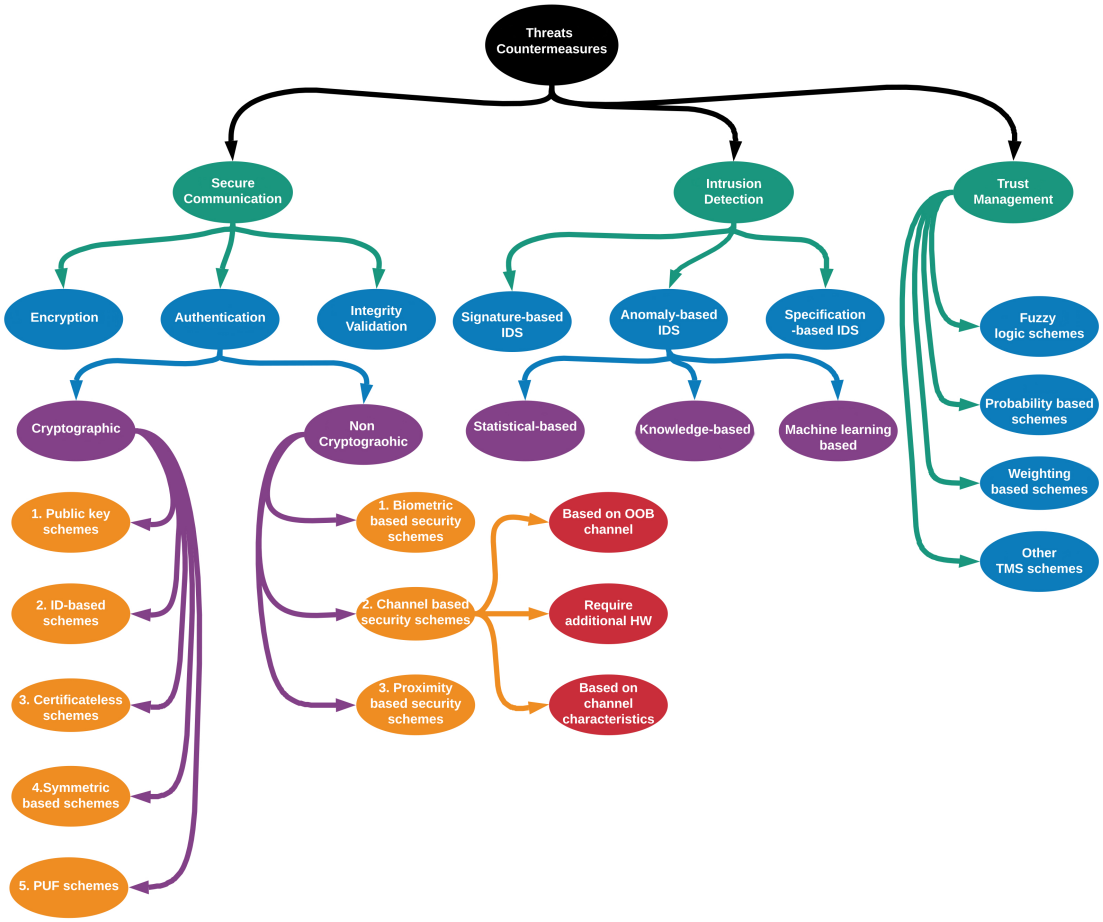


Figure 2.4: WMSN Security Countermeasures Taxonomy

- *Session key establishment:* After a successful authentication process between two nodes, a secure session key should be generated and exchanged securely in order to secure the subsequent communications [70].
- *Forward secrecy:* It ensures that the session key is still secured even if one or the two communicating parties are compromised; moreover, even when the adversary has one or both private keys [72].
- *Revocability:* It is the ability to revoke any misbehaved node effectively [70].
- *Non-repudiation:* It is imperative that the transmitted messages in WMSN are not repudiated; therefore, the sender cannot deny his sent messages [69].
- *Resilience to well-known attacks:* Any proposed security solution must be immune to well-known attacks. For instance, resilience to replay attack, resilience to impersonation attack, resilience to modification attack, and resilience to man-in-the-middle attack [17].
- *Key escrow resilience:* Generating keys by Private Key Generator (PKG) raises a key escrow problem in both Public Key Infrastructure (PKI) and ID-based infrastructure as well [73].

Authentication and Key Establishment

Authentication and key establishment form one of the essential foundations of securing the communication of WMSN. Much research has been proposed in the literature to address this issue. However, many of the proposed schemes fail to meet the aforementioned security requirements or are vulnerable to certain attacks [74], [75]. Therefore, authentication and key establishment are still an open area for research in WMSN [71]. Authentication gives the receiver the ability to verify the origin of the received messages in order to identify any malicious sender claiming legitimacy. These authentication and key establishment schemes could be classified as follows:

Non-cryptographic security schemes All non-cryptographic security schemes are based on assumptions related to the physical characteristics of the network. It is worth mentioning that this type of security scheme is classified as non-cryptographic due to the used technique to achieve authentication; however, this does not mean that they are unable to generate keys. For instance, some biometric security schemes are able

to generate keys, which could be used later to encrypt the exchanged messages. Non-cryptographic security schemes could be classified as follows:

1. *Biometric based security schemes:* Much research in the literature uses unique data from the body domain to achieve lightweight authentication. This trend of research emerges because of the thought that it is difficult to forge physiological body signs. The authors in [76] propose a novel approach for key generation between neighbor nodes based on electrocardiogram (ECG) signals. Although the proposed ECG-based key generation is able to secure intra-BAN communication without key establishment costs, it has some drawbacks. For instance, it is applicable just in case of all the nodes are located on the surface of the human body, while in WMSN, nodes could be implanted underneath the skin or in the vicinity of the body. Therefore, biometric-based security schemes could be considered scope-specific schemes. Moreover, most of the physiologically based security schemes proposed in the literature do not meet the aforementioned requirements. Neither ESKAS [77] nor OPFKA [78] meets the unlinkability and mutual authentication requirements. Furthermore, not all the nodes have the ability or the hardware to collect the required authentication data. The authors in [79] proposed a symmetric security scheme to generate and distribute the cryptographic keys using the ECG signal for WMSN. WMSN nodes should be able to sense the ECG signal using a time synchronization method in order to generate the security key. Informal and formal security analyses are used to prove the robustness of the proposed method.

2. *Channel based security schemes:* It is another security scheme trend, which takes advantage of the physical characteristics to authenticate nodes, and in some cases, to extract keys. It is built on the assumption that channel characteristics between two nodes are the same. Channel-based security schemes could be categorized into three types based on the use of physical characteristics:

    - Security schemes based on an out-of-band (OOB) communication channel: Some research proposes using a new auxiliary channel to facilitate node authentication by assuming the resilience of the new out-of-band channel to eavesdropping attacks. The proposed OOB channel varies from audio to ultrasound channels. For instance, authors in [80] introduce a way of authentication using visual OOB and with the patient's help. By comparing the blinking LED patterns, the user is able to accept or deny the authentication

process. Although the proposed security scheme in [80] meets mutual authentication, forward secrecy, and revocation requirements, it is still regarded as a scope-specific scheme as some nodes could be implanted inside the body of the patient. Moreover, the modular exponentiation operations used in the proposed scheme do not fit the strict resource limitations of WMSN.

- Security schemes that require additional hardware: One of the channel-based schemes that requires using additional hardware, which is not recommended from the WMSN perspective due to the tough resource limitations. For instance, Good Neighbor [81] is a proposed solution for pairing two devices using multiple antennas on the receiver side.

- Security schemes based on channel characteristics measurements: the authors in [82, 83] propose a lightweight authentication scheme based merely on Received Signal Strength (RSS) measurements. Because of the wireless channel correlation, a condition arises for node placement, which must be distributed in the half-wavelength range. BANA [83] solely provides an authentication mechanism for on-body devices, which are on the Line of Sight (LOS) and does not extract any keys, whereas MASK-BAN [82], which considers the channel variations of heterogeneous nodes and their reflection on RSS, provides authentication and keys extraction mechanism as well.

As most of the proposed security solutions are for on-body nodes, channel-based security schemes are scope specific and may not apply to in-body nodes. Furthermore, many physical parameters could affect RSS, such as the surrounding environment, nodes' positions, and mobility.

3. *Proximity-based security schemes:* By exploiting the small-scale fading variations on wireless channels when two wireless devices are close to each other and with the help of a third-party RF source, the authors in [84] introduce their proximity-based scheme to extract the secret key. However, the central dilemma in such schemes is that the two involved parties must reside inside the half-wavelength in order to have the same small-scale fading variation, and it is recommended to be 0.1λ or less while the adversary should reside on 0.4λ or more. It is worth mentioning that [84] is a generic security scheme that could be used for sensor networks as well.

Cryptographic security schemes Cryptographic security schemes vary depending on the types of keys and could be classified into the following categories.

1. *Public key signature schemes:* Public key cryptography (PKC) is based on generating two non-identical keys. One key is used for encryption or signature generation, whereas the other is used for decryption or signature verification. There is a clear consensus that conventional PKC is not applicable to be used in wireless healthcare applications because of the restricted resources. The main idea of PKC is completely based on two mathematical problems. The first one is a straightforward mathematical problem to generate the public and private keys, while the second one is the reverse operation to calculate one key knowing the other one, which must be extremely hard. However, the extreme hardness of this mathematical operation still needs to be proved [85]. The first mathematical problem could be either an integer factorization problem like in (Rivest, Shamir, and Adelman) RSA or a discrete logarithmic problem like in Elliptic Curve Cryptography (ECC). However, both of them are impractical for WMSN applications because they are voracious in using minimal resources. Thus, some research focuses on improving the performance of the inherited algorithms, such as authors in [86] where a hybrid multiplication method is used in order to limit the number of memory access, which speeds up the process by around seven times. On the other hand, the traditional cryptographic algorithm RSA still uses a longer key 1024-bit than ECC 160-bit for the same security level, which is still memory voracious. Therefore, ECC-based PKC seems more interesting than RSA. A pre-configurable library based on ECC-PKC for wireless sensor networks has been proposed in [87].

2. *ID-based signature schemes:* it is a public-key cryptography that was first proposed in 1984 [88]. In Identity-based Public Key Cryptography (ID-PKC), the node's public key is built from a combination of identity information like a network address, whereas the private key is generated by a Trusted Third Party (TTP) named Private Key Generator (PKG), hence there is no need for Certificate Authority (CA). Since then, many ID-based signature schemes proposed in the literature [89, 90, 91]. However, all these solutions are designed for client-server infrastructure, so it cannot meet all the security requirements of WMSN. Moreover, in addition to their vulnerability to some well-known attacks, all ID-based signature schemes face the same key escrow problem because of the TTP entity. Problems of having one PKG could be summarized into: First, the PKG is able to decrypt any transmitted message. Second, because of having all private keys, the PKG is able to forge any node's signature.

3. *Certificateless signature schemes:* A new public-key cryptography that lies between

PKC where the heavy computational overhead of verifying certificates with CA and ID-PKC schemes, which suffer from key escrow problem. This new concept was first proposed in the literature by authors in [92]. Although Certificateless Public Key Cryptography (CL-PKC) solves the inborn key escrow problem in ID-PKC because of using PKG as a TTP, it still uses a TTP entity. However, this TTP entity, named Key Generator Center (KGC) in CL-PKC, does not hold any nodes' private keys but a master key instead of it. The KGC is just responsible for providing the partial private key (D_A), which is sent to the other entities to produce their own keys [92]. In addition to having a partial private key, KGC also generates and holds a master key. Afterward, many certificateless authentication schemes have been proposed in the literature [93, 52, 70, 94, 95]; however, most of which focus just on a remote authentication, which is the authentication between the coordinator node and the Application Providers (APs). Although some schemes like in [52] propose a multi-layer authentication scheme, it is still considered a certificateless signature scheme just beyond the intra-BAN domain. Moreover, the author in [96] finds out that the proposed scheme in [70] is vulnerable to impersonation attack; therefore, neither mutual authentication nor non-repudiation requirements are satisfied.

4. *Symmetric based schemes:* In this kind of authentication scheme, a pre-shared master key, in addition to a unique identifier for each node are used to achieve mutual authentication as well as key establishment between two nodes like in [71]. Although the authors claim that their proposed algorithm fulfills the unlinkability and forward secrecy requirements, it is clear that it is not, as the adversary can easily calculate the value (γ), which is unmaskable in the proposed scheme. The authors in [97] suggest a modification to [71] in order to fulfill the forward secrecy and unlinkability requirements. However, there is still a key escrow problem where the coordinator node saves the master key in addition to all nodes' identities.

5. *Physical Unclonable Function schemes:* Physical Unclonable Function (PUF) is the fingerprint of the node's hardware. This property occurs because of the unavoidable manufacturing difference between nodes. This uniqueness and randomness hardware feature is very attractive to researchers to build and design security solutions based on it. The authors in [98] take advantage of the Integrated Circuits (ICs) variant delay characteristics to introduce PUF to be used for authentication and key generation. In [99] a mutual authentication mechanism between any two WMSN sensor nodes using PUF is introduced; however, the authentication process cannot be achieved without the help of the coordinator, which according to

the authors' assumption, cannot be compromised. The authors in [100] present an authentication and key establishment scheme, not only between sensor nodes and the coordinator but also between any two nodes with the assistance of the coordinator. Therefore, sensor nodes are afterward able to communicate directly with each other. This proposed shared secret establishment aims at authenticity verification without considering confidentiality. A simplified authentication solution has also been proposed in [101], which is resilient to some known attacks, such as impersonation attacks, replay attacks, and tampering attacks. However, some security requirements like anonymity, mutual authentication, and unlinkability are not considered. Another PUF-based scheme is proposed in [102] for the multi-hop body area network. It is a hierarchical authentication scheme to allow nodes that are not in the direct communication range of the sink to authenticate themselves in an efficient way. Moreover, a cloud TTP has been used in this scheme to store the Challenge-Response Pairs (CRPs) with a view to reducing the storage overhead.

The surveyed authentication and key establishment schemes are summarized in Table 2.1, stating the solution approach, the stated requirements it fulfills, and the potential improvements to fulfill additional requirements.

Integrity Validation

Message integrity is the process of verifying that the received message is intact and is exactly as sent. This ensures that the transmitted message from the sender to the receiver is not altered by any type of manipulation such as changing content, adding fragments, removing fragments, or content transposition. This also includes any type of transmission error. A secret key is required to ensure the message authenticity; however, confidentiality by itself is not enough to protect data from being modified by an adversary during the transmission phase. For instance, an external adversary may intercept and modify the message before re-transmitting it to the receiver. Although applying a message integrity scheme can be a simple task to implement when coupled with a durable cryptographic scheme [103], applying an efficient message integrity mechanism in WMSN can still be a challenging requirement due to the unique characteristics of WMSN [104].

IEEE 802.15.6 provides a mechanism to verify the message authenticity in both security levels one and two [1]. The Media Access Control (MAC) frame body is formatted as illustrated in Fig. 2.5. The MAC frame body length is variable and can scale up to a maximum value defined in the parameter "pMaxFrameBodyLength", which is set in the

Table 2.1: Authentication and Key Establishment Schemes

Lit.	Title	Year	Solution	Stated Fulfilled Requirements	Potential Improvements	Comments
[94]	Certificateless pairing-free authentication scheme for wireless body area network in healthcare management system	2020	Certificateless signature schemes	Mutual authentication, Forward secrecy, Anonymity, Session key establishment, Key escrow resilience, Unlinkability	Security analysis against more attacks	Remote certificateless authentication scheme.
[52]	A lightweight multi-layer authentication protocol for wireless body area networks	2018	Certificateless signature scheme	Lightweight, Non-reputation, Mutual authentication, Session key establishment, Key escrow resilience, Forward secrecy	Anonymity	A Certificateless scheme in the section between the PDA and AP.
[70]	Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks	2015	Certificateless signature scheme	Non-repudiation, Anonymity, Key escrow resistance, Revocability, Mutual authentication, Forward Secrecy	Non-repudiation, Mutual authentication [96]	Remote certificateless authentication scheme only beyond the intra-BAN. The authors in [96] report that this scheme is vulnerable to type-I adversary; therefore, it does not meet the non-repudiation and mutual authentication requirements.
[71]	Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks	2017	Symmetric based schemes	Anonymity, Mutual authentication, Session key establishment, Lightweight, Unlinkability, Forward secrecy	Unlinkability, Forward secrecy [97]	Although authors state that their scheme meet the unlinkability and forward secrecy requirements, it is not [97]. Security is evaluated using AVISPA and compared to other schemes.
[76]	ECG-cryptography and authentication in body area networks	2012	Biometric based security scheme	Lightweight, Session key establishment, Mutual authentication	Anonymity, Unlinkability, Forward secrecy, Non-repudiation, Revocability	A scope specific solution. It requires all nodes to have the same physiological sign sensor.
[77]	An efficient and secure key agreement scheme using physiological signals in body area networks	2012	Biometric based security scheme	Session key establishment, Mutual authentication	Lightweight, Anonymity, Unlinkability, Forward secrecy, Revocability	A scope specific solution. It requires all nodes to have the same physiological sign sensor.
[78]	OPFKA: Secure and efficient Ordered-Physiological-Feature-based key agreement for wireless Body Area Networks	2013	Biometric based security scheme	Lightweight, Resistance against brute force attacks, Session key establishment, Mutual authentication	Anonymity, Unlinkability, Forward secrecy, Revocability	A scope specific solution. It requires all nodes to have the same physiological sign sensor.
[80]	Secure Ad-Hoc Trust Initialization and Key Management in Wireless Body Area Networks	2013	Channel based security scheme	Lightweight, Revocation, Mutual authentication, Forward secrecy	Lightweight requirement may need further enhancements	Patient-aided scheme is not an intuitive schemes. Heavy scheme because of the using of modular exponentiation operations, which is not proper for tough resource constraints.
[82]	MASK-BAN: Movement-Aided Authenticated Secret Key Extraction Utilizing Channel Characteristics in Body Area Networks	2015	Channel based security scheme	Lightweight, Session key establishment, Mutual authentication	Anonymity, Key escrow resilience	Limited to the on-body nodes.
[83]	BANA: Body Area Network Authentication Exploiting Channel Characteristics	2013	Channel based security scheme	Lightweight	Session key establishment, Mutual authentication	It just considers the LOS of the on-body nodes without generating keys (An authentication scheme only) [82].
[84]	Proximate: proximity-based secure pairing using ambient wireless signals	2011	Proximity based security scheme	Lightweight, Session key establishment	Further optimizations are required to improve its functionality and test its operability for WMSN	It is a generic security scheme that could be used for sensor networks as authors state. It requires a tough nodes distribution constraint.
[79]	A new biometrics-based key establishment protocol in WMSN: energy efficiency and security robustness analysis	2020	Biometric based security scheme	Lightweight, Session key establishment, Resilience to some known attacks	Forward Secrecy	A scope specific solution. It requires all nodes to have the same physiological sign sensor.
[91]	A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem	2011	ID-based signature schemes	Anonymity, Mutual authentication, Session key establishment, Revocability, Lightweight	Key escrow resilience	Remote authentication scheme, designed for client server infrastructure.

Table 2.1: Authentication and Key Establishment Schemes (continued)

Lit.	Title	Year	Solution	Stated Fulfilled Requirements	Potential Improvements	Comments
[93]	An efficient and lightweight certificateless authentication protocol for wireless body area networks	2013	Certificateless signature schemes	Anonymity, Mutual authentication, Non-reputation, Lightweight	Insecure scheme [69]	Adversary is able to trace the user information during the session phase [68]. Remote authentication scheme.
[97]	Highly Efficient Privacy-Preserving Key Agreement for Wireless Body Area Networks	2018	Symmetric based schemes	Anonymity, Mutual authentication, Session key establishment, Lightweight, Unlinkability, Forward secrecy	Key escrow resilience	The authors in this research propose a modification to [71] in order to fulfil the unlinkability and forward secrecy requirements.
[99]	Encryption-free Authentication and Integrity Protection in Body Area Networks through Physical Unclonable Functions	2018	Physical Unclonable Function scheme	Mutual authentication, Lightweight, Resilience to impersonate attack, Revocability	Anonymity	It shares secrets between any nodes pair with the help of the coordinator.
[100]	Lightweight mutual authentication among sensors in body area networks through Physical Unclonable Functions	2017	Physical Unclonable Function scheme	Lightweight, Mutual authentication, Session key establishment, Resilience to impersonate attack, Revocability	Anonymity, Unlinkability	The shared secret in the proposed scheme provides just message authenticity without confidentiality.
[101]	Wireless Body Area Network Identity Authentication Protocol Based on Physical Unclonable Function	2018	Physical Unclonable Function scheme	Lightweight, Resilience to some known attacks	Anonymity, Mutual authentication, Unlinkability, Forward Secrecy	A simple authentication scheme based on PUF that does not fulfil some security requirements.
[102]	A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network	2020	Physical Unclonable Function scheme	Lightweight, Mutual authentication	Anonymity, Unlinkability	TTP-based escrow problem.

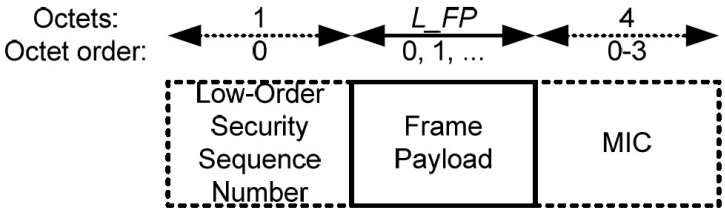


Figure 2.5: MAC Frame Body Format [1]

standard to 255 octets. IEEE 802.15.6 supports two types of frames, secured frames, and unsecured frames. The two communication parties choose the security level according to their security requirements during the association process. The Message Integrity Code (MIC) and "Low-Order Security Sequence Number" fields only exist in the secured frames. The "Low-Order Security Sequence Number" field assists in achieving message freshness by using it in both replay detection and nonce construction [1], while the MIC field is used to achieve message integrity by setting it to the Message Authentication Code (MAC) [1].

The Message Authenticating Code (MAC) process is achieved using the Advanced Encryption Standard (AES-128) as an underlying block cipher algorithm with CCM mode

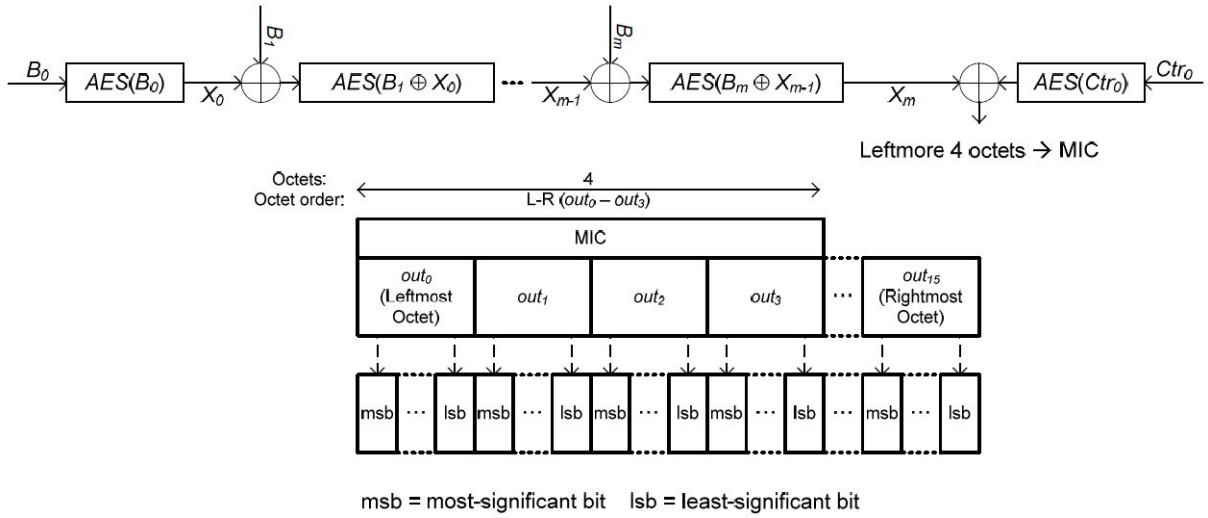


Figure 2.6: MIC calculation and transmit order [1]

(CCM stands for **C**ounter with **C**ipher Block Chaining **M**essage Authentication Code), which is provided in the standard defined by the National Institute of Standards and Technology (NIST) Special Publication SP-800-38C [105]. In security level 1, where no encryption mechanism is provided, the Message Authentication Code (MAC) is computed directly from the frame being transmitted by the sender and the received frame at the receiver. In contrast, in security level 2, the Message Authentication Code (MAC) is computed from the unencrypted version of the frame at the sender and from the decrypted version of the received frame at the receiver. Eq. (2.1) and Eq. (2.2) are used to calculate the MIC field as follows,

$$MIC = LMB_n(M) = AES(Ctr_0) \oplus X_m \tag{2.1}$$

$$X_0 = AES(B0), X_i = AES(B_i \oplus X_{i-1}), i = 1, ..., m \tag{2.2}$$

where M is the message bit string, $LMB_n(M)$ is to specify the leftmost bits of M, $AES(B0)$ is the forward cipher function output of the AES applied to block B and $\oplus$ is the bitwise XOR. AES uses Pairwise Temporal Key (PTK) in unicast secured communication and Group Temporal Key (GTK) in multicast secured communication. Fig. 2.6 shows the process of the Message Authentication Code (MAC) calculation and the transmitting order, starting from the first octet on the left to the last one on the right.

Encryption

Monitoring the vital signs of the human body demands exchanging extremely sensitive information. This information is used to diagnose health conditions, and then an action can be taken to provide the required telemedicine, such as injecting an insulin dosage. Therefore, in order to ensure the confidentiality and privacy of patient health information, data must be exchanged and stored in an encrypted form. Many encryption algorithms have been proposed in the literature, such as 1024-bit RSA [106] and 3DES [107]; however, these traditional encryption algorithms are not suitable for sensor nodes with stringent resource limitations. WMSN requires a lightweight encryption algorithm that is energy and computation efficient. Heavy encryption algorithms with long key size, high number of rounds, or large block size have not been considered due to their inapplicability to WMSN.

Due to the importance and potential applications of lightweight cryptography, NIST began in 2015 the process of standardizing lightweight cryptographic algorithms that fulfill the requirements of constrained devices such as sensor nodes [108]. However, in order to fulfill the resource restrictions, the following aspects should be considered when choosing a suitable lightweight cryptographic function [109]:

- Key size: with extremely limited storage, for instance, MICAz has only 4-KB EEPROM storage [110], the key size of the cryptographic algorithms plays a significant role. A cryptographic algorithm, which has a smaller key size and provides the same security level, is highly recommended. In [111], authors introduce SIMECK, a block cryptographic algorithm. SIMECK is a hardware-oriented algorithm inspired by the SIMON encryption algorithm's design. With its small key size (64, 96, or 128), it shows a more optimized performance regarding memory and power consumption. mCrypton [112] is a block cipher cryptographic algorithm based on SPN (Substitution Permutation Network) structure. It has three different key sizes, where the smallest one is 64 bits. mCrypton is designed to fit into the low resources environment. Another small key size is TWINE [113], which uses 80 bits key with a Feistel structure.

- Block size: The block size is also another important factor towards a lightweight cryptographic algorithm. Smaller block sizes can decrease the processing time and enhance power consumption. Moreover, medical sensors usually transmit small messages containing vital medical signals; therefore, the smaller block size is more efficient. SPECK [114] is a block cipher based on ARX (Addition-Rotation-XOR)

structure. It supports a variety of block sizes ranging between 32 bits and 128 bits. SIMON [114] belongs to the same family of SPECK. However, unlike SPECK, which is designed for software implementation, SIMON is a hardware-oriented algorithm. Another SPN-based lightweight cryptographic algorithm is introduced in [115] named RECTANGLE. It uses 64 bits block size with a bit-slice technique in order to achieve rapid execution.

- Number of rounds: Lightweight cryptographic algorithms usually use simple arithmetic, logic, and shifting operations to fit into the limited resource restrictions such as ARX structure. Using simple operations leads to increasing the number of rounds. PRINCE [116] is a block cipher, a hardware-oriented cryptographic algorithm that aims to enable encryption in just one clock cycle by using a modest number of rounds (12 rounds), which requires a short time to be executed. A 4-round cryptographic algorithm is introduced in [117] called Hummingbird-2. In addition to encryption, it is able to generate MAC (Message Authentication Code). Another encryption algorithm that uses a low number of rounds is LWE [118]. LWE is a 3-round block cipher algorithm. It has been designed to be light enough in order to meet the resource restrictions of medical sensors and IoT. The key and the block size are 64 bits. The performance of LWE is contrasted with well-known lightweight encryption algorithms, such as Rectangle [115] and TWINE [113].
  Moreover, in cryptography, when a number of rounds are required to produce the cipher, a round-key is usually used for each round. The algorithm used to produce the round-key from the key is called the key schedule. Consequently, the more complex the key schedule algorithm is, the more memory and computation power it requires. Therefore, the key schedule algorithm could be regarded as another factor to be considered.

The surveyed encryption algorithms are listed in Table 2.2, stating the block and key sizes, the number of rounds, the algorithm structure, and the possible attacks that might compromise the proposed algorithm.

2.6.2 Intrusion Detection Systems

Secure communication is regarded as the first defense line for WMSN security. However, a further security solution that is able to protect WMSN from intruders and monitor the network for suspicious activities can significantly enhance the security of WMSN. IDSs are introduced to form an additional defense layer to protect from malicious activities

Table 2.2: Lightweight Cryptographic Algorithms

Lit.	Algorithm	Year	Block size	Key size	Rounds #	Structure	Possible Attacks
[111]	SIMECK	2015	32, 48, 64	64, 96, 128	32, 36, 44	ARX	Differential attacks [119]
[112]	mCrypton	2005	64	64, 96, 128	12	SPN	Meet-in-the-Middle Attack [120]
[113]	TWINE	2011	64	80, 128	36	GFN	Single-key attack using biclique technique [121]
[114]	SIMON	2013	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192, 256	32, 36, 42, 44, 52, 54, 68, 69, 72	ARX	23-round linear attack with key guessing technique [122]
[115]	RECTANGLE	2015	64	80, 128	25	SPN	Differential attack (18 out of 25 rounds) [115]
[116]	PRINCE	2012	64	128	12	SPN	Key recovery attack (6 rounds version) [123], Sieve-in-the-middle attack (8 rounds) [124]
[117]	Hummingbird-2	2011	16	128	4	SPN	A probabilistic attack (theoretical) [125]
[114]	SPECK	2013	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192, 256	22, 23, 26, 27, 28, 29, 32, 33, 34	ARX	Sub-cipher attack [126]
[118]	LWE	2020	64	64	3	SPN	No reported attacks yet.

and inside and outside abuses. Based on the methods of detection, IDS schemes can be grouped as follows:

Signature-based IDSs

By defining a signature for each attack pattern, the IDS can detect malicious activities when an attack pattern matches any defined signature. The main disadvantages of the signature-based IDSs are the need to be updated periodically and the inability to catch unknown attacks. A hybrid, lightweight, multi-level, and distributed model that imitates the human immune system with a signature database has been presented in [127]. It takes advantage of the danger model used by our immune system, where a particular type of cell, Dendritic cells, stimulates other cells to form immune responses to any antigens. Copying this alerting mechanism to WSN, in addition to the use of predefined features, enable the IDS to detect intrusions.

Anomaly-based IDSs

By profiling the normal traffic and operations of the network, anomaly-based IDSs could raise an alert when anomaly behavior is captured. According to [128], the anomaly-based IDSs could be divided into three categories:

Statistical-based: It depends on building a statistical reference profile for the network in the optimal conditions without any malicious activities. Afterward, periodic profiles

are being generated for the monitored network. The generated profiles are then compared to the reference model to calculate the anomaly probability based on a threshold. A cooperative and a statistical-based IDS has been proposed in [129]. The authors use various types of statistics, such as Forward Percentage (FP) and Local Forward Percentage (LFP), in order to detect anomaly behavior. Owing to using the attack's consequence and technique, the proposed scheme is able to determine the attack type and the attack origin. Although it gives good accuracy in detecting selfish activities, such as DoS attacks and sleep deprivation via malicious flooding attacks, it shows less accuracy in detecting blackhole and spoofing attacks. However, sharing the responsibility of intrusion detection between nodes with the same resources could be regarded as a good choice to share the computational overhead.

Knowledge-based: A previous knowledge of the networks under different circumstances ranging from normal conditions to under attack, is used to detect malicious activities. Expert systems and Finite State Machine algorithms are used as a detection engine for a knowledge-based IDS [127, 128].

Machine learning based: A more intelligent method to detect any malicious activities in the network, even new ones. Machine learning-based IDSs are able to learn from analyzed patterns and update their intrusion detection engine periodically without being explicitly programmed, which gives them the ability to detect not just well-known attacks but new and unknown attacks as well. Random Forest, Genetic Algorithm (GA), Support Vector Machines (SVMs), Neural Networks, and Logistic Regression are all examples of machine learning algorithms used in the literature. iDetect [130] is a distributed IDS that aims to enhance the detection rate of attacks and choose the most suitable features set that optimize the performance and saves computational power. iDetect IDS is implemented using a multi-objective genetic algorithm to produce the optimal set of features to be used as an input for the intrusion detection engine in order to detect WMSN attacks. The authors in [131] proposed a distributed IDS framework based on mobile agent technology where all nodes inside WMSN participate in the malicious activities detection process by migrating the intrusion detection software from one node to another. The authors state that their framework is able to reduce the communication burden of sending the log in the traditional frameworks, which results in reducing the power consumption. Another hierarchical and distributed IDS based on autonomous mobile agents where the detection software moves from one node to another has been presented in [132]. The proposed IDS framework is evaluated using five different machine learning algorithms

Decision Tree (DT), Support Vector Machines (SVM), Random forest (RF), Naive Bayes Classifier (NBC), K-Nearest Neighbors (KNN). The proposed framework's performance results in around a 6% increase in the power consumption; however, the results do not show a positive reflection of the proposed framework on accuracy. The authors in [133] investigated personal medical devices' vulnerabilities by launching different kinds of attacks. Moreover, they proposed HEKA, a passive IDS, to monitor and detect malicious activities. The anomaly detection performance of HEKA is evaluated using four machine learning algorithms, SVM, KNN, RF, and DT, which have been trained using the generated n-grams of the network traffic. The results show an average detection accuracy of around 98% for MITM, Replay, false data injection, and DoS attacks. Moreover, the performance of composite attacks of MITM with false data injection and MITM with replay attacks are evaluated and showed an average accuracy of around 95%. Another approach to detecting anomalies in the healthcare systems is proposed in [134]. The authors combined two kinds of features, network and biometrics, to enhance the detection performance. The collected dataset is used to train four machine learning algorithms RF, KNN, ANN, and SVM. The Results show an AUC score improvement between 7% and 25%.

Specification-based IDSs

It uses a technique that is located in the middle between signature-based IDSs, where predefined rules are used to detect well-known attacks, and anomaly-based IDSs, where normal behavior is defined to detect any abnormal behavior, even unknown ones in contrast to signature-based IDSs [128]. In specification-based IDSs, programmers manually define constraints and features that describe the normal operating behavior in contrast to anomaly-based IDSs, where they are generated automatically. Consequently, the intrusion detection engine can detect suspicious behavior by finding the deviation of the two behaviors.

The IDSs surveyed in this section are listed in Table 2.3, stating the used techniques, topology architecture, used simulators, attacks tested for verification, and the fulfilled requirements.

2.6.3 Trust Management Systems

Authentication and key establishment between nodes in WMSN are an imperative task to ensure a high security and privacy level. However, maintaining the successfully established security mechanisms demands a trust relationship between nodes. TMSs assess

Table 2.3: Intrusion Detection System Schemes

Lit.	Title	Year	Solution	Technique	Architecture	Simulator	Attacks	Fulfillment	Comments
[127]	A Multi-Level Intrusion Detection System for Wireless Sensor Networks Based on Immune Theory	2018	Hybrid of anomaly and signature-based IDS	Immune theory technique called Danger Theory	Distributed	Cooja [135]	Blackhole attack, Selective forwarding attack, DDoS attack, Wormhole attack	A lightweight IDS approach that considers power consumption.	This approach is originally proposed for WSN and needs modifications to fit other technologies.
[133]	Heka: A novel intrusion detection system for attacks to personal medical devices	2020	Anomaly-based IDS	Four machine learning algorithms (SVM, KNN, DT, RF)	Distributed	N/A	MITM, replay, false data injection and DoS attacks	A high detection accuracy for single and combined attacks.	The performance of HEKA is evaluated using datasets generated from different real devices.
[129]	A cooperative intrusion detection system for ad hoc networks	2003	Statistical-based IDS	Mobile agent technology	Distributed	ns-2 [136], MobiEmu [137]	Blackhole attack, Spoofing attack, Selfishness attack, DoS attack, Sleep deprivation attack	A good accuracy in detecting some attacks. It shares the overhead between nodes.	It is able to identify the attacker as well as the attack type.
[130]	iDetect: an intelligent intrusion detection system for wireless body area networks	2016	Anomaly-based IDS	Multi-objective genetic algorithm	Distributed	TOSSIM [138]	Jamming attacks, Selective forwarding attack	A good accuracy without breaching the acceptable level of energy consumption.	It reduces the computation complexity by using a reduced set of features for detection.
[132]	Distributed intrusion detection using mobile agents in wireless body area networks	2017	Anomaly-based IDS	Five machine learning algorithms (DT, SVM, RF, NBC, KNN)	Hierarchical and distributed	Castalia [139]	Denial of Service Attacks, Data Falsification, Passive Listening	Accuracy and power consumption are assessed for the following machine learning algorithms DT, SVM, RF, NBC, KNN.	Around 6% rise in power consumption; however, results do not show the positive reflection of the proposed framework on the accuracy metric.
[131]	Autonomous mobile agent based intrusion detection framework in wireless body area networks	2015	Anomaly-based IDS	Mobile agent technology	Distributed	Not used	No attacks are tested for evaluation	Authors state that the proposed framework is able to reduce the communication overhead.	The proposed framework has to be evaluated to find out its feasibility for WMSN.
[140]	A Novel Intrusion Detection System for Wireless Body Area Network in Health Care Monitoring	2010	Anomaly-based IDS	Negative Selection Algorithm	Centralized	QualNet [141]	Denial of Service attacks	A good accuracy in detecting compromised nodes. Minimising the false positives rate.	The performance of the proposed scheme has been evaluated for different routing protocols.
[134]	Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study	2020	Anomaly-based IDS	Four machine learning algorithms (RF, KNN, ANN, SVM)	Distributed	N/A	MITM attacks	A dataset representing two kind of features (network and biometrics).	Further investigation is required to optimize the system overhead.

nodes with an aim to differentiate between trustworthy and untrustworthy nodes. While traditional security mechanisms rely on the nodes' current status, TMSs consider the present behavior of the nodes as well as their behavior history. Thus, deploying a TMS enhances the level of security by continuously monitoring the nodes' behavior and their performance. In light of this, trust can be defined as follows:
Node X trusts node Y if and only if X has adequate confidence in Y's behavior and performance in the future.

Similar to other security aspects of WMSN, deploying a TMS is a challenging task [142], and many factors must be taken into account when proposing a TMS for WMSN. These factors include:

- WMSN architecture.
- Scarcity of resources.
- Communication overhead.
- Resistant to TMS-related attacks.

Taking into account the cooperative manner of nodes within the WMSN, many potential applications emerge for using TMSs, ranging from access control and role assignment [143] to trust-based routing protocols [144]. Therefore, TMSs will play a vital role in any interaction and cooperation between nodes inside the WMSN.

The trust estimation engine usually depends on two sources of information in order to produce the trust value:

- Direct trust: Trustor monitors and evaluates the trustee's successful and unsuccessful interactions when both have direct communication and without the help of a third party.
- Indirect trust: When the trustor is not adjacent to the trustee or does not have any historical information to assess the trustee, the trustor depends on the received recommendations from other parties to evaluate the trustee.

TMSs are prone to several internal attacks; therefore, a robust TMS design is sought to defeat these kinds of attack. Below is a list of the most common attacks:

- On-off attack: The malicious nodes change their behavior alternately between good and bad manners in order to remain undetected, and consequently, cause serious

damage and degrade the overall performance significantly [67].

- Bad-mouthing attack: One of the dishonest recommendations attacks. This type of internal attack occurs when a malicious node colludes to destroy some victimized nodes' reputation by giving negative recommendations [145].
- Ballot stuffing attack: Another type of dishonest recommendations attack. It is the opposite scenario of the bad-mouthing attack. It happens when some nodes give positive recommendations for a malicious node [146].
- Collusion attack: Unlike bad-mouthing and ballot stuffing attacks where just one malicious node provides dishonest recommendations. In the collusion attack, a set of nodes participates in the attack by providing false information, which may mislead the system into making unfair decisions depending on the received false information from different sources [147].

Based on the method of estimating the trust value, TMSs could be divided into four groups:

Fuzzy logic based TMSs

In fuzzy-based TMSs, trust value is estimated using fuzzy logic and predefined criteria that have a fuzzy nature.
DTMS [148] is a fuzzy logic-based TMS, which is uniformly distributed amongst the sensor nodes. The authors suggest many different criteria to be taken into account when estimating the direct trust value. However, considering all these criteria can increase the computational overhead. DTMS estimates the current trust value by weighting the direct and indirect trust values for each adjacent node. Afterward, the total trust is estimated from the current and previous trust values using the same weighting technique. As a result, DTMS shows superior performance compared with earlier trust management models. However, using trust matrices and tables can produce a significant network overhead [149]. FTM-IoMT [150] is another fuzzy-based trust management system proposed for the Internet of Medical Things (IoMT) to prevent Sybil attacks. It is a centralized approach that uses integrity, receptivity, and compatibility to evaluate the trust value for the requesting nodes. However, it shows significant processing overhead, which requires further investigation to reduce the packet delivery latency and the server-side overhead.

Probability based TMSs

Many research projects are put forward using the probability distribution theory in order to estimate the trust value using the inference of former values. Almost all probability-based TMSs use beta probability distribution to evaluate the reputation value; however, other probability distributions such as exponential probability distribution and binomial probability distribution are also used in the literature [151, 152].
To the best of our knowledge, RFSN [153] is the first trust management scheme that uses the beta distribution to evaluate the reputation value for WSN. RFSN uses a watchdog mechanism to collect new observations, which are used to update the posterior reputation value. Another probability-based TMS is ETRES [151], which uses the exponential probability distribution in order to represent the reputation value of a node by assuming that the future behavior will have the same mode of the node history. ETRES only considers indirect recommendations when the certainty of the trust level is not adequate to enhance power consumption. The uncertainty is obtained using the information entropy theory. Both direct reputation value and indirect one are weighted in order to give more significance to the up-to-date information and the most reputable recommender, respectively. Later, the overall trust value is estimated using the confidence factor technique. Comparing to RFSN [154] and BTMS [155], ETRES shows a bit higher performance.

Weighting based TMSs

A simple way to evaluate the nodes' behavior and generate the trust value for each node based on weighting the nodes' reputation over time. Although this type of TMS is easy to deploy, it lacks a robust math ground [142]. TMR [156] is a weighting-based TMS using risk assessment. Considering the risk factor can prompt a fast reaction to misbehaved nodes by making destroying the obtained trust value easier than building it. Evaluating the risk factor enhances the trust management model's reliability by making it more sensitive to any malicious activities. In TMR, both direct trust and indirect recommendations, in addition to the risk factor and previous trust value, are considered to evaluate the current trust value. RaRTrust [157] is another example of weighting-based TMS where the authors use the timing window technique for processing the previous trust values. RaRTrust considers the risk assessment technique by adopting a balancing factor that makes acquiring trust harder than losing it. While RaRTrust shows resiliency to on-off attacks and bad-mouthing attacks, TMR is just able to defeat on-off attacks.

Other TMSs

There are some TMSs that do not fall within the previous categorization. In [158], a cluster-based with a 3-tier architecture TMS is proposed where tier 1 just focuses on nodes registration. In tier 2, five levels of misbehavior are defined. Although the proposed scheme considers the previous information to find out malicious nodes during the trust process, the used machine learning algorithm is not disclosed. Furthermore, the authors give the same weight to both direct and indirect trust values. To evaluate the overall trust value, a traditional summation technique is used, which may lead to an un-scaled value depending on the number of adjacent nodes. In tier 3, the whole process is about monitoring the consumed power of the Cluster Head (CH), and when a threshold is triggered, a new CH is chosen based on the level of trust that nodes have.

The TMSs surveyed in this section are summarized in Table 2.4, stating the topology architecture, simulators used, and the proposed contributions.

2.7 Research Opportunities

WMSN brings new research opportunities to wireless healthcare networks. However, the ongoing development of WMSN encounters severe resource constraints, in addition to challenges posed by the sensitive nature of WMSN data and the potentially catastrophic consequences of compromising network nodes. Therefore, many research opportunities emerge in order to meet these strict restrictions. This section states potential research directions to enhance WMSN technology, including security, communication protocols, and power consumption considerations. In this thesis, we have addressed two research gaps as showin in Section 2.7.1, assuming that a secure authentication and key establishment that fulfills all the security requirements specified in Section 2.6.1 is in place.

2.7.1 Bridged Research Gaps

In this subsection, we discussed the research gaps that have been bridged in this thesis.

1. Trust Management System

Trust management is regarded as a complementary solution to enhance the overall security of WMSN. In TMS, the behavior of surrounding nodes is monitored to detect any misbehavior, such as dropping attacks. As traditional routing metrics cannot deal with the free will of the relaying nodes, TMS could be a potential security countermeasure to

Table 2.4: Trust Management System Schemes

Lit.	Title	Year	Solution	Technology	Architecture	Simulator	Contributions	Comments
[148]	A fuzzy fully distributed trust management system in wireless sensor networks	2016	Fuzzy logic based TMS	Wireless Sensor Networks	Uniform distributed arcitecture	TRMSim-WSN [159]	DTMS shows superior performance comparing with earlier TM models. It detects compromised, selfish and malicious nodes.	Using trust matrices and tables produces significant overhead [149].
[151]	An Effective Exponential-Based Trust and Reputation Evaluation System in Wireless Sensor Networks	2019	Probability based TMS	Wireless Sensor Networks	Distributed	MATLAB	A bit better performance comparing to RFSN [154] and BTMS [155]. Resilience to on-off attack. Resilience to selective forwarding attack. Detecting compromised nodes.	According to the authors, energy consumption will be considered in the future research.
[156]	A reliable trust management scheme in wireless sensor networks	2015	Weighting based TMS	Wireless Sensor Networks	Distributed	MATLAB	It considers the risk factor. Resilience to on-off attack.	The used weighting factors have to be optimized.
[157]	A risk-aware reputation-based trust management in wireless sensor networks	2016	Weighting based TMS	Wireless Sensor Networks	Distributed	MATLAB	It considers the risk factor in addition to a sliding timing window of interactions and local ratings. Resilience to on-off attack.	RaRTrust is able to reduce the effect of bad-mouthing attack.
[158]	A cluster based energy efficient trust management mechanism for medical wireless sensor networks (MWSNs)	2018	Machine learning technique	Medical Wireless Sensor Networks	Cluster based with 3-tier (Hybrid)	ns-2 [139]	Energy efficient by rotating cluster head among the nodes.	The used machine learning algorithm is not disclosed. Unscaled total trust value.
[150]	FTM-IOMT: Fuzzy-based trust management for preventing Sybil attacks in internet of medical things	2020	Fuzzy logic based TMS	Internet of Medical Things	Centralized	Cooja [135]	A centralized approach to evaluate the trustworthiness using integrity, compatibility and receptivity features.	Further work is needed to reduce the server side overhead and the packet delivery latency.

evaluate the nodes' trustworthiness and avoid any malicious path. However, deploying a TMS in WMSN is challenging for several reasons. First, the TMS itself is vulnerable to some specific attacks, such as on-ff and bad-mouthing attacks, which require a robust design to defeat these kinds of attacks. Second, the limited resources of WMSN and its sensitive applications. Third, there is a lack of a realistic testbed to evaluate the robustness of TMS proposals. The first two points have been addressed in Chapter 3 where a robust TMS has been proposed to deal with on-off attacks. It is worth noting that the proposed TMS is not prone to bad-mouthing attacks as it is only consider observations to evaluate the trust relationship. Moreover, a realistic trust management testbed has been proposed in Chapter 6 to address the third point.

2. Routing

Routing protocols applicable to WMSN are still an open research opportunity. Dedicated protocols need to be considered taking into account the nature of WMSN, where packets have to pass through relaying nodes in order to reach the sink node. There is a wide range of routing protocols proposed for WSN. However, it is not an easy task to choose the most suitable routing protocol for WMSN, which conserves power consumption and minimizes communication overhead. Different approaches have been proposed in the literature, such as thermal-aware and postural movement-based routing protocols [160]. However, these routing proposals could be regarded as scope-specific solutions. Moreover, they cannot ensure reliable data transfer. This research gap has been bridged in Chapters 4 and 5 of this thesis.

2.7.2 Potential Research Gaps

In this subsection, a list of potential research opportunities are discussed in the domain of healthcare wireless networks, which have not been addressed in this doctoral research.

1. Authentication and Key Establishment

Authentication and key establishment pose a wide range of research opportunities. However, most of the literature's proposed schemes are either still vulnerable to some kinds of attack or do not fully address the WMSN security requirements discussed in Section 2.6.1. Moreover, any proposed scheme for key establishment and key revocation, when necessary, must prove its merit in resource conservation. It is worth mentioning that authentication and key establishment is a critical issue to tackle; however, bridging this gap is out of the scope of this thesis.

2. Encryption

Data encryption is essential when personal and sensitive data are involved, as is the case with WMSN. Encryption should provide data protection in both transmission and storage stages to prevent attacks such as eavesdropping-based attacks. Proposing a lightweight and energy-aware encryption algorithm that satisfies both the security requirements and the stringent resource constraints is still an open area of research with vast contribution opportunities.

3. Intrusion Detection System

Intrusion detection is another potential area of research. Traditional security schemes aim to keep attackers out; however, they do not have the ability to detect internal attacks and react to them accordingly. Therefore, applying IDS to WMSN can address attacks and vulnerabilities where traditional schemes fail to do so. For instance, DoS attacks affect the network's availability and may not be detectable by traditional security countermeasures. Therefore, further investigation for developing a WMSN-specific IDS is necessary to detect different types of attacks.

4. Sustainable Power Source

One of the main challenging tasks is maintaining a continuous source of power to nodes implanted inside the body since replacing the battery of implanted devices requires surgery. Therefore, finding other sources of energy is still an open area of research. For instance, according to [161], each cm^2 of the human body is able to produce 20 mW compared to $15mW/cm^2$ produced by solar panels, which makes it a promising alternative source of energy.

5. Security Evaluation

Most security countermeasures are evaluated using simulations in order to prove their effectiveness. This is due to the difficulty of getting real testbeds. Therefore, there is a lack of realistic evaluation regarding the effectiveness of the proposed security mechanisms applicable to medical devices. This creates additional research opportunities for the realistic evaluation of WMSN security countermeasures.

6. Smart Transmission

Transmission is the main contributor to power consumption. Adopting any smart transmission technology, such as cognitive radio, may enhance power consumption and prolong battery life dramatically. Therefore, the optimized transmission is a significantly challenging task and still has opportunities for further research contributions. It is worth mentioning here that our proposed methods in Chapters 4 and 5 aim to reduce the number of transmissions by always choosing the shortest reliable paths.

7. Data Processing

Data processing is another open research area in the Internet of Things (IoT) field, of which WMSN is part. WMSN nodes generate a massive amount of data that needs to be stored and processed in a secure manner to guarantee patients' security and privacy. On the other hand, different WMSN sensors may register the same physiological signals, which consequently pose a challenge for data processing. Therefore, effective methods for big data processing and data fusion are possible research opportunities.

8. Access Control

Developing an adequate access control mechanism is another possible research direction. Access control allows physicians and technicians to adjust the configuration of the sensor nodes. Two types of access control should be considered, which are attribute-based and role-based, in order to ensure patient safety and privacy.

9. Upper Stack Protocols

In February 2012, the first and the only WMSN standard were released [1]. IEEE 802.15.6 standardizes the Media Access Control (MAC) layer that supports three different physical layers. It provides low power and short-range wireless communication for devices that operate on, in, or around the human body. Thus, the IEEE standard does not provide a full protocol stack. IEEE 802.15.4/Zigbee is a full protocol stack developed by Zigbee Alliance, an organization working in conjunction with IEEE TG4. IEEE 802.15.4/Zigbee is developed to fulfill the requirements of WSN based on IEEE 802.15.4 [162]. IPv6 over Low Power Personal Area Network (6LoWPAN) is another upper stack introduced by the Internet Engineering Task Force (IETF) to fit into the requirements of WSN. Many research [163, 164, 165, 166] investigate 6LoWPAN and Zigbee protocol stacks as upper stack for IEEE 802.15.4. However, both upper stack protocols have to be fully

investigated with IEEE 802.15.6.

2.8 Summary

In this chapter, a review of the current research and future research directions on Wireless Medical Sensor Networks have been presented. First, a concise overview of WMSN topology and design requirements have been discussed. Furthermore, security requirements and challenges have been investigated, and a wide range of threats and attacks are identified. Different types of security countermeasures have been discussed throughout the chapter to meet the security requirements of WMSN and defeat potential threats. Specifically, secure communication, intrusion detection systems, and trust management systems. Finally, potential research opportunities and directions have been proposed.

Chapter 3

An Effective Lightweight Trust Management System

Ensuring reliable data delivery requires choosing the trustworthy path to the destination. Although deploying traditional security countermeasure is imperative to protect the network, they cannot deal with other nodes' free will. Therefore, there is a need to integrate a security tool into the routing protocol to help it avoid paths with malicious sensor nodes. In this chapter, a novel method is developed to evaluate the trust relationship between sensor nodes. Our proposed Lightweight Trust Management System (LTMS) provides a further line of defense to detect packet dropping attacks launched by malicious, selfish, or even faulty sensor nodes. Simulation results show that LTMS is more robust against complicated on-off attacks and can significantly reduce the processing overhead.

The main findings of this chapter were presented at the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) 2020 [4] and IEEE Conference on Communications and Network Security (CNS) 2020 [66].

3.1 Introduction

Security concerns are still challenging obstacles to the widespread adoption of WMSNs. Internal security threats, such as packet dropping attacks, may have catastrophic consequences. Compromised, selfish, or even faulty Sensor Nodes (SNs) may drop critical messages, such as urgent notifications of abnormal heart rhythms or insulin dose release

orders, and consequently violate routing protocol and endanger the patient's life. This kind of attack cannot be prevented by traditional cryptographic measures described in Chapter 2.6 as malicious SNs are already authenticated and may have a copy of the security keys. Therefore, establishing a trust relationship between SNs within the network is regarded as a complementary security solution to the cryptographic measures to protect the network from malicious activities [145].

Trust Management Systems (TMSs) offer a further level of defense against internal attacks by monitoring other nodes' behavior. Various potential applications emerge from establishing a trust relationship between nodes ranging from routing [167] to defeating threats [168]. Several TMSs have been introduced in the literature for WSNs [169, 170, 171, 168]; however, only a limited number of TMS have been proposed to fit WMSN [145].

In addition to the security concerns inherited from Wireless Sensor Network (WSN), WMSN has unique characteristics that impose further challenges in adopting the existing security measures of WSN. Therefore, operation requirements such as traffic rates, network topology, resource limitations, and intolerant applications must be taken into account in order to design an effective trust scheme that fits WMSN. First, some SNs generate low traffic rates around 1 packet/s [172], such as heart rate sensors. Second, the network topology of WMSN is star topology in accordance with IEEE 802.15.6 standard [1]. Third, SNs, especially implanted ones, suffer from strict resource limitations. For instance, The battery is expected to last for years before getting replaced via surgery. Hence, a lightweight trust scheme is a must. Fourth, WMSN provides very critical applications that cannot tolerate any prolonged detection periods.

On the other hand, although TMS are a promising solution to detect packet dropping attacks and other misbehaviors, they can be manipulated by intelligent adversaries. TMS schemes are prone to on-off attacks, where smart adversaries change their behavior between good and bad alternately in order to redeem themselves from the burden of the bad behavior [67]. Moreover, adversaries can launch more sophisticated on-off attacks by changing the packet dropping rates or launching on-off attacks with non-identical periods.

3.2 Contribution

The main contribution of this chapter is to propose an effective and lightweight trust management system for WMSN. Two methods have been proposed to evaluate the trust relationship between SNs. The first is lightweight to fit in-body SNs, while the second is provided with an on-off protection module for on-body and off-body SNs. A comprehensive analysis is offered to prove the robustness of LTMS against on-off attacks. Moreover, all the necessary code is made available on GitHub for reproducibility purposes.

3.3 Related Work

Trust and reputation systems have emerged to defend against internal attacks. Various methods to model the trust relationship between nodes ranging from probability-based to fuzzy logic, are proposed in the literature for both Mobile Ad hoc Network (MANET) and WSN [173, 170, 168, 156, 174, 169, 153]. However, few research have targeted WMSN [145, 175]. Therefore, trust schemes proposed for either MANET or WSN have to be further assessed in terms of WMSN operating conditions, network topology and resource limitations.

Numerous research is put forward based on the Bayesian inference since future behavior can be inferred based on historical observations. Different kinds of probability distributions are used for modeling in order to evaluate the trust value, such as beta distribution [153, 168, 176], binomial distribution [175], exponential distribution [174] and Gaussian distribution [177]. Although the probability theory offers a robust mathematical basis to model trust and reputation systems, it needs prolonged time to detect malicious activities since the trust value represents a long-term value [178]. To overcome this limitation, different approaches are adopted in the literature. The longevity factor, which gives more weight to recent observations, has been widely used in the literature to reflect the current behavior of the trustee [153, 174, 175, 168, 176]. A sliding time window is also proposed in the literature to enhance the malicious detection rate [170, 145, 179]. The punishment factor is another method to overcome the aforementioned issue. It has been widely adopted in the literature to give more weight to bad behavior [180, 156, 67].

To the best of our knowledge, authors in [181] are the first to introduce the beta-based reputation model. Later on, Reputation-based Framework for Sensor Networks (RFSN) [153] was introduced as a beta-based reputation framework for WSN. RFSN is built to integrate new updates into the reputation evaluation process. Those updates are obtained

using a watchdog mechanism that collects cooperative and non-cooperative interactions. In addition, RFSN adopts a forgetting mechanism with a view to giving more weight to recent observations. Because of the effectiveness of the probability-based reputation evaluation system, many improvements are introduced in the literature that show promising results [145, 168, 176, 174]. Authors in TWSN [169] use a different technique to evaluate direct trust by calculating the forwarding ratio based on the accumulated successful and unsuccessful actions. Afterward, the forwarding ratio is compared with the previous one to compute the node behavior's fluctuation, which will be used later to evaluate the direct trust using the cosine function. Authors in [145] proposed ReTrust, which is a trust management scheme for WMSN. According to the authors, ReTrust is a lightweight and attack-resistant scheme that fits WMSNs. ReTrust adopts a sliding time window to update the trust value by using a dynamic exponential decreasing longevity factor in order to underweight old observations, which causes a significant processing overhead. Moreover, it ignores any historical interactions beyond the sliding window, which could be manipulated by smart adversaries. Many trust schemes have used ReTrust as a benchmark scheme to contrast with [156, 67, 170, 179, 182]. BDTMS [175] is a trust management scheme for WSN that targeting healthcare applications. It uses a longevity factor to reflect the recent behavior of the trustee. Neither ReTrust nor BDTMS considers the unique characteristics of WMSN, such as traffic rates, or evaluates the processing overhead. However, these characteristics have been considered in developing our trust management scheme in order to fit WMSN.

Although different approaches are introduced in the literature to reduce the detection time as discussed in the above paragraphs, the proposed methods are still insufficient to address the reported drawback in [178]. First, using the longevity factor to speed up the detection process influences the evaluated trust value when reducing the value of the longevity factor, which means the evaluated trust value does not reflect the trustworthiness of the trustee. Second, adopting the sliding time window technique to enhance the detection time has two shortcomings. The trust value in such schemes represents a short-term value limited to the length of the time window, which does not necessarily reflect the trustee's trustworthiness. Moreover, increasing the length of the time window increase the computational overhead. Finally, incorporating a punishment factor may increase the processing overhead and is not able to deal with various dropping patterns. Therefore, there is a need to a novel method that is able to detect behaviour changes swiftly with minimum computational overhead. Moreover, the proposed method must comply with the WMSN operating conditions.

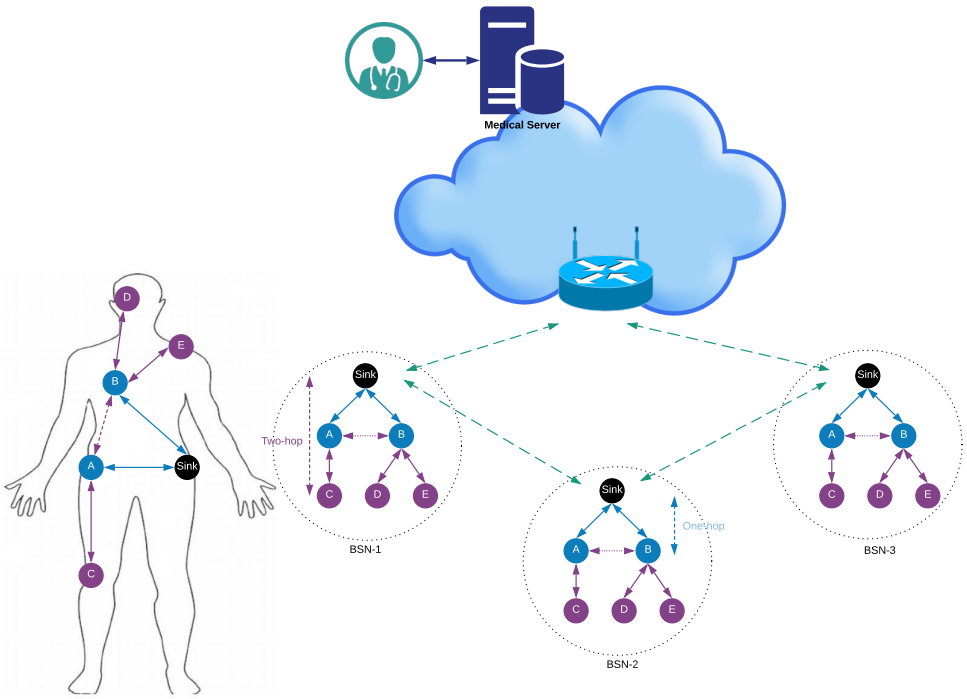


Figure 3.1: Network Structure

3.4 Network and Threat Models

A WMSN may comprise up to 64 SNs in accordance with IEEE 802.15.6 standard [1]. The SNs have strict resource constraints, and this plays a significant role in adopting any security measure. Moreover, in-body SNs have further power limitation as replacing the battery may require a surgery. For instance, pacemakers' batteries, which are lithium iodide cells, are expected to last for around seven years before being replaced via surgery [183]. Fig 3.1 illustrates an exemplary WMSN in a hospital. All sensed information is to be sent to the BSN's sink node, which in turn forwards this critical information to the medical server for processing. Authorized physicians are then able to access the patient's medical records and intervene if necessary.

3.4.1 Network Model

SNs are classified into three different types based on their role. The sink node is the gateway of the WMSN to external networks. End SNs are designed to sense the body signals and exchange the messages with the sink if they are in direct communication or via relay SNs when they are out of the communication range. The topology is a multihop

star topology. In this research, we differentiate between two types of SNs based on their resource limitations and traffic rates as follows.

- In-body SNs are end nodes implanted inside the human body to sense the body's vital signs, such as pacemakers. They have minimal resources, and their power source is expected to last for years. They use low traffic rates, around one packet per second [172]. These unique features of resources and traffic rates impose further requirements to deploy any proposed TMS.
- On-body and off-body SNs are distributed on the body surface or in the vicinity of the body, such as heart rate and motion sensors. They have better resources and processing capabilities. Moreover, replacing nodes' batteries does not require surgical interventions. They usually use higher traffic rates as they are expected to relay messages for other SNs.

This differentiation is used to propose two different trust evaluation methods.

3.4.2 Threat Model

WMSN is prone to different security threats because of the sensitive data it generates and the broadcast nature of the wireless networks. In this research, we assume that secure communication is already fulfilled by deploying an authentication and key establishment scheme that meets the security requirements discussed in Chapter 2. Internal attacks are usually launched by SNs that have passed the authentication process and may have had a copy of the security keys. These SNs are regarded as legitimate SNs from the cryptographic measures perspective. By monitoring the behavior of the SNs within the WMSN, the TMS can defend against internal attacks, such as packet dropping attacks. Dropping packets cannot just disrupt the routing operation, but it may endanger the patient's life by dropping physician notification messages or drug delivery orders.

On the other hand, the process of evaluating the trust based on direct observations is vulnerable to on-off attacks, where the malicious node changes its behavior between malicious and benign alternately with a view to keep itself undetected [67]. The on-off attack cycle consists of on and off periods. Malicious agent behaves badly during the on period and well during the off period. Therefore, defending against this kind of attack requires a robust design to protect the trust evaluation engine.

3.5 Trust Evaluation

In this section, our proposed trust evaluation scheme for WMSN is presented. Two methods are proposed to evaluate the trust value. The first is introduced for in-body SNs to meet its extreme resource limitations, while the second is for on-body and off-body SNs to further immune the trust evaluation engine from on-off attacks.

3.5.1 Definitions

The process of evaluating the trust value of nodes within the network is done in a distributed manner, where each node has its instance of the trust evaluation engine. As the trust relationship is established between two entities for a specific task, we refer to the party who performs the action as an *agent* and the party who monitors the agent and holds the trust value as a *subject*. The *action* could be any service provided by an agent to a specific subject, which is packet forwarding in our case. Defining trust and reputation is still an open issue [142]. Although reputation and trust are used interchangeably sometimes in the literature, there is a difference between them. Reputation value is usually inferred from the behaviour history, while calculation of trust value is a subjective expectation that may consider more factors, which are not necessarily related to the trustee's trustworthiness, such as the trustee's location. *Reputation* is defined as the perception that the agent does not have any intention to change its known behavior. Therefore, reputation value is inferred directly from the observation history. *Trust* is defined as having adequate confidence in the agent's future behavior. It is a subjective value as the subject may consider different factors to evaluate the trust value that are not necessarily related to the agent's honesty. In this context, we assume that the subject overhears the agent to observe forwarded packets, which considered good behavior, and dropped packets, which considered bad behavior. These direct observations are used to evaluate direct trust in order to identify malicious agents. Reputation-based trust is defined as follows:

$$T_{ij}(t) = f(Rep_{ij}(t)) \tag{3.1}$$

where $T_{ij}(t)$ represents the trust value maintained by the subject i for the agent j at the time unit t and $Rep_{ij}(t)$ is the reputation value.

3.5.2 Beta Distribution based Trust Model

The reputation value maintained by the subject represents a belief that the subject predicts the agent's future behavior based on it in a manner that reduces uncertainty. The Bayesian reputation model assumes that the behavior of the agent can be estimated based on a probability distribution. The expected value of the probability distribution represents the reputation value, which gets updated once new observations are available using Bayes' theorem [178]. Therefore, the updated (posterior) parameters of the probability distribution function is calculated by adding the previous (prior) version of the parameters to the current observations.

In LTMS, we consider the packet forwarding service to evaluate the trust relationship between the subject and the agent as it is an essential service for multi-hop ad-hoc wireless networks and could be combined with a routing protocol to ensure reliable data delivery. In this case, the subject maintains two time series $s(t)$ and $u(t)$ for successful and unsuccessful actions, respectively. The observed action has two states to represent if the packet is forwarded successfully or not. These observations are considered a sequence of trials with binary outcomes (Successful, Unsuccessful), which form a binary space of disjoint elements. Therefore, this binomial Bayesian reputation system can be modeled using a Beta Probability Density Function (PDF) as follows [181].

$$\begin{aligned} f(p_x|\alpha,\beta) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p_x^{\alpha-1}(1-p_x)^{\beta-1} \\ &= \frac{1}{B(\alpha,\beta)} p_x^{\alpha-1}(1-p_x)^{\beta-1} \\ where & \begin{cases} 0 \leq p_x \leq 1 \\ \alpha > 0 \\ \beta > 0 \end{cases} \end{aligned} \tag{3.2}$$

$\Gamma(.)$ is the gamma function, $B(.)$ is the beta function and the coefficient $\frac{1}{B(\alpha,\beta)}$ is a normalizing constant. There are two restrictions for Eq. 3.2. The first is $p_x \neq 0$ if $\alpha < 1$, and the second is $p_x \neq 1$ if $\beta < 1$. The reputation value is the expected value of Eq. 3.2

and is defined in Eq. 3.3.

$$\begin{aligned} Rep_{ij}(t) &= E(X) \\ &= \int_0^1 p_x f(p_x|\alpha,\beta)dx \\ &= \int_0^1 p_x \frac{p_x^{\alpha-1}(1-p_x)^{\beta-1}}{B(\alpha,\beta)}dx \\ &= \frac{\alpha_t}{\alpha_t+\beta_t} \end{aligned} \tag{3.3}$$

where X is a random variable, $Rep_{ij}(t)$ represents the reputation value maintained by the subject i for the agent j, $E(X)$ is the expected value of the beta distribution, x represents the outcome of successful actions, α and β are the probability distribution function shape parameters or the levels.

Once the system is initialized, it is expected that no observations are obtained; thereby, it is important to initialize the reputation value. Authors in [153] suggest the following initial value when there is no prior knowledge:

$$\begin{aligned} Rep_{ij} &= E[Uni(0,1)] \\ &= E[Beta(1,1)] \\ &= 0.5 \end{aligned} \tag{3.4}$$

This means when no observations are available, the probability variable is uniformly distributed over the interval [0,1], and the initial reputation value is 0.5, and thereby the reputation value is evaluated using Eq. 3.5.

$$Rep_{ij} = \frac{\alpha+1}{\alpha+\beta+2} \tag{3.5}$$

The reputation value is updated by updating the beta distribution shape parameters α and β. To the best of our knowledge, all the probability distribution-based TMSs use the same updating mechanism to update the reputation value by incorporating a longevity factor to give more weight to the current observations, as shown in Eq. 3.6 and Eq. 3.7

$$\alpha_t = \lambda.\alpha_{t-1} + s(t) \tag{3.6}$$

$$\beta_t = \lambda.\beta_{t-1} + u(t) \tag{3.7}$$

where $\lambda \in [0, 1]$ is the longevity factor, $s(t)$ and $u(t)$ are the number of observations at the time unit t for both successful and unsuccessful time series, respectively. The value of λ specifies the exponential decay of the observation history. Smaller values can adopt recent behavior change better than bigger ones; however, the observation history is forgotten quickly. Therefore, the values 0.8 and 0.9 are widely used in the literature for λ [170, 175]. Fig. 3.2 shows how a beta-based reputation engine reflects the reputation value using different longevity weights for benign and malicious nodes.

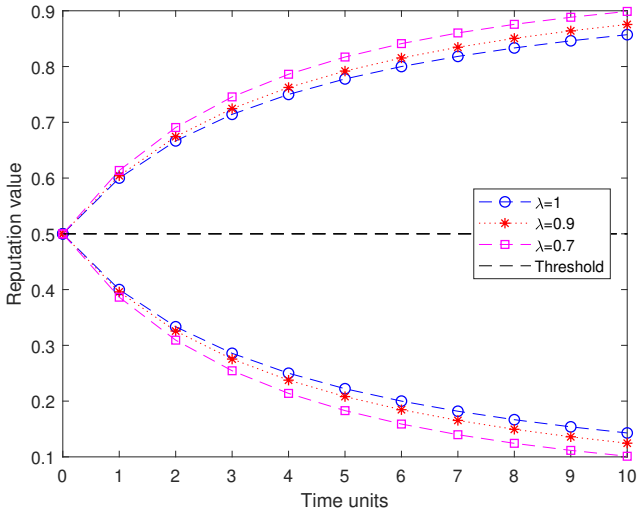


Figure 3.2: The beta-based reputation model with different longevity weights for benign and malicious nodes

3.5.3 The Proposed Method for In-Body SNs

The beta-based reputation evaluation model provides a robust basis on the theory of statistics to evaluate the trust relationship between SNs [181]. However, the beta model, in its current form, fails to detect malicious behavior effectively. It needs more time to reflect any behavior change which does not fit the critical applications of the WM-SNs. Authors in [178] compare the effectiveness of beta based reputation model with hidden Markov models and report this drawback. This issue applies to other probability distribution-based reputation models as they all use the same updating technique to update the reputation value. The traditional updating mechanism uses a single-weight exponential smoothing technique, which fails to reflect any sudden malicious behavior fast because the evaluated reputation value represents the long-term expected value of

the probability distribution. Therefore, it needs more time to detect any malicious behavior. Smart adversaries may exploit this drawback to launch complicated attacks such as on-off attacks.

In order to reduce the time required to detect malicious behavior of the beta-based reputation model and thereby enhance the detection rate, the trend of the agent's behavior has to be considered. Therefore, we consider the exponentially smoothed difference between two subsequent time units of the agent's behavior by evaluating the additive slope of the successful and unsuccessful series. Two double exponential weights λ and ω are used as opposed to a single weight in models proposed in the literature. Taking into account that the difference between two subsequent time units is always one, Eq. 3.8-3.11 show the proposed updating mechanism where the levels (beta distribution shape parameters) α_t and β_t are updated by considering the slope between the two subsequent time units, which represents the difference between observations for each series.

$$\alpha_t = \lambda(\alpha_{t-1} + b_{t-1}) + (1 - \lambda)s_t \tag{3.8}$$

$$b_t = \omega(\alpha_t - \alpha_{t-1}) + (1 - \omega)b_{t-1} \tag{3.9}$$

$$\beta_t = \lambda(\beta_{t-1} + d_{t-1}) + (1 - \lambda)u_t \tag{3.10}$$

$$d_t = \omega(\beta_t - \beta_{t-1}) + (1 - \omega)d_{t-1} \tag{3.11}$$

where b_t and d_t are the slopes at the time unit t for the successful and unsuccessful series, respectively, $\lambda \in [0, 1]$ and $\omega \in [0, 1]$ are the weighting coefficients, s_t and u_t are the number of observations at the time unit t of the successful and unsuccessful actions, respectively.

This proposed double exponential updating method shows a prompt reaction to any sudden behavior change. However, smart adversaries can take advantage of the model dynamicity to launch complicated on-off attacks. Therefore, we adopt the asymmetry principle of trust, which considers trust as a fragile thing that is hard to earn, but easy to lose [184]. Adopting this technique can defend against on-off attacks (see Section 2.5.2) and make the response to any malicious activity faster. The smoothing coefficient ω impacts the detection response speed, which is maximized when $\omega = 1$, and this means it just depends on the current change of beta levels. As incorporating the slopes into the

updating mechanism can reflect any sudden behavior change, which makes earning and losing trust identical, Algorithm 1 deals with this concept to make the trust value easy to lose and hard to earn.

Algorithm 1: Reputation updating algorithm

Input: Observations & beta shape parameters at t and $t-1$
Output: Updated shape parameters
initialization;
while *true* **do**
  if $b_{t-1} \leq 0$ && $d_{t-1} > 0$ **then**
    $\alpha_t = \lambda(\alpha_{t-1} + b_{t-1}) + s(t)$;
    $\beta_t = \lambda(\beta_{t-1} + d_{t-1}) + u(t)$;
    $b_t = \alpha_t - \alpha_{t-1}$;
    $d_t = \beta_t - \beta_{t-1}$;
  else
    $\alpha_t = \lambda.\alpha_{t-1} + s(t)$;
    $\beta_t = \lambda.\beta_{t-1} + u_(t)$;
    $b_t = \alpha_t - \alpha_{t-1}$;
    $d_t = \beta_t - \beta_{t-1}$;
  end
end

During the attack, the slope b_t maintains negative values; hence, the level α_t may accumulate negative values depending on the duration of the attack. At the same time, the level β_t, which refers to the malicious activities, develops over the attack duration with a view to make forgetting the bad behavior harder. Therefore, the trust value is evaluated using Eq. 3.12.

$$T_{ij}(t) = Rep_{ij}(t) = \begin{cases} \frac{\alpha_t}{\alpha_t + \beta_t} & for \;\; \alpha_t > 0 \\ 0 & otherwise \end{cases} \tag{3.12}$$

3.5.4 The Proposed Method for On-Body and Off-Body SNs

In this subsection, we propose our method to evaluate the trust value for on-body and off-body SNs. These SNs still have resource limitations; however, the processing capabilities are higher than the implanted ones. More importantly, replacing batteries of on-body and off-body SNs does not require surgical intervention. Therefore, a further level of protection to defend against on-off attacks is introduced to enhance the overall security.

The reputation value evaluated using beta models represents a long-term value. It reflects the accumulated long observation history. Although this feature is useful to assess the trustworthiness of the SNs, it can be exploited by adversaries to launch sophisticated attacks. Many trust management schemes in the literature adopt the sliding time window technique in order to address this security concern [145, 170, 179]. However, adopting a sliding time window has some limitations as the trust value reflects only the length of the time window; moreover, it requires more processing each time the trust value is computed.

Our proposed method for on-body and off-body SNs incorporates the short-term and long-term reputation values along with our proposed updating mechanism presented in Algorithm 1 in order to defend against on-off attacks. This protection module is only triggered when an on-off behavior is detected. The first cycle of the on-off attack is considered just a malicious activity because the on-off attack is a repeated malicious activity that can only be detected from the second cycle. Therefore, if the same behavior reoccurs, the on-off module is triggered to defend against on-off attacks. The detailed process is shown in the Algorithm 2.

where $thr1$ represents the threshold to differentiate between malicious and benign SNs, which is usually set to 0.5 in the literature [174, 153, 175, 170, 156], $thr2$ represents the expected trustworthiness that the SNs have in normal operation, $ShRep_{ij}(t)$ represents the short-term reputation value at the time unit t, and $cycle$ and $malicious$ are two variables to differentiate between sudden misbehavior and on-off attacks.

Reputation Engine Initializing

There are two points to consider upon system initialization. The first is the initial reputation value, which has been discussed earlier to overcome the case where no observations are available. The second is related to our proposed reputation engine, as computing the slope needs at least two time units of observations.

The reputation value evaluated using the Eq. 3.5 considers adding the value 1 to both α_t and β_t to ensure that the initial reputation value is 0.5 at the beginning. This formula is usually adopted by schemes that consider the reputation over a predefined time window, because it prevents the system from division by zero problem when no observations are available for the whole time window. On the other hand, adding 1 to the numerator and 2 to the denominator will influence the reputation value, especially when the number of observations is limited, as depicted in Fig. 3.3

Algorithm 2: Trust evaluation for on-body and off-body SNs

Input: Observations & beta shape parameters
Output: Trust value
initialization;
while *TRUE* **do**
  if $b_{t-1} \leq 0$ *&&* $d_{t-1} > 0$ **then**
    $\alpha_t = \lambda(\alpha_{t-1} + b_{t-1}) + s_t$;
    $\beta_t = \lambda(\beta_{t-1} + d_{t-1}) + u_t$;
    $b_t = \alpha_t - \alpha_{t-1}$;
    $d_t = \beta_t - \beta_{t-1}$;
  else
    $\alpha_t = \lambda.\alpha_{t-1} + s_t$;
    $\beta_t = \lambda.\beta_{t-1} + u_t$;
    $b_t = \alpha_t - \alpha_{t-1}$;
    $d_t = \beta_t - \beta_{t-1}$;
  end
  if $\alpha_t \leq 0$ **then**
    $Rep_t^{ij} = 0$;
  else
    $Rep_t^{ij} = \frac{\alpha_t}{\alpha_t + \beta_t}$;
  end
  if $T_{t-1}^{ij} \geq thr_1$ *&&* $Rep_t^{ij} < thr_1$ **then**
    if $malicious > 0$ **then**
      $cycle = t - malicious$;
      $malicious = 0$;
    else
      $malicious = t$;
    end
  end
  if $cycle > 0$ *&&* $Trust(t-1) < thr_2$ **then**
    $ShRep_t^{ij} = mean(T_{t-cycle:t}^{ij})$;
    $T_t^{ij} = min(ShRep_t^{ij}, Rep_t^{ij})$;
  else
    $T_t^{ij} = Rep_t^{ij}$;
    $cycle = 0$;
  end
end

Therefore, the reputation value in LTMS is evaluated using the formula in Eq. 3.3, which is simpler and does not influence the reputation value. However, to overcome the

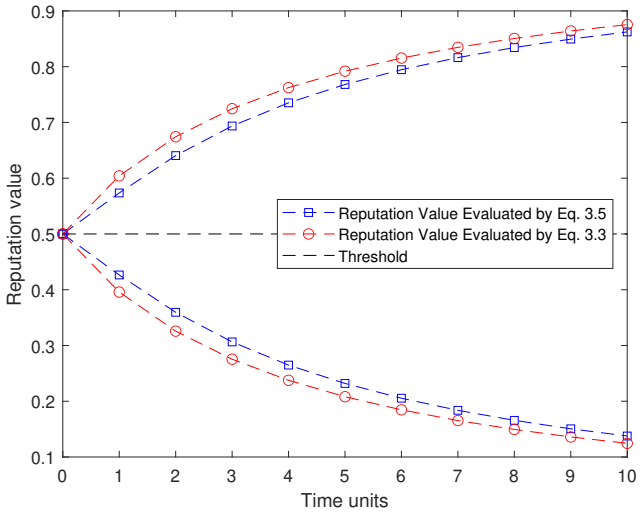


Figure 3.3: The difference between reputation values evaluated using Eq. 3.3 and Eq. 3.5 over 10 time units

aforementioned problems and assign the proper initial value, we assume a predefined time unit t_0 where the number of successful and unsuccessful actions are set to 1. Moreover, as there is no actual observations are available at the beginning, but the predefined observations in the time unit t_0, the initial slopes' values are initialized as follows:

$$b_{t_0} = d_{t_0} = 0 \tag{3.13}$$

$$b_{t_1} = s_{t_1} - s_{t_0} \tag{3.14}$$

$$d_{t_1} = u_{t_1} - u_{t_0} \tag{3.15}$$

where s_t and u_t are the number of observations in the time unit t for both successful and unsuccessful actions, respectively.

3.6 Simulation and Analysis

In this section, our proposed trust management scheme for WMSN is simulated and analyzed. The simulator NS-3.30 [185] is used to run the simulation scenarios. NS-3 simulator has been chosen because it is an open-source network simulator developed mainly for research. All SNs have the ability to forward packets, while one of them acts as a sink. Ad hoc On-Demand Distance Vector (AODV) routing protocol [186] is installed in

each SN to relay packets to the sink. However, it has been modified to simulate malicious activities by introducing new attributes to launch packet dropping attacks, which will be detailed in the next paragraph.

During the simulation, benign SNs drop received packets from others with a drop ratio of 10%, whereas the packet dropping ratio of malicious SNs is 75% unless otherwise indicated. The decision to forward or drop a packet is taken randomly for each received packet individually. The simulation consists of two phases. In the first phase, the network is initialized, and the malicious SNs behave well to increase their trustworthiness. During the second phase, they launch on-off attacks in order to disrupt the network operation and keep themselves undetected. This mimics a realistic scenario when a few nodes get compromised suddenly and start acting maliciously. The first phase of the simulation is $50s$, while the second phase is $150s$ unless otherwise indicated. The subject continues to interact with malicious agents even after they are detected in order to study the behavior of the trust schemes under on-off attacks as the trust value develops over time. The on and off periods are identical during the simulation unless otherwise indicated, and the time unit is set to 1 second.

3.6.1 Security and Efficiency Analysis

In this subsection, we present the efficiency analysis of our two methods to evaluate the trust value. The performance of the proposed methods has been contrasted with ReTrust [145] and RaRTrust [170]. These two benchmarks have been picked based on their performance under on-off attacks. As TMS schemes are vulnerable to on-off attacks, we have assessed the robustness of the direct trust evaluation methods of a set of trust management schemes proposed for WSNs [170, 174, 169, 153] and WMSNs [145, 175] under on-off attacks as discussed in our publications [66, 4]. Both ReTrust [145] and RaRTrust [170] showed the best performance in defending against on-off attacks. RaRTrust has been proposed for WSN while ReTrust has been proposed for WMSN. Their outstanding performance in detecting on-off attacks can be attributed to adopting the sliding time window technique, which can only reflect the recent behavior of SNs.

Table 3.1 shows the parameters of each scheme. It is worth mentioning that we adopt the same parameters' values as declared in their publications as they reflect the best performance. Moreover, the longevity factor λ for our scheme is set to equal or higher than other schemes as smaller values enhance the detection performance of trust management schemes and may make the comparison unfair. The exponential slope weight ω is set to 1

as discussed earlier. The expected trustworthiness parameter thr_2 is set to 0.85, indicating that benign nodes have a trust value between 0.85 and 1. On the other hand, Table 3.2 shows the simulation parameters, which have been chosen to represent a WMSN. We used nine SNs to build the network, as illustrated in Fig. 3.1, which are sufficient to reflect the behavior of the TMS. The packet size is set to 264B in accordance with IEEE 802.15.6 standard [1]. To evaluate the robustness of our design, four traffic rates have been chosen that cover low to high traffic rates, starting from $1p/s$ and up to $100p/s$. The traffic is generated based on the exponential distribution using the parameterized probability density function as shown in Eq. 3.16.

$$p(x;b) = \begin{cases} \mu e^{-\mu x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad x \in [0,b] \tag{3.16}$$

where x is a random variable represents the time gap between two consecutive packets, μ is the rate parameter, and b is the bound parameter. As the exponential distribution is theoretically unbounded, the bound b is defined to make the generated values bounded over the interval $[0,b]$.

Table 3.1: Trust Schemes Parameters

Scheme	**Parameters**
ReTrust [145]	ϕ=0.9, Time Window (TW)=6 time units
RaRTrust [170]	λ=0.8, TW=6 time units
LTMS	λ=0.9, $\omega = 1$, $thr_2 = 0.85$

Table 3.2: Simulation Parameters

Parameter	Value
Application	Poisson random traffic
Exponential transmission interval μ	1, 2, 10, 100
Packet size	264B
Routing Protocol	AODV (modified version)
Radio Range	1m
Propagation delay model	Constant speed propagation delay
Propagation loss model	Range propagation loss
Number of SNs	9
Time unit	1s
Simulation Time	200s, 400s

Trust Evaluation for In-Body SNs

In-body SNs have very tough resource limitations. They are designed to perform a specific function. For instance, a patient who has a heart problem may have an Implantable Cardioverter Defibrillator (ICD) to monitor his/her heart rate and treat any abnormal heart rhythms. Similarly, an implanted insulin pump monitors and controls the blood sugar level. These functions generate low traffic rates. Monitoring the heart rate, for example, generates a traffic rate of around 1 packet/s [172]. Therefore, existing trust management schemes must be assessed under low traffic rates. Fig. 3.4 illustrates how the trust value is developing under on-off attacks for a low traffic rate. The on-off cycle is set to 10 time units, and the traffic rate μ is set to 1 and generated exponentially. Algorithm 1, referred to as LTMS(1) is contrasted with ReTrust [145] and RaRTrust [170]. Results show that both RaRTrust and ReTrust struggle to work properly under low traffic rates. In RaRTrust, the trust evaluation process fails most of the time as just a few points appear in the figure because the first step of the trust evaluation is calculating the forwarding ratio at the current time unit, which fails due to the lack of observations at certain time units. Although ReTrust is able to evaluate the trust value during the simulation time, it fails to reflect the good behavior during the first phase when no attack occurs. In the second phase, the trust value fluctuates around the threshold without being able to reflect the bad behavior. On the other hand, our proposed algorithm LTMS(1) can reflect the actual trust value when the agent is behaving well; furthermore, when the attack is running, it shows a quick response and can keep the trust value under the

threshold most of the time.

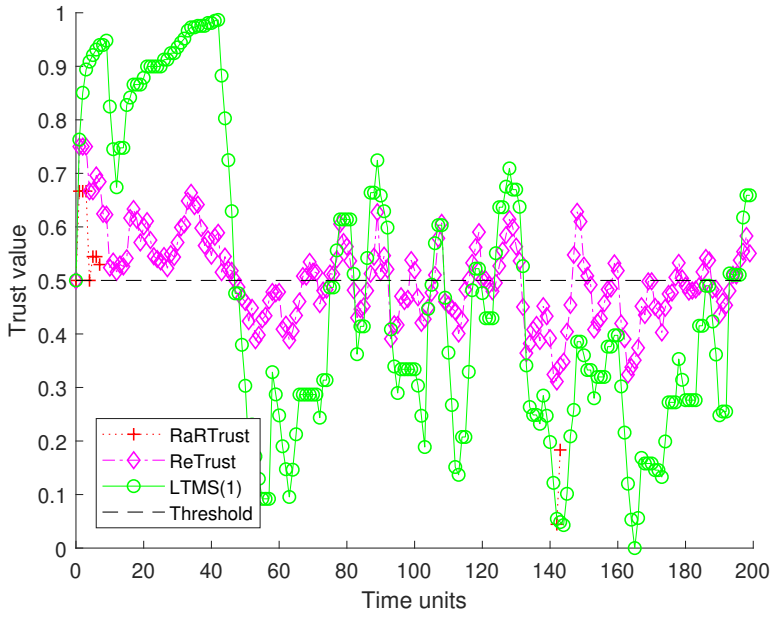


Figure 3.4: Trust evaluation for in-body SNs where the on-off attack starts at $50s$

Trust Evaluation for On-Body and Off-Body SNs

In this experiment, we evaluate our proposed method for on-body and off-body SNs. The simulation is run for 400s, where an on-off attack is launched after the first phase. The on-off cycle is set to 30 time units, and the traffic rate μ is set to 100. High traffic rate is used in this experiment because on- and off-body SNs are expected to relay packets and to ensure that the proposed method performs well for high traffic rates. After three consecutive cycles, the attack is paused to study the behavior of acquiring trust, then the attack resumes. Fig. 3.5 illustrates how trust value develops under the attack. Both ReTrust and RaRTrust demonstrate similar behavior. Both lose and earn trust easily, which causes fluctuations around the threshold during the on-off attack. On the other hand, our method LTMS(2) demonstrates an outstanding behavior. It loses trust quickly when malicious activity is detected, while it makes the trust harder to earn in the off period. The first cycle of the on-off attack is regarded as just malicious behavior as shown in the period between 50 and 100. Therefore, from the second cycle on as shown after the time unit 100, the on-off attack is detected and the on-off module in the LTMS(2) algorithm is triggered to make earning trust during the off period harder, as

shown between the time units 100 and 200. The on-off attack has been stopped between the time units 200 and 250 to illustrates how LTMS(2) makes acquiring trust harder for malicious nodes. Moreover, when the attack reoccurs as shown after the time unit 250, LTMS(2) maintains the same behavior in detecting on-off attacks.

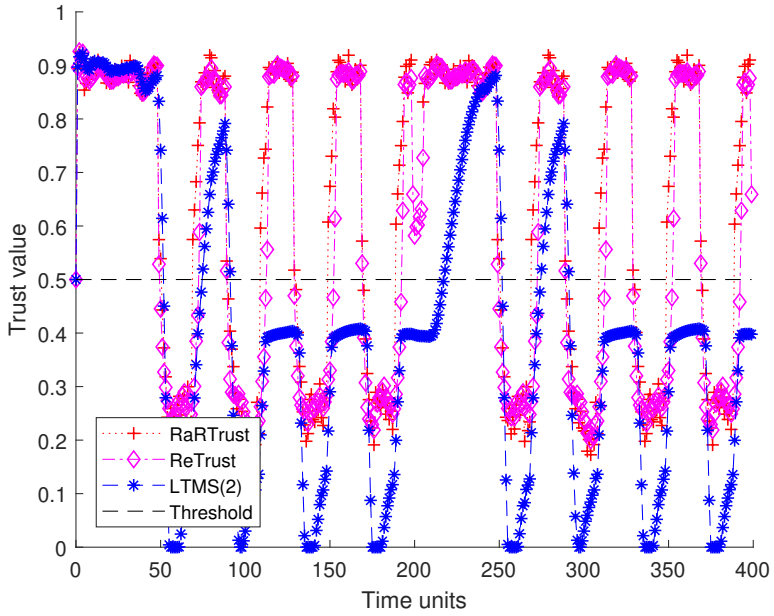


Figure 3.5: Trust evaluation for on-body and off-body SNs where the on-off attack starts at $50s$ and re-occurs again at $250s$

3.6.2 On-Off Performance Metric

In order to assess the effectiveness of trust management schemes under on-off attacks, we introduce the on-off Attack Detection Metric (ADM). ADM is defined in Eq. 3.17 as the ratio of the detection time to the on-off attack time.

$$ADM = \frac{|DT|}{|AT|} \tag{3.17}$$

where $|DT|$ denotes the number of the time units when the attack is detected, and $|AT|$ denotes the total number of time units of the on-off attack.

This metric is able to reflect the robustness of the trust management schemes under on-off attacks. The performance of our proposed methods are evaluated in three different scenarios to ensure that the proposed methods are able detect sophisticated dropping

patterns under different operation conditions. Firstly, we evaluated the proposed schemes for different traffic rates ranging from low to high as some SNs generate very low traffic rate around $1p/s$. Secondly, the robustness of the proposed methods is evaluated for variable drop rates. Thirdly, non-identical on-off attacks with different cycles have been launched to ensure that the proposed methods are able to detect complicated on-off attacks.

Variable Traffic Rates

Two sets of traffic rates are chosen. The first contains low traffic rates ($\mu = 1$, $\mu = 2$), which represents the traffic of the in-body SNs, whereas the second set ($\mu = 10$, $\mu = 100$) is chosen for medium and high traffic rates. Fig. 3.6a - 3.6d show the detection rate performance of the aforementioned schemes for different traffic rates and different on-off attack cycles.

For low traffic rates, RaRTrust shows the lowest detection rates for all on-off attack cycles. It struggles to detect the on-off attacks with a detection rate around 0 for $\mu = 1$, while ReTrust shows better performance compared with RaRTrust. It detects around 65% of the on-off attacks when the on-off cycle is 10 time units. By increasing the duration of the on-off attack cycle, the performance of ReTrust decreases to around 60%. On the other hand, our proposed methods demonstrate superior performance compared with ReTrust and RaRTrust, with detection rates up to 80% and 93% for LTMS(1) and LTMS(2), respectively.

For medium and high traffic rates, RaRTrust starts to defend against on-off attacks with a detection rate of around 50% for $\mu = 10$ and around 40% for $\mu = 100$. In contrast with RaRTrust, ReTrust shows better performance with a detection rate that starts at around 76% and decreases to 60% for $\mu = 10$, and starts at around 86% and decreases to reach 62% for $\mu = 100$. On the other hand, our method LTMS(2) shows the best performance in detecting the on-off attacks. For $\mu = 100$, it starts at just below 98% for the on-off cycle 10 time units and reaches around 88% for the on-off cycle 40 time units. For $\mu = 10$, it starts at just below 97% and reaches 80% for the on-off cycle 40 time units. Finally, it is worth mentioning that our lightweight method for in-body SNs LTMS(1) shows better results than both ReTrust and RaRTrust in detecting on-off attacks for medium and high traffic rates with a significantly lower processing overhead.

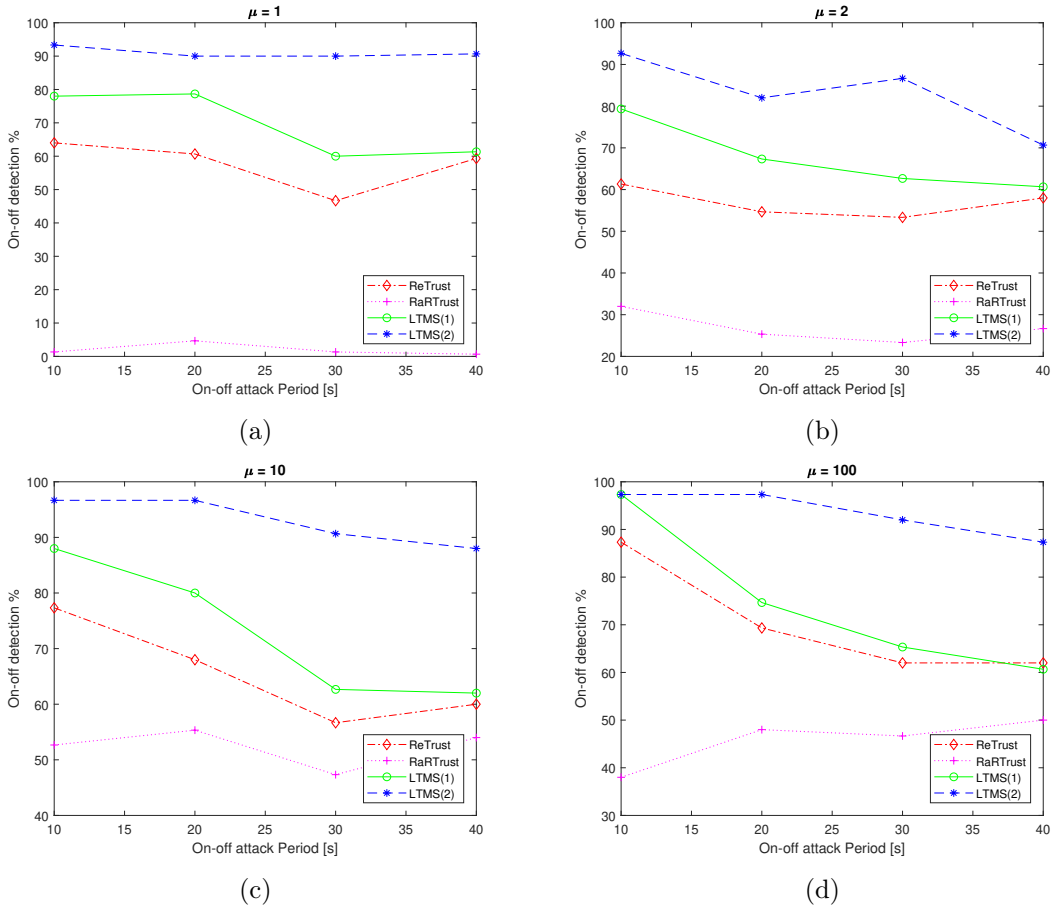


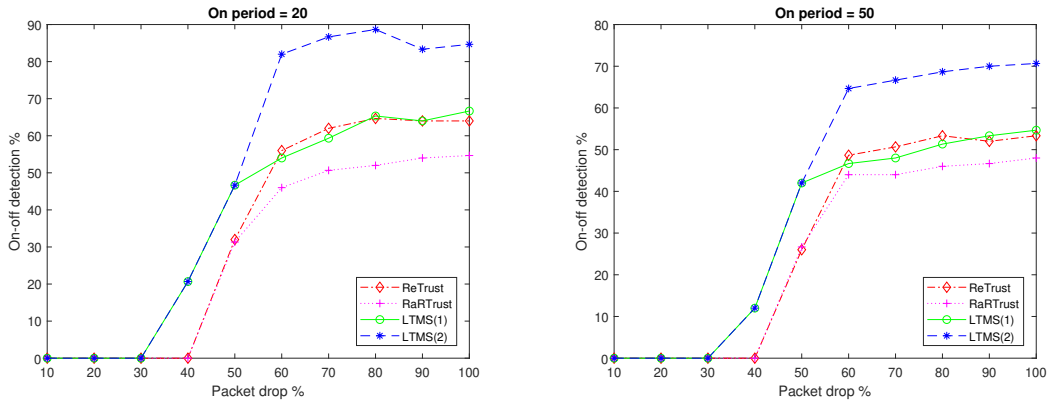
Figure 3.6: The detection performance for variable traffic rates

Variable Drop Rates

In this experiment, we evaluate the attack detection performance for different packet dropping rates. The drop rate varies from 10% to 100% during the on period instead of the previous fixed drop rate of 75%. Fig. 3.7a and 3.7b illustrate the detection rates for two different on periods 20 and 50 time units, where the traffic rate μ is set to 100. ReTrust and RaRTrust detect attacks starting from the drop rate of 40%. Between 40% and 50%, ReTrust and RaRTrust show identical results, then ReTrust overcomes RARTrust in the detection rate.

On the other hand, our proposed methods are able to detect attacks starting from 30% drop rate thanks to its novel updating mechanism, making it more sensitive to dropping attacks. LTMS(2) shows superior performance comparatively, while LTMS(1) shows a

close performance to ReTrust between 50% and 100%.



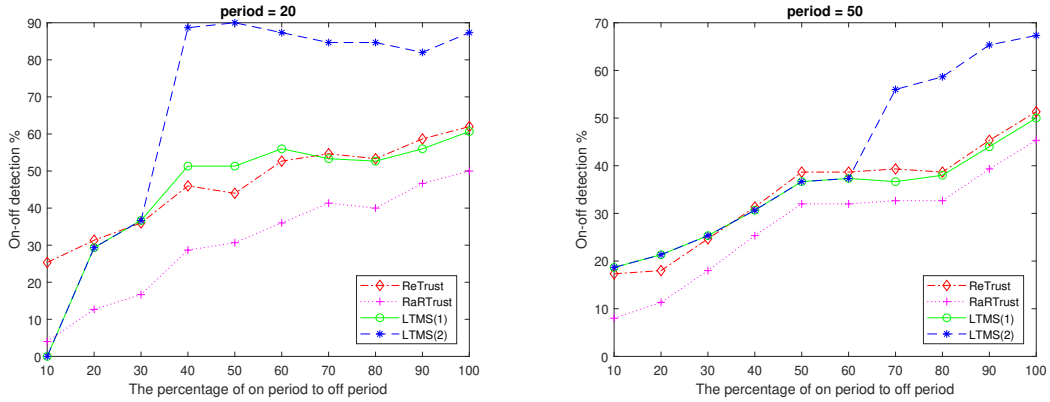
(a) (b)

Figure 3.7: The detection performance for variable drop rates

Non-Identical Periods

In this experiment, more sophisticated on-off attacks are launched by varying the on and off periods. Obviously, it is harder to detect on-off attacks when the on period is less than the off period. Hence, the on period is set to be a ratio of the off period ranging from 10% to 100%. The traffic rate μ is set to 100. Two on periods 20 and 50 time units are used to evaluate the performance. Fig. 3.8a and 3.8b show the detection performance of the aforementioned trust schemes when the on period is 20 and 50 time units. For on period equals to 10% of the off period, both ReTrust and RaRTrust show a detection rate greater than 0. However, from our point of view, two sequential time units of bad behavior are not enough to destroy the earned trust. ReTrust and RaRTrust adopt a sliding time window to calculate the trust; meaning, they adopt the most recent changes regardless of the agent's history, as illustrated in Fig. 3.4 and 3.5. By increasing the ratio of the on period, the performance of both ReTrust and RaRTrust is enhanced for both on periods; however, ReTrust shows better performance.

On the other hand, LTMS(1) and LTMS(2) detect attacks starting from 20% and 10% for the on periods 20 and 50, respectively. LTMS(1) shows a close performance to ReTrust. However, LTMS(2) shows prominent performance comparatively after 30% and 60% for the on period 20 and 50, respectively.



(a) (b)

Figure 3.8: The detection performance for different on-off ratios

3.6.3 Computational Overhead

In this subsection, we compare the average processing time consumed by each of the trust management schemes using the MATLAB platform. The test is carried out on Intel Core i5-8500T processor at 2.1GHz and 8GB RAM using the data sets generated by our simulation scenarios. Fig. 3.9 illustrates the average processing time of ReTrust, RaRTrust, LTMS(1) and LTMS(2) under on-off attack. RaRTrust consumes the highest average processing time of $3.2 \times 10^{-4} s$, while ReTrust shows a better processing overhead compared with RaRTrust as it consumes around $2.74 \times 10^{-4} s$. On the other hand, our method LTMS(1) for in-body SNs consumes the lowest processing time among all trust schemes. It consumes $0.85 \times 10^{-4} s$, which saves around 73% and 69% of the processing time of RaRTrust and ReTrust, respectively. Moreover, our method for on-body and off-body SNs consumes around $1.7 \times 10^{-4} s$ average processing time, which saves around 47% and 38% of the processing time of RaRTrust and ReTrust, respectively.

3.7 Conclusion

Security concerns prevent the widespread adoption of the WMSN advancements as dropping attacks may disrupt the operation of the routing protocols, which could have severe consequences on patients' health. Therefore, there is a need for an effective trust management system to reinforce the security of WMSN data delivery with a minimum resource footprint. In this chapter, we proposed a trust evaluation model for WMSN. The proposed scheme uses novel updating and evaluating mechanisms. LTMS is a lightweight,

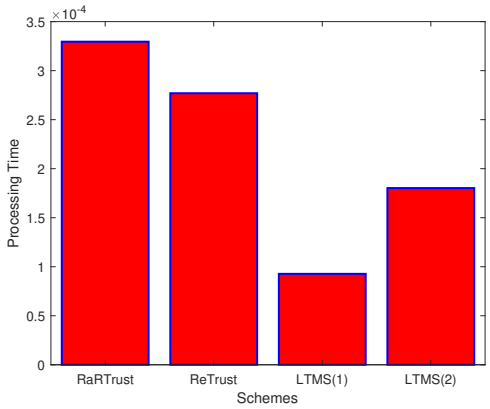


Figure 3.9: The average processing time

attack-resistant trust scheme for in-body, on-body, and off-body SNs. The experimental results show that available TMSs are not designed to work under low traffic rates and are vulnerable to on-off attacks. On the other hand, our proposed methods outperform the state-of-the-art trust management schemes while preserving resources, making them a suitable candidate to meet WMSN security requirements.

Chapter 4

Lightweight and Reliable Routing Using Q-Learning

Wireless medical sensor networks (WMSNs) offer innovative healthcare applications that improve patients' quality of life, provide timely monitoring tools for physicians, and support national healthcare systems. However, despite these benefits, widespread adoption of WMSN advancements is still hampered by security concerns and limitations of routing protocols. In addition, routing in WMSNs is a challenging task due to the fact that existing routing proposals overlook some WMSN requirements. In this chapter, we propose a lightweight and reliable multi-agent reinforcement learning-based routing protocol (RRP) to overcome these challenges. RRP is a lightweight attacks-resistant routing protocol designed to meet the unique requirements of WMSN. It combines a novel RL model to reduce resource consumption with our effective trust management system, presented in Chapter 3, to defend against various packet-dropping attacks. Experimental results prove the lightweightness of RRP and its robustness against blackhole, selective forwarding, sinkhole and complicated on-off attacks.

The main findings of this chapter has been accepted to publish in the IEEE 20th Annual Consumer Communications & Networking Conference (CCNC) 2023. Also, it has been submitted to IEEE Internet of Things Journal for review.

4.1 Introduction

The wireless nature and the critical applications provided by WMSN make it vulnerable to various security attacks and misconduct activities, the most important of which are packet-dropping attacks. These kinds of attacks are called internal attacks because they are launched by the Sensor Nodes (SNs) themselves for different reasons. For instance, if an SN gets compromised and starts dropping packets, it disrupts the overall network operations. Another example is when an SN acts selfishly and stops relaying packets for others to save power or gain extra resources unfairly. In both cases, the consequences would be detrimental and could endanger the patient's life. Moreover, many dropping attacks discussed in the literature have different characteristics and dropping patterns, making them difficult to detect and defeat, such as selective forwarding attacks [187], blackhole attacks [188], and sinkhole attacks [189].

In addition to the security concerns inherited from Wireless Sensor Networks (WSNs), WMSN has additional unique characteristics, such as resource constraints, critical applications, network topology, and low traffic rates. While routing in WSN is still challenging, with much research being put forward constantly to produce an efficient routing protocol [190], it is more challenging in WMSN due to its unique characteristics. Reinforcement Learning (RL) based routing protocols have been introduced in the literature to address the routing problem in WSN [38, 39, 40]. Although this approach allows SNs to learn the optimal path to the destination, it has a few limitations. To the best of our knowledge, the learning agent in all these proposed schemes has to receive a reward for each sent/forwarded packet and then update its estimation to find the optimal path for future packets. This mechanism is voracious in terms of resource consumption and may not fit the resource-constrained SNs of WMSN. Moreover, choosing the lowest cost path does not guarantee delivery reliability as the chosen path may contain one or more malicious nodes. Therefore, our proposed RRP uses a novel design to produce a lightweight and efficient routing protocol. Moreover, an effective Trust Management (TM) scheme is integrated with the RRP to ensure high delivery reliability. The reward function has been re-defined as a punishment function based on the trustworthiness of potential routes for reliable delivery.

4.2 Contribution

The main contribution of this chapter is fourfold. First, the unique requirements of designing an efficient and reliable routing protocol for WMSN are specified. Second,

reformulating the RL model in a resource-conservative way to overcome the resource limitations of WMSNs. Third, an efficient, lightweight, and reliable RL-based routing protocol combined with a trust management scheme is proposed. Fourth, a comprehensive analysis is carried out to prove the merit of our routing protocol against well-known dropping attacks.

4.3 Related Work

Routing is quite a challenging task in WMSN. The main challenge is to achieve reliable data delivery with minimum resource consumption in order to ensure high longevity of network operation [191]. Various routing protocols have been proposed in the literature to ensure reliable data transfer in WSN using different metrics and algorithms. However, only a few schemes targeted WMSN, such as [192, 44, 45]. Moreover, WMSN has unique characteristics and requirements, making inherited routing protocols from WSN not necessarily fit WMSN. Therefore, there is still an imperative research gap to design a routing protocol that fits WMSN and meets its requirements.

Reinforcement learning has been widely used in the literature to find the optimal path with minimum overhead. Researchers use different metrics to achieve this goal, such as delivery latency, residual energy, and geographical distance [193]. However, this kind of metrics cannot deal with the free will of the other nodes. Relay nodes could get compromised or act selfishly and hence stop relaying packets for other nodes, which results in detrimental consequences. Therefore, there is a need to incorporate a security measure to avoid malicious paths. TMS provides an effective and robust measure to evaluate the trustworthiness of other nodes. To the best of our knowledge, only two schemes [42, 45] are proposed in the literature that combine a TM scheme with a Q-learning routing model. Authors in [42] provide a secure, lightweight routing scheme for WSN. However, it is unclear how the trust relationship is evaluated, which makes this scheme not reproducible due to missing details. Authors in [45] proposed QRT, a routing protocol designed for non-cooperative biomedical mobile wireless sensor networks. It has been proposed as an extension to RL-QRP [44] to deal with various kinds of misbehaving activities. The authors adopted the beta distribution trust scheme and integrated it with the Q-learning routing engine to produce a reliable routing protocol. However, proving its merit needs further investigation. Both ESRQ and QRT have not been thoroughly evaluated under different dropping attacks, especially on-off attacks. Moreover, all the proposed RL-based routing protocols in the literature use the same traditional RL model,

which is a resource-consuming model and is not suitable for deployment on resource constrained SNs.

4.4 Protocol Design

In this section, the proposed routing protocol for WMSN is discussed in detail. The design starts by presenting the network and threat models, which leads to specifying the protocol designing requirements. Our novel RL model to produce a lightweight routing protocol is then discussed. The delivery reliability is achieved by integrating our proposed TM scheme into the routing decision engine. Moreover, two updating mechanisms have been proposed to accelerate the algorithm convergence as well as conserve resources.

4.4.1 Network Model

WMSN consists of a set of bio-sensor nodes that could be placed on the body surface, inside the body, or off the body. These SNs have the ability to sense the body's physiological signals, such as body temperature, glucose levels, Electrocardiogram (ECG), and pulse rate. However, SNs have strict resource limitations that impose further constraints in adopting security countermeasures. Therefore, lightweight countermeasures and protocols are essential to extend the battery life and avoid unnecessary surgery complications. All sensed information is forwarded to the sink node, which in turn forwards them to the remote medical server where physicians can monitor, analyze and even intervene when necessary.

Field hospitals are temporary hospitals set up due to civil emergencies, such as battlefields, disease outbreaks, and pandemics. For example, many field hospitals have been established in many parts of the world during the ongoing COVID-19 pandemic, especially in developing countries. In our experiments, the topology of a wireless medical sensor network of a field hospital ward is adopted. Fig. 4.1, shows the simulated ward in our experiments, which is $50m \times 10m$ where patient beds are distributed in an efficient way to save physical space and provide an adequate space to care at the same time. A maximum number of 64 SNs can be accommodated in this medical unit in compliance with IEEE 802.15.6 standard [1]. The network topology is a multi-hop star topology where SNs sense various bio-signals and forward them to the sink node. The communication range of the SNs is $5m$; hence, SNs relay frames for other adjacent nodes. Therefore, an efficient, lightweight, and reliable routing protocol is required to forward the frames from the sensing units to the sink node, which in turn forwards them to the

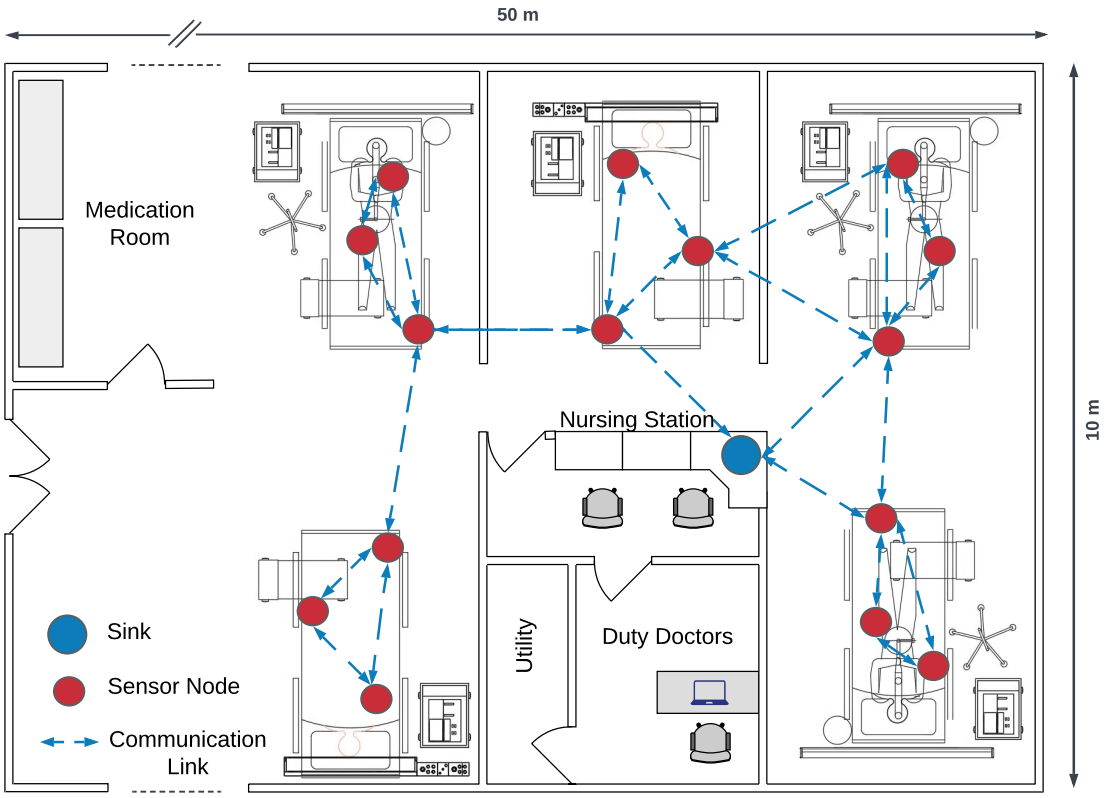


Figure 4.1: Network Model

medical server. It worth mentioning that the SNs will be randomly distributed for the simulation scenarios. However, this will not affect the performance of RRP as the action values are estimated using a model-free RL algorithm.

4.4.2 Threat Model

Due to the sensitive nature of the WMSN applications and the broadcast nature of the wireless communication, many potential threats may disrupt the network operation and endanger the patients' lives. Threats can be classified into internal and external. External threats could be defeated by deploying cryptographic security measures, such as authentication and encryption. Our proposed ecosystem assumes that secure mutual authentication is achieved and security keys are established. On the other hand, internal threats are difficult to defeat as they could be launched by legitimate nodes that have successfully got authenticated and may have a copy of the security keys. Therefore, this work aims to demonstrate the effectiveness of our RRP against packet dropping attacks,

one of the devastating internal threats on WMSN.

Packet-dropping attacks are regarded as one of the most devastating internal attacks because of their consequences on the patient's life. For instance, a malicious node could drop a command sent by a physician to an insulin pump to release the insulin dose into the bloodstream. In addition, dropping could occur due to malicious activities like when a node got compromised, selfish behavior when a node acts selfishly with a view to saving resources, or when packets pass through overloaded nodes. Adversaries could launch different kinds of dropping attacks or may change the dropping patterns with a view to keeping themselves undetected. RRP protocol is evaluated for various kinds of dropping attacks with different parameter settings, such as blackhole, sinkhole, selective forwarding, and on-off attacks, which will be discussed in detail in Section 4.5.

4.4.3 Design Requirements

Various objectives have been considered when designing RRP. These objectives include efficiency, lightweightness, scalability, and resiliency.

Efficiency is the first objective of designing a routing protocol. Ensuring a high packet delivery ratio is a must for any routing protocol. However, choosing the optimal path between the sender and receiver determines the routing protocol's efficiency, which is a crucial requirement for resource-constrained devices, such as SNs. The lowest cost path must always be chosen to ensure high efficient routing protocol. RF activities, especially transmission (TX), constitute around 80% of the consumed energy [194]. In order to reduce the consumed energy, SNs must always choose the shortest path in order to reduce the number of transmissions. Therefore, RRP has been designed to always choose the shortest reliable path regardless of the network size, nodes deployment, or traffic rate.

Lightweightness is a key requirement to fit the strict resource constraints of the SNs. All proposed Q-learning-based routing protocols in the literature consider transmitting one packet as a complete action, which calls for updating the Q-table for each sent or forwarded packet [46, 40, 44, 45]. This method is a resource-consuming process, particularly when more packets are generated or forwarded. Therefore, in RRP, the RL model has been reformulated to consume less memory and processing resources.

Scalability is another requirement. In a multi-agent environment, each agent has to consider the actions of other agents, which causes a scalability problem when the number

of agents increases as the action space grows exponentially [195]. Moreover, agents in a networked environment suffer from the partial observability problem as they do not have a full view of the network. Therefore, decentralized learning with a networked agent approach [196] was adopted in RRP to enable the learning agents to collaborate with their neighbors by sharing information. This approach is regarded as a solution to the poor scalability of fully centralized learning, and centralized training with decentralized execution approaches [195]. In addition, RRP has been evaluated for variable traffic rates and the maximum number of SNs in the network as defined in IEEE 802.15.6 [1].

Attack resilience is the most challenging task in designing a reliable routing protocol for WMSN. Dropping attacks could be catastrophic not only for the network operation but also for the patients. Authors in [197], investigated the performance of Routing Protocol for Low-Power and Lossy Networks (RPL), which is one of the candidate routing protocols for low-power and lossy networks, under blackhole attacks. The results showed a loss of a large amount of data. Therefore, RRP has been designed to resist all kinds of known dropping attacks. Moreover, it is also resilient to route poisoning attacks.

4.4.4 Multi Agent Reinforcement Learning

Reinforcement Learning (RL) is an area of machine learning that focuses on how intelligent agents interact with an environment through a series of state-action pairs to maximize the cumulative rewards. In Multi-Agent Reinforcement Learning (MARL), many agents interact with a mutual environment and with each other to achieve a particular goal [195]. This interaction could be a collaboration to accomplish a common task, a competition to accomplish a self-goal, or a mix of both, like when members of two teams of a game collaborate with each other and at the same time compete with the other team.

In the traditional RL model, as illustrated in Fig. 4.2, at each time step t, the RL agent in an environment's state $s_t \in \mathbb{S}$ chooses an action $a_t \in \mathbb{A}$, which causes the environment to move to state $s_{t+1} \in \mathbb{S}$ and the agent to receive a reward $r_{t+1} \in \mathbb{R}$.

In routing applications, the agent learns a routing policy that chooses the optimal path to the destination by experimenting different actions and gathering evidence from the environment. The learning process in such a case must be online and continual due to the dynamicity of the network. The learned routing policy specifies the optimal adjacent node for each agent to forward its frames to. This routing policy is constantly updated to reflect any change in the network. Moreover, as the network represents the environment

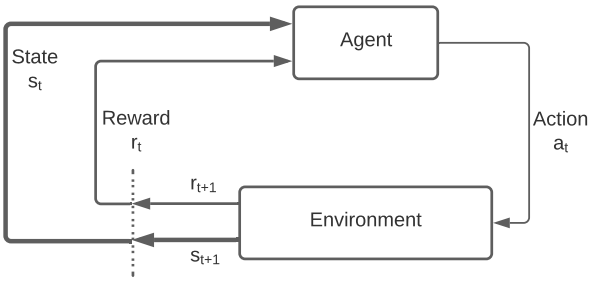


Figure 4.2: Traditional RL Model

in the RL model, a model-free RL algorithm is needed to deal with any network size or nodes' locations. Therefore, Q-learning has been used in RRP, which is an off-policy, value-based, model-free reinforcement learning algorithm to evaluate the value of an action in a particular state [39]. Each agent maintains a Q-values table of $|\mathbb{S}| \times |\mathbb{A}|$ represents the expected long-term rewards if the agent takes the action a_t at the state s_t.

4.4.5 The Proposed Synchronous Q-Routing Model

With the aforementioned design requirements in mind, RRP is built using the Q-learning algorithm, incorporating the proposed trust management scheme in Chapter 3 to ensure an efficient, lightweight, and reliable routing protocol. The learning agent is modelled as 3-tuple $(\mathbb{S}, \mathbb{A}, \mathbb{R})$. WMSN network represents the environment $\mathbb{E}$, which includes SNs that exchange messages where one of them acts as a sink S. Each state $s \in \mathbb{S}$ represents a SN. The action $a \in \mathbb{A}$ is defined as selecting the next forwarder to relay packets to a destination. The learning agent receives a reward $r_{t+1} \in \mathbb{R}$ for each action a_t.

RRP defines $Q^i_{t+1}(s^i_t, a^i_t)$, which is the updated Q value of node i, given the state s^i_t and the action a^i_t, as the estimated future rewards. Each learning agent maintains a Q-table, which gets updated once the agent performs an action a_t and observes the reward r_{t+1} as in Eq. 4.1.

$$Q^i_{t+1}(s^i_t, a^i_t) \leftarrow (1-\eta)Q^i_t(s^i_t, a^i_t) + \eta[r^i_{t+1}(s^i_{t+1}) + \gamma \underset{a \in A}{max} Q^i_t(s^i_{t+1}, a^i_t)] \tag{4.1}$$

where $\eta \in [0, 1]$ is the learning rate where small values of it cause long learning time and large values may cause oscillations, $\gamma \in [0, 1]$ is the discount factor for the future rewards where small values of it make the agent myopic and cares more about the immediate

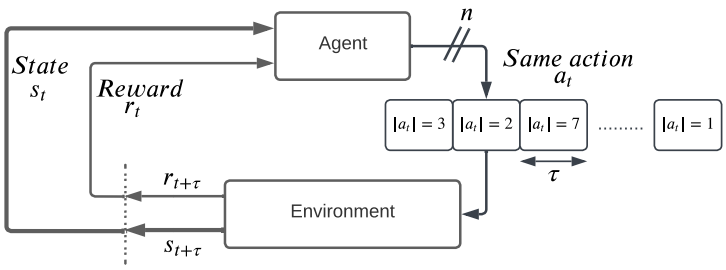


Figure 4.3: Graphical representation of the proposed RL model

rewards. In order to ensure reliable forwarding, trust is incorporated in estimating the reward. This makes the learning agent chooses the optimal reliable path. Moreover, the reward calculation is defined as a punishment to force the learning agent to choose the shortest path to the destination, as shown in Eq. 4.2.

$$r^i_{t+1}(s^i_{t+1}, j) = \begin{cases} -(1 - T^{ij}_t) & if \quad O^{ij}_t \neq \{\phi\} \\ -(1 - T^{ij}_{t-\delta}) & if \quad O^{ij}_t = \{\phi\} \wedge |O^{ij}| > \epsilon \\ 0 & Otherwise \end{cases} \tag{4.2}$$

where $r^i_{t+1}(s^i_{t+1}, j)$ is the new reward received by node i which chose node j as a forwarder at the end of the time unit t, T^{ij}_t is the trust value maintained by node i for node j at time unit t, δ is a time lag used to get the last evaluated trust value, O^{ij}_t is the observations maintained by node i for node j at time unit t, ϵ is the threshold to specify the minimum required evidence. The trust value T^{ij}_t is computed using Algorithm 2 as detailed in Chapter 3.

The proposed RL Model

To the best of our knowledge, RRP proposes the first RL model that uses the time window technique to reduce the computational overhead of the traditional RL model, as shown in Fig. 4.2. In the traditional RL model, the learning agent updates its Q table for each sent or forwarded packet using Eq. 4.1 and 4.2. This excessive updating process depletes the available limited resources, especially when the network size or the traffic rates increases. For instance, the authors in [198] used the microcontroller MSP430F1611 to build an implantable pacemaker. The used microcontroller operates at $8MHz$ with only $48KB$ Flash memory and $10KB$ Static Random Access Memory (SRAM). This stringent resource limitation requires resource-conservative methods to

reduce the computational overhead to the minimum.

Therefore, in RRP, we reformulated the existing RL model, assuming that the network will be static for a short period, which is an acceptable assumption as nodes could be regarded as stationary for a short interval. This assumption allows the learning agent to perform the same action multiple times during a short period of time before receiving the corresponding reward. Adopting this method significantly reduces the computational overhead by periodically updating the Q tables. The actions and rewards are re-defined in the proposed RL model in which the agent performs the same action a_t during the time unit t and gets its reward $r_{t+\tau}$ at the end of the time unit at $t+\tau$ as illustrated in Fig. 4.3. In the traditional RL model, the learning agent needs to observe the reward and update its Q-table for each packet, while RRP evaluates the reward and updates the Q-table after a defined time unit τ in order to reduce the computational overhead. This proposed method is referred to as synchronous updating. Moreover, asynchronous updating is also used in RRP to help the algorithm to converge swiftly, which will be elaborated further in Section 4.4.6.

The routing task must be achieved in a distributed manner as no agent has a full view of the network states. Therefore, RRP uses decentralized learning where the RL agents exchange their best Q values with their neighbors as detailed in Algorithm 3. The exchanged values are then used to update the Q-table and determine the best forwarder to the destination. Once the following action is taken, it changes the environment states, requiring periodic updates. Actions should not be greedily selected all the time for two reasons. First, routing is an online continual learning task. Second, exploiting the best action prevent the algorithm from converging to the global optimum. Therefore, $\varepsilon-$greedy strategy [199] is used to explore the environment with a probability of θ and exploit the best action with a probability of $(1-\theta)$. During the exploration phase, a random action $a_t^i \in \mathbb{A}$ is selected to search for possible alternative paths. At the beginning, RRP has no knowledge about the environment; hence the future rewards are initialized to zero for each neighbor $n^i \in N_t^i$, which is more realistic and requires no additional hardware or pre-configuration like those introduced in [45, 44], where the authors used positioning information.

4.4.6 Updating Methods

In the RRP routing protocol, two kinds of Q table updating methods are used in order to reduce resource consumption and enhance the algorithm convergence, as shown in

Algorithm 3: RRP protocol for making routing decisions

Input:
The reward: $r^i_{t+1}(s^i_{t+1}, j)$
The Q table: Q_t
The trust table: T_t
Output: The optimal next hop
initialization:

$$Q^i_0(n^i \in N^i_t) = \begin{cases} 0 & if \quad n^i \neq S \\ 1 & if \quad n^i = S \end{cases}$$

$T^i_0(n^i \in N^i_t) = 0.5$

$$a^i_1 = \begin{cases} S & if \quad S \in N^i \\ n^i & | \quad n^i \in N^i \end{cases}$$

while *TRUE* **do**
  $Wait \quad \tau$
  $Broadcast \quad max(Q^i_t)$
  $\forall j \in N^i$, $update(Q^{ij}_t)$ using Eq. 4.1
  if $\varepsilon - greedy > \theta$ **then**
    $a^i_{t+1} \leftarrow n^i_t \mid n^i_t \in N^i_t$
  else
    $a^i_{t+1} \leftarrow \underset{n^i_t \in N^i_t}{argmax}\ Q^i_t(s^i_t, a^i_t)$
  end
end

Algorithm 4. A synchronous update is used to update the Q table at the end of each time unit with a view to reducing the processing overhead. As the action in our model consists of multiple sub-actions on a predefined time unit, the learning agent performs the same sub-action multiple times during the period τ, which means all packets will be forwarded to the same next hop. Meanwhile, the agent is observing the behavior of its next hop to evaluate its trustworthiness. By the end of the time unit, the agent is able to evaluate the trust value at time t and gets its reward $r^i_{t+1}(s^i_{t+1})$. Each agent broadcasts its best estimation to adjacent nodes periodically. These broadcasted estimations are then used to update the Q table using the gained reward as in Eq. 4.1. However, as each agent only forward packets to one node during the time unit, it will not get rewards for other adjacent nodes, but it could receive an updated estimation from them. For instance, node i has $a^i_t = j$ at time t and receives updates from nodes j and k. In this

case, RRP updates the Q value of node j using Eq. 4.1 and checks how certain it is about node k by checking the number of recent observations. If node i has adequate observations about node k, it will use the most recent reward $r^i_{t-\delta}(s^i_{t-\delta}, k)$ to update the Q^{ik}_t. Otherwise, it will ignore the received estimation and keep the Q value unchanged. This technique immunizes RRP from adopting fake second-hand information without being certain enough about the sender's trustworthiness. Moreover, it allows the protocol to respond quickly to network dynamicity.

On the other hand, although the proposed synchronous updating is very resource-efficient, as presented in the next section, it could be slow to converge and may need more learning time as the learning agent could keep forwarding packets to the wrong path for the whole time unit. This usually happens if loops occur when the learning agent is exploring the network. Unlike traditional learning model where the learning agent risks losing one packet for each exploring step, the synchronous updating model could lose more packets because it keeps forwarding to one next-hop during one time unit. Therefore, RRP introduces a loop detection and avoiding algorithm as shown in Algorithm 5. Once a loop is detected, or there is a possibility for a loop to occur, the asynchronous update is called as shown in Algorithm 4. The updating process penalizes the corresponding Q value, which allows the protocol to choose another promising next hop. This technique enables RRP to perform efficiently and converge swiftly. Table 4.1 defines all the used symbols in Chapters 4 and 5.

4.5 Evaluation and Performance Results

This section simulates and analyzes the RRP routing protocol. Various simulation scenarios have been considered using different parameters setting and under different dropping attacks.

4.5.1 Experimental Setup

A WMSN of 64 SNs has been adopted to comply with IEEE 802.15.6 [1]. The SNs have been distributed randomly in an area of $50m \times 10m$ mimicking a ward in a field hospital as shown in Fig. 4.1. One SN acts as a sink while other nodes have the ability to relay frames for other SNs. The traffic is generated using the exponential probability density function as shown in Eq. 4.3. It worth mentioning that the trust relationship is evaluated using Algorithm 2 for on and off body SNs. Therefore, all the experiments in Chapters 4 and 5 are mimicking on and off body SNs.

Algorithm 4: Synchronous and asynchronous Q table updating

Input:
The Q table: Q_t^i
The reward: $r_{t+1}^i(s_{t+1}^i, j)$
The trust table: T_t
Output: Updated Q Table: Q_{t+1}^i
if *Synchronous Update* **then**
  foreach $j \in N_t^i$ **do**
    if $j == a_t^i$ **then**
      update Q_t^{ij} using $r_{t+1}^i(s_{t+1}^i, j)$
    else
      if $|O^{ij}| > \epsilon$ **then**
        update Q_t^{ij} using recent $r_{t-\delta}^i(s_{t-\delta}^i, j)$
      else
        $Q_{t+1}^{ij} \leftarrow Q_t^{ij}$
      end
    end
  end
end
if *Asynchronous Update* **then**
  if $\eta == 1$ **then**
    $r_{t+1}^i(s_{t+1}^i, j) = -e^{-\mu}(1 - T_t^{ij})$
  else
    $r_{t+1}^i(s_{t+1}^i, j) = -(1 - T_t^{ij})$
  end
  if $RQ_{t-1}^i(s_{t-1}^i, j)$ **then**
    update Q_t^{ij} using r_{t+1}^i and $RQ_{t-1}^i(s_{t-1}^i, j)$
  else
    $Q_{t+1}^i(s_t^i, a_t^i = n_j) \leftarrow Q_t^{ij} - \zeta$
  end
  $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
end

$$p(x;\mu) = \begin{cases} \mu e^{-\mu x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{4.3}$$

where μ is the rate parameter.

Algorithm 5: Loop processing

Input: A packet to forward: P_t^{sd}
Output: Updated Routing
while *TRUE* **do**
  if $\forall\, i \in \mathbb{N}$ *receives* $P_{t+\delta}^{id}$ **then**
    Asynchronous Q table update as in Algorithm 4
    $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
    Update P_t^{id}
    Send P_t^{id}
  end
  if $\forall\, i \in \mathbb{N}$ *receives* $P_t^{jd} \wedge a_t^i = j$ **then**
    Asynchronous Q table update as in Algorithm 4
    $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
    Forward P_t^{jd}
  end
end

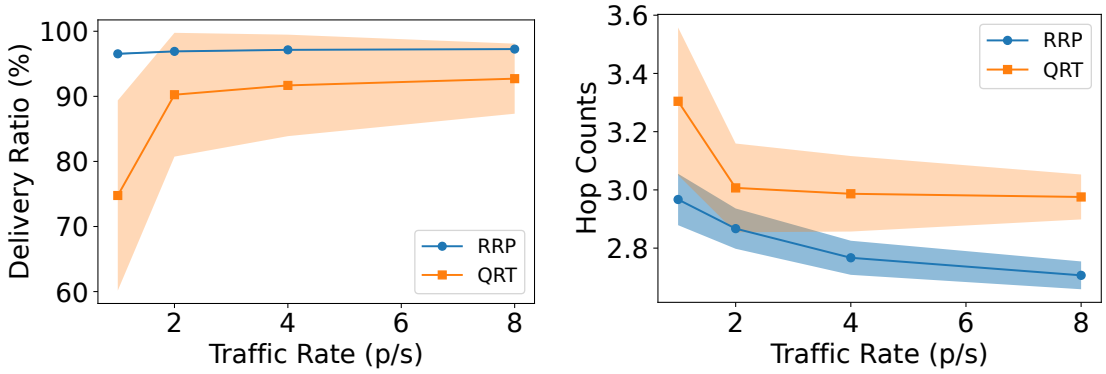
RRP has been benchmarked with QRT [45], which is an extension to RL-QRP routing protocol [44] where the authors integrated a reputation and trust scheme to deal with non-cooperative and misbehaving nodes in biomedical sensor networks. QRT was the only available RL-based routing protocol in the literature designed for WMSN that incorporates the TM scheme to achieve reliable data delivery. In order to ensure a fair comparison between the two protocols, the reported parameters setting of QRT have been adopted. Table 4.2 shows the simulation parameters setting. The learning rate η and the discount factor γ have been set to 0.5. The experiments were carried out using a discrete event simulator based on Simpy [200]. The simulation time is $500s$, where the first $50s$ is regarded as a training period unless otherwise indicated. This training period has been specified to allow QRT to converge, followed by a relatively long simulation time to study the stability of routing decisions of both protocols. During the simulation, the agents adopt the $\varepsilon-$greedy strategy to balance between exploration and exploitation where ε is set to 0.1 as in QRT. Each experiment has been repeated 30 times, and then the results have been averaged out and reported with one standard deviation. It is worth to mention here that when the sample size is 30, the sampling distribution approximates the Gaussian distribution [201].

Table 4.1: Symbols used in Chapters 4 and 5

Symbol	Meaning
S	The sink node
$\mathbb{N}$	The set of all sensor nodes in the network
$\eta \in [0,1]$	The learning rate parameter
$\gamma \in [0,1]$	The discount factor
$Q^i_{t+1}(s^i_t, a^i_t)$	The updated Q values of node i, given the state s^i_t and the action a^i_t
$r^i_{t+1}(s^i_{t+1})$	The new reward received at the end of the time unit
T^{ij}_t	The trust value maintained by i of node j at time t
ϵ	A threshold to specify the minimum required evidence
τ	The time window $[s]$
N^i_t	The neighbors of node $[i]$ at time t
$n^i \in N^i_t$	A neighbor of node i
$s^i_t \in \mathbb{S}$	The state of node i at time window t
$a^i_t \in \mathbb{A}$	The taken action by node i at the time window t
Q^{ij}_t	The Q value maintained by node i for node j at the time window t
θ	The exploration rate
O^{ij}	The observations maintained by node i for node j
$RQ^i_{t-1}(s^i_{t-1}, j)$	The last expected future reward received from node j
μ	The traffic rate
δ	A time lag
$\zeta \in]\,0,1]$	A loop penalizing parameter
$\alpha,\ \beta \in\ [\,0,1]$	The beta distribution levels
$b_t,\ d_t$	The slopes at time t
λ	The longevity factor

Table 4.2: RRP Simulation Parameters

Parameter	Value
Application	Poisson random traffic
Exponential transmission interval μ	1, 2, 4, 8
Radio Range	5m
Propagation loss model	Range propagation loss
Number of SN	64
Time unit	1s
Simulation Time	500s
Learning Period	50s
Learning rate η	0.5
Discount factor	0.5
$\varepsilon-$greedy	0.1



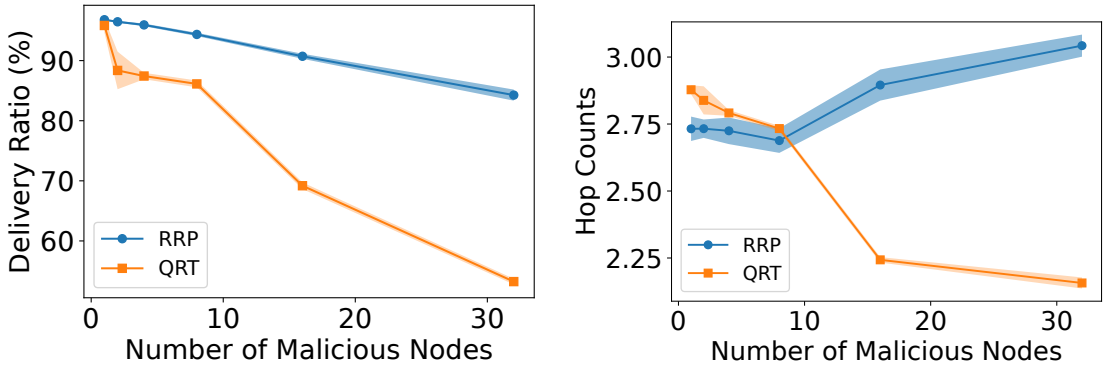
(a) (b)

Figure 4.4: The average delivery ratio and hop counts under normal operation

4.5.2 Normal Operation

In this experiment, the performance of RRP has been evaluated, assuming that there are no malicious activities inside the network. Benign nodes randomly drop around 1% of the received packets to relay as WMSN is intolerant to higher rates of packet loss. This experiment aims to ensure that RRP chooses the optimal path to the destination with the highest delivery ratio. Some SNs generate low traffic rates around 1 packet/s, such as heart rate sensors [172]. Therefore, the experiment has been run for four different traffic rates starting from $\mu = 1p/s$ and doubling the traffic rate each time. Fig. 4.4a and Fig. 4.4b show the average delivery ratio, and the average hop counts with one standard deviation, respectively. Results show that RRP achieves the highest delivery ratio with minimal variability, while QRT did not work well for the lowest traffic rate with a delivery ratio of 75%. QRT's performance shows a slight improvement for traffic rates starting at $\mu = 2p/s$ to achieve around 90%; however, the high variability of the delivery ratio confirms that QRT struggles to converge, as shown in the shadowed area of QRT protocol.

On the other hand, Fig. 4.4b reveals that RRP chooses the shortest path to the destination compared to QRT. This proves that although RRP normally updates the routing decision periodically, it performs efficiently thanks to its asynchronous updating methods to avoid bad routing decisions. It is worth noting that the performance is slightly enhanced for higher traffic rates because the learning agents can get more evidence from the environment to enhance their routing decisions.



(a) (b)

Figure 4.5: The average delivery ratio and hop counts under blackhole attacks

4.5.3 Blackhole Attacks

The Blackhole attack is a well-known attack in WMSN where compromised nodes drop all the received frames instead of forwarding them to the destination. This causes severe detrimental consequences, especially for medical applications [188]. In this experiment, the delivery ratio and the hop counts are evaluated under different blackhole attacks. The number of malicious nodes was doubled each time, starting from one and up to 50% of the total number of the SNs. The experiment was run for 30 times for each parameters setting, and then the results are averaged out and reported with one standard deviation as shown in Fig. 4.5a and Fig. 4.5b. The results reveal a superior performance for RRP in contrast with QRT. Although QRT performed well when there is only one malicious node, the delivery ratio sharply dropped by introducing more malicious SNs to the network due to the inability to detect the malicious paths. In contrast, RRP showed a steady superior performance even when 50% of the SNs are malicious. It is worth mentioning that the slight decrease in the delivery ratio of RRP when increasing the number of malicious SNs is due to ε-greedy strategy where 10% of the actions are made randomly with a view to exploring the environment. On the other hand, the hop count results explain how each protocol responds to the hostile environment. Fig. 4.5b shows that RRP performs better when there are up to 8 malicious SNs. When the number of malicious nodes increases, RRP needs more hops to reach the destination to avoid malicious SNs. However, in QRT, the number of hops needed to get to the destination is decreased unexpectedly by increasing the number of malicious nodes, which explains the poor delivery ratio. These results indicate that QRT failed to build reliable paths that avoid malicious nodes and confirm that RRP chooses the most reliable shortest paths.

4.5.4 Selective Forwarding Attacks

In the selective forwarding attack, the malicious nodes forward some frames and drop others selectively [63]. This behavior is hard to detect as the same malicious node could be trustworthy for some nodes and untrustworthy for others. In this experiment, RRP has been evaluated under selective forwarding attack, where malicious nodes randomly choose a list of neighbors not to relay their frames. Two scenarios have been considered. In the first, the malicious node randomly chooses a list of several neighbors x_t^i to drop their frames as in Eq. 4.4:

$$|x_t^i| = \begin{cases} \frac{|N_t^i|}{2}, & if \quad |N_t^i| = 2k \\ & where \quad k \in \mathbb{W} \\ \frac{|N_t^i|}{2} + 1, & if \quad |N_t^i| = 2k + 1 \end{cases} \tag{4.4}$$

This means that 50% to 67% of the received frames to relay will be dropped. In the second scenario, malicious nodes run volatile selective forwarding attacks by randomly changing x_t^i after every 20% of the simulation time. Fig. 4.6a and Fig. 4.6c show the delivery ratio under both attack scenarios. Our proposed RRP protocol outperforms QRT in both scenarios and provides a reliable delivery with minimal variability. At the same time, QRT shows a high variability when the number of malicious nodes is less than 25% of the total number of SNs, indicating difficulty in converging. By increasing the number of malicious nodes, the delivery ratio of QRT decreases significantly.

On the other hand, the hop counts results shown in Fig. 4.6b and Fig. 4.6d reveal how each protocol responds to the hostile environment. RRP performs better when the number of malicious nodes is less than 25%. Moreover, when the number of malicious nodes reaches 50%, the hop count gradually increases to avoid any paths through malicious nodes resulting in a slight increase in variability. This variability increase in the hop counts indicates the ability to find redundant, reliable paths to the destination, which could be seen in the stable delivery ratio. In contrast, QRT needs more hop counts for the limited number of malicious nodes. Furthermore, it fails to find reliable paths inferred from its low delivery ratio and hop counts.

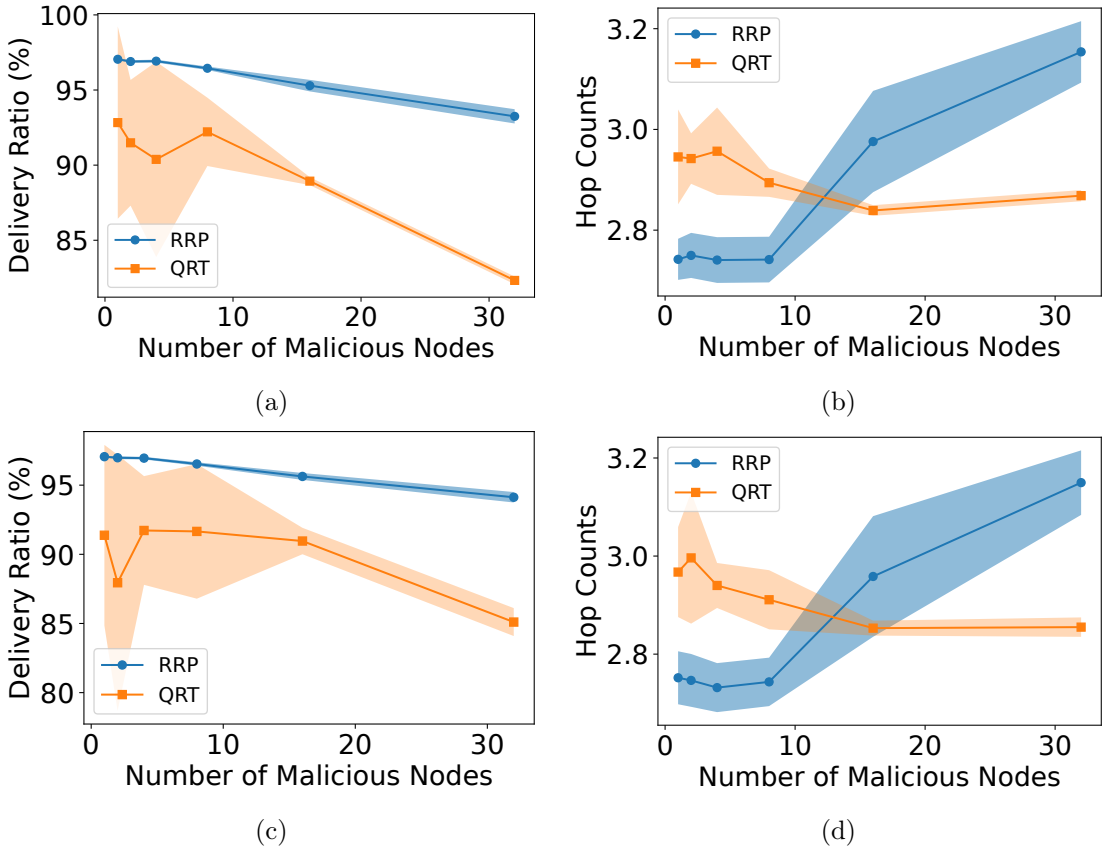


Figure 4.6: The average delivery ratio and hop counts under selective forwarding attacks

4.5.5 Sinkhole Attacks

The sinkhole attack is one of the most destructive attacks on routing protocols in which the malicious node attracts the network traffic by advertising false routing information [189]. It is a mix between a dropping attack and a route poisoning attack in which the malicious nodes try to poison the routing tables of other nodes inside the networks by advertising dishonest estimations to its neighbors. This complicated attack has been chosen the prove the robustness of RRP design. In RL-based routing protocols, the learning agents exchange routing information to update the Q table and re-evaluate the optimal paths. When the adversary advertises false overestimated information to a specific destination, it can poison the Q tables of other nodes and attract all the traffic in order to drop it.

In this experiment, the robustness of RRP is evaluated under different poisoning levels.

Four scenarios have been considered in this experiment. The malicious nodes advertise the actual Q values increased by 25%, 50%, 75% and 100%. In the last scenario, when the Q values are increased by 100%, the malicious nodes will advertise the value zero to the network, which is the highest Q value that could be achieved as the reward function is designed to penalize dropping activities to ensure that the learning agents will always choose the most reliable shortest path. Fig. 4.7a, 4.7c, 4.7e and 4.7g show the delivery ratio for the four scenarios. What stands out in these figures is the stable delivery ratio of RRP for different route poisoning levels, which reveals a high resiliency to sinkhole attacks. Moreover, Fig. 4.7b, 4.7d, 4.7f and 4.7h reveals how RRP finds the optimal paths through a hostile environment. RRP shows the same behavior as previous experiments when the number of malicious SNs increases. It avoids malicious nodes by choosing the most reliable path with minimal achievable hop counts. It is worth noting that in Fig. 4.7h when the malicious nodes advertise zeros as their best estimation, RRP shows a slight increase in hop counts even for a low number of malicious nodes, but with a high delivery ratio. The reason behind this behavior is that advertising this level of fake information affects the Q tables of the surrounding nodes, making the learning agent even tries to avoid the neighbors of malicious nodes.

On the other hand, QRT shows a good delivery ratio when only one malicious node exists. However, by increasing the number of malicious nodes, the delivery ratio drops significantly to levels below 50% for 32 malicious nodes. This failure in avoiding malicious nodes can be clearly seen in the hop counts results, where QRT experiences a steep drop in contrast to what is expected, which explains the meager delivery ratio as packets would be ended up in a sinkhole.

4.5.6 On-Off Attacks

Although trust management schemes detect malicious activities, they are vulnerable to on-off attacks, where smart adversaries can change their behavior alternately to cheat the TMS and keep themselves undetected [202]. As RRP incorporates trust information into the reward function, it needs to prove its merit against complicated on-off attacks. Therefore, we evaluate the performance of RRP under different complicated on-off attacks in this experiment as the failure to detect on-off attacks, negatively impacts the performance of trust-based routing protocols by making them make wrong routing decisions. The on-off attack cycle consists of one on and one off periods. During the on period, the adversary drops packets intentionally, while during the off period, it behaves

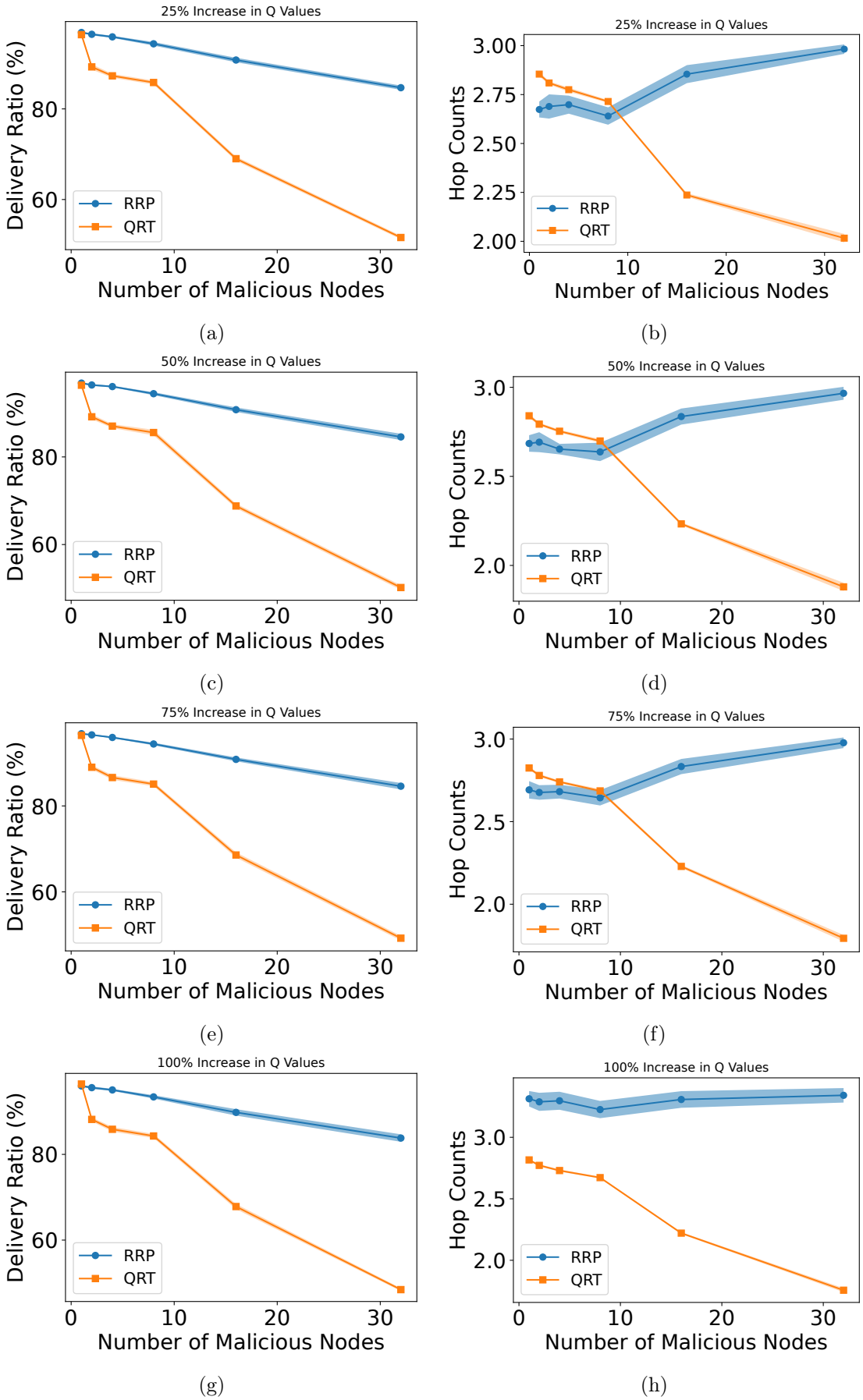


Figure 4.7: The average delivery ratio and hop counts under sinkhole attacks

well to rebuild its trust score and keep itself undetected. In this experiment, three simulation scenarios have been considered, variable traffic rates, variable on-off cycles, and non-identical periods.

Variable Traffic Rates

Experiments in Chapter 3 shows that some TM schemes failed to operate properly under low traffic rates. Thus, our first simulation scenario is designed to evaluate the routing performance under on-off attacks for different traffic rates starting from a low traffic rate $\mu = 1$ and doubling it each time. Fig. 4.8a, 4.8c, 4.8e and 4.8g show the delivery ratio for the traffic rates $\mu = 1$, $\mu = 2$, $\mu = 4$ and $\mu = 8$, respectively. The on-off attack cycle is set to 20. Results show that QRT does not work properly for low traffic rates. Moreover, by increasing the number of malicious nodes, the delivery ratio decreased significantly, which could be attributed to the inability to avoid malicious nodes from the routing path as shown in Fig. 4.8b, 4.8d, 4.8f and 4.8h. On the other hand, RRP shows superior performance for all traffic rates. It achieved a delivery ratio between around 90% to 97% for all malicious nodes ratios. The hop counts results show that RRP can find alternative paths to avoid malicious nodes, which obviously appears when having $25\% - 50\%$ of nodes behaving maliciously. It is worth noting that by increasing the traffic rate, RRP can find more optimal paths, which could be attributed to having more evidence from the environment.

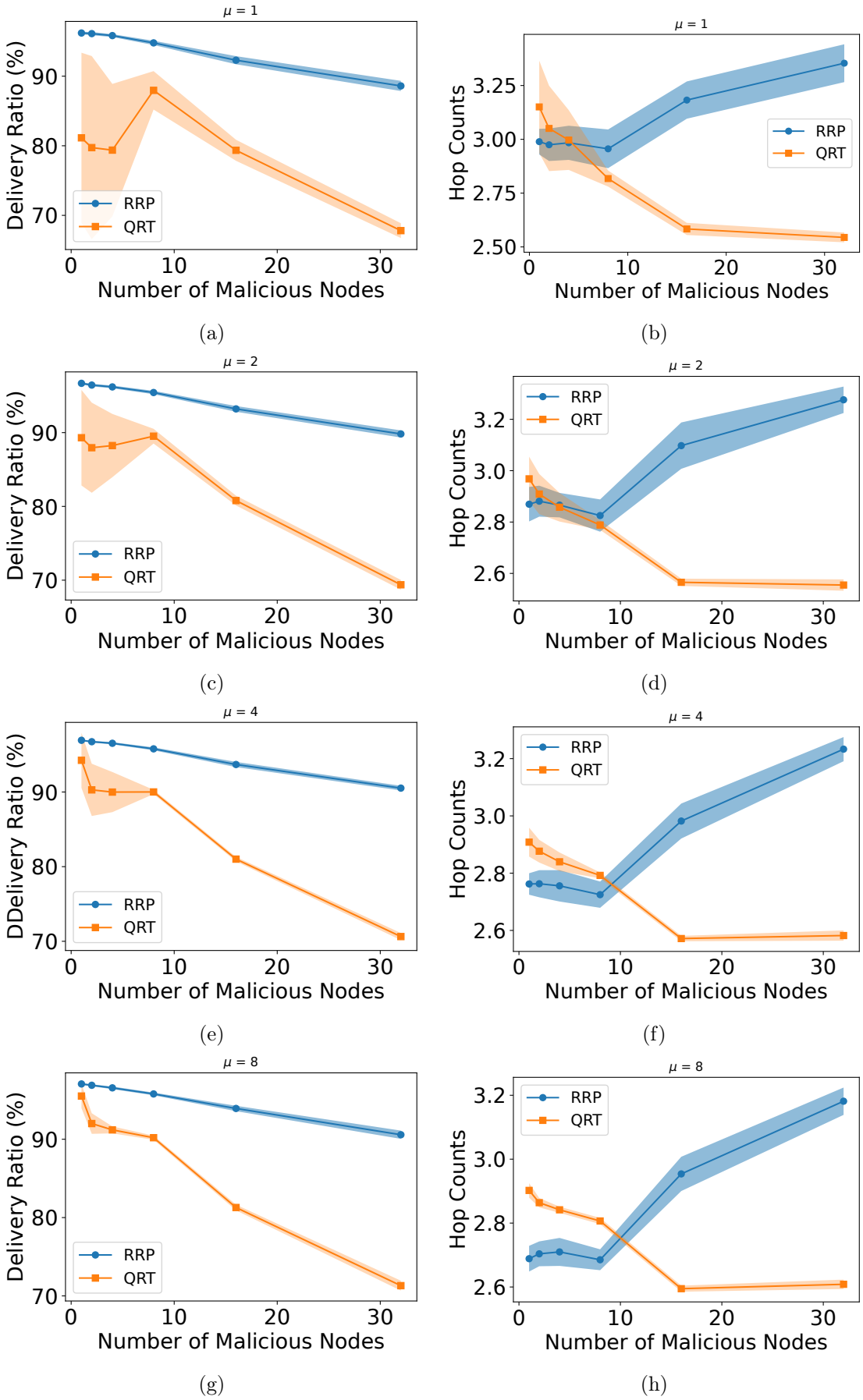


Figure 4.8: The average delivery ratio and hop counts under On-Off attacks for variable traffic rates

Variable On-Off Cycles

This experiment evaluates the performance under different on-off attack cycles. The on-off attack's cycle varies from $10s$ to $40s$. Fig. 4.9a, 4.9c, 4.9e and 4.9g show the delivery ratio for different cycles, while Fig. 4.9b, 4.9d, 4.9f and 4.9h show the hop counts' results. RRP shows superior and stable performance for all on-off cycles. It achieved the same delivery ratio of the previous scenario, coupled with the same behavior of selecting paths to destinations, which again indicates its robustness against different on-off attacks. On the other hand, QRT shows lower delivery ratios with high variability for a low number of malicious nodes. By increasing the number of malicious nodes, the delivery ratio decreased significantly, which explains the reason behind the decreasing hop counts when having more malicious nodes.

Non-Identical Periods:

Smart adversaries can launch more sophisticated on-off attacks by making the on period smaller than the off one, which can cheat the TMS and make the attack more difficult to detect. In this experiment, non-identical on-off attacks are launched by making the on period less than the off period. Four scenarios have been considered by varying the on period from 25% of the off period and up to 100%. The on-off cycle is set to $20s$ and the traffic rate $\mu = 4$. Fig. 4.10a, 4.10c, 4.10e and 4.10g show the delivery ratio for different on period's ratios, while Fig. 4.10b, 4.10d, 4.10f and 4.10h show the hop counts. RRP shows a stable, superior performance, indicating its ability to detect the attack and isolate the malicious nodes. On the other hand, QRT shows low delivery ratios by increasing the number of malicious nodes with high variability for the low number of malicious nodes, which indicates difficulty in converging to the global optimum.

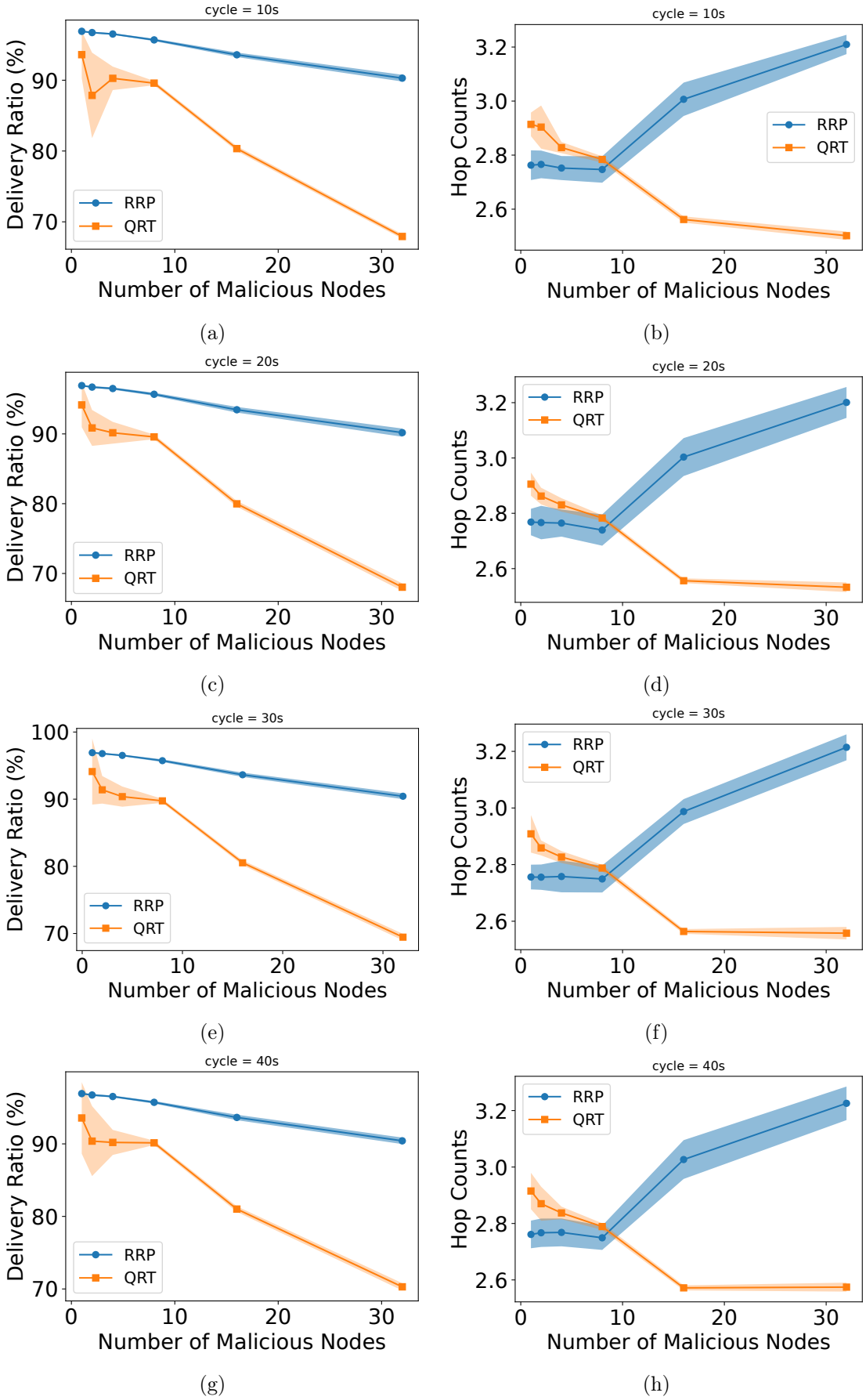


Figure 4.9: The average delivery ratio and hop counts for different On-Off attacks' cycles

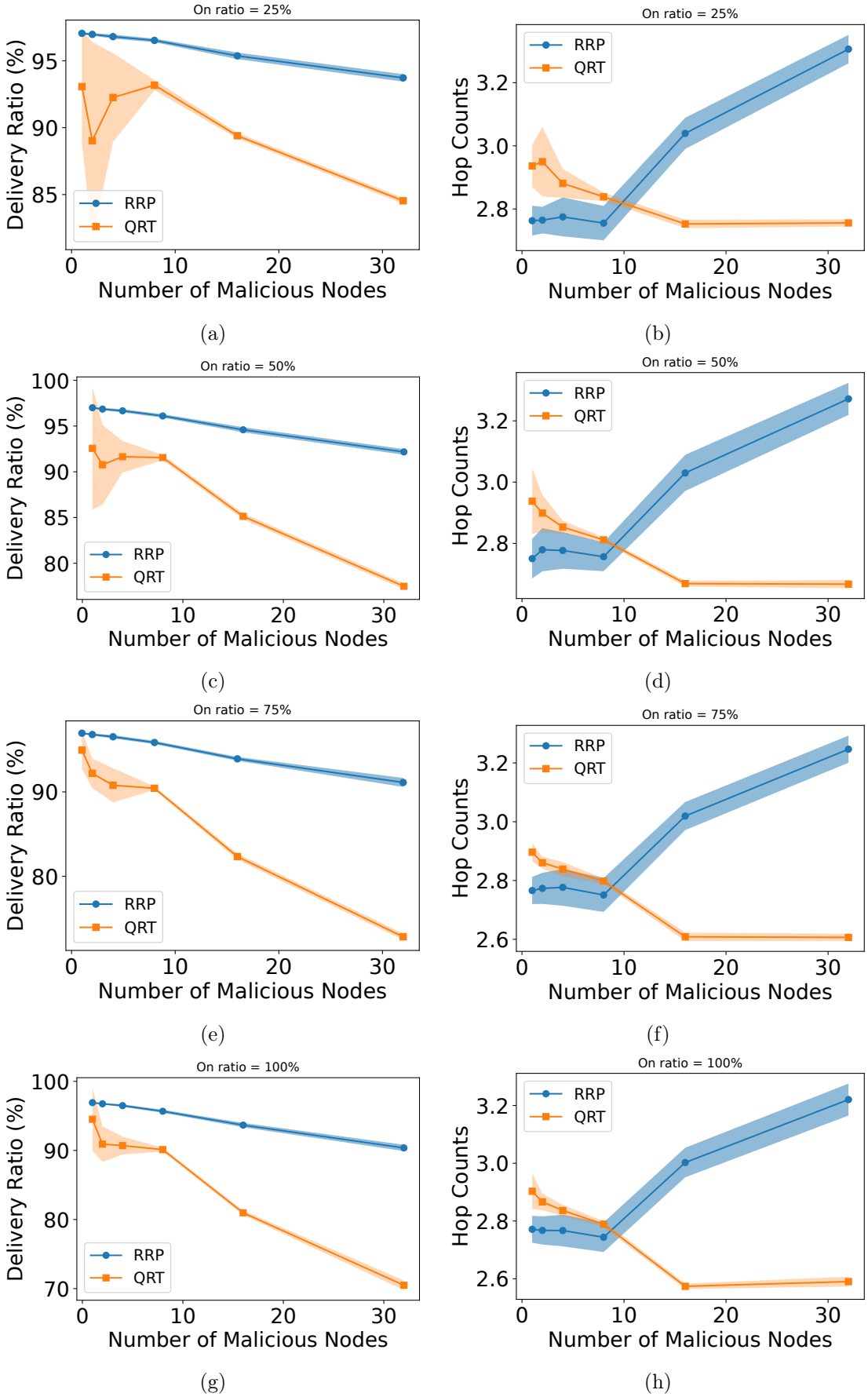


Figure 4.10: The average delivery ratio and hop counts under non-identical on-off periods

4.5.7 Network Dynamicity and Convergence

The convergence time is a crucial factor in routing applications as slow convergence results in more packets to lose, which could endanger the patient's life. Moreover, nodes' mobility could change the environment and require the algorithm to re-converge again. In this experiment, the convergence has been studied for the stationary and non-stationary environment under blackhole attacks where 50% of the nodes are malicious. First, stationary SNs have been considered to compare the convergence time of both protocols. Fig. 4.11 shows the convergence time of both protocols. RRP is able to converge with less than $20s$ thanks to its asynchronous updating method in which the agent updates its routing decision engine once evidence obtained from the environment. This method makes RRP an adaptive protocol that can reflect any environment change. In contrast, QRT needs around double this time to converge. It is worth noting that QRT shows a bit better performance at the start because it uses positional information to make routing decisions, whereas RRP only depends on its trial/error process to learn the optimal routing decisions.

In the second scenario, mobility has been introduced to study how algorithms re-converge in a dynamic environment. For example, patients can change their locations within the hospital ward. Therefore, in this experiment, two different patients will change their locations at time $50s$ and $100s$. The patient could have up to 3 SNs. Thus, three simulations have been run for 1, 2, and 3 randomly chosen SNs for each patient. The results show a fast re-convergence in all cases for RRP, as shown in Fig. 4.12a. Once the environment change happens, RRP updates its routing engine asynchronously to reflect the new environment. This could be seen as a slight decrease in the delivery ratio at the time of movements, followed by fast re-convergence. On the other hand, QRT experienced a noticeable decrease with difficulty in re-converging, especially after the second movement, as shown in Fig. 4.12b. The reason behind this poor performance could be attributed to considering positional information, which influences the routing decision.

4.5.8 Computational Overhead

In this subsection, we compare the average processing time and memory consumption of both protocols, RRP and QRT. The experiment was carried out on an Intel Core i5-8500T processor at 2.1GHz and 8GB RAM. The simulation has been run for 30 times, and then the results have been averaged out and reported with one standard deviation.

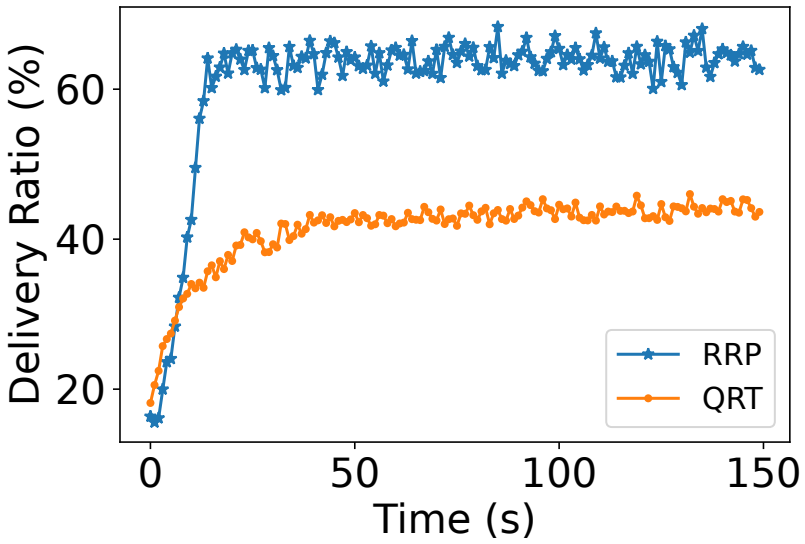
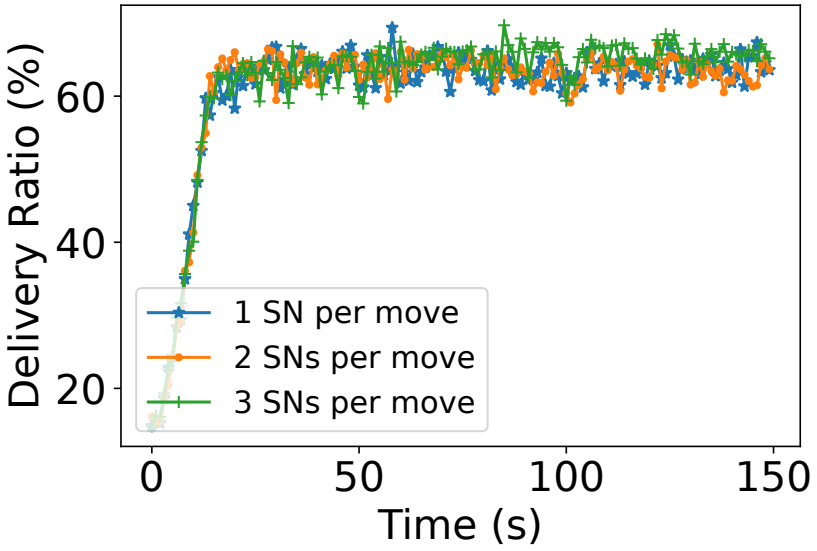


Figure 4.11: The average convergence time for static SNs

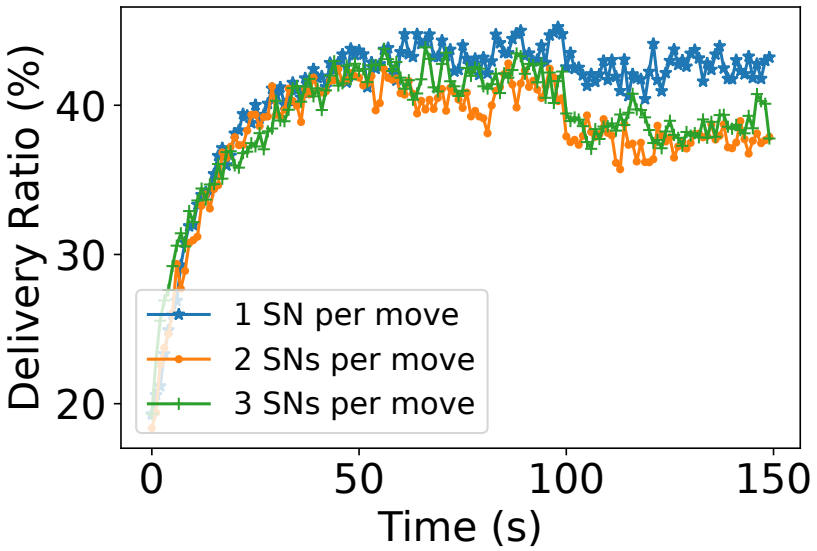
The network is in normal operation, and no attacks are launched during the simulation. The traffic rate is set to $\mu = 4p/s$ as QRT does not perform properly for lower traffic rates.

Fig. 4.13a shows the average processing time of RRP and QRT for the whole simulation process, including all the SNs. The results show that QRT consumes more processing time than RRP. Moreover, results show high variability of around 23%. This variability indicates that the algorithm sometimes takes longer to converge; hence, more packets will loop inside the network before reaching their destination. On the other hand, RRP consumes less processing time and saves around 35% of the processing time of QRT. Moreover, RRP shows almost no variability, indicating the stability of performance and the ability to converge at approximately the same time for different simulation runs. This lightweight processing overhead is attributed to the proposed resource-efficient RL model, where the learning agent receives one reward for multiple actions and hence updates the routing engine less than the traditional RL model.

The second important performance metric is memory consumption. The average memory consumption was calculated and reported with one standard deviation for the whole simulation process, including all the nodes in as shown in Fig. 4.13b. The memory allocation has been traced during the simulation using tracemalloc [203], a trace memory allocation module. Results show that QRT consumes a considerable amount of memory,



(a)

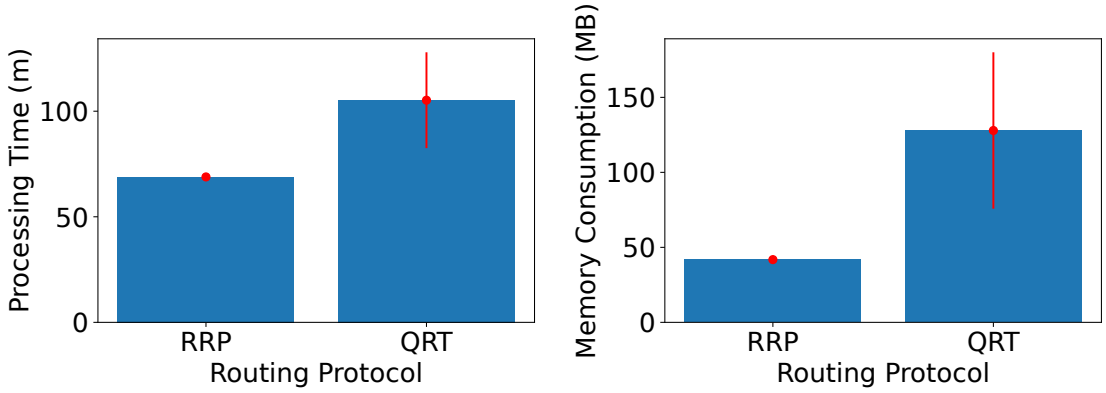


(b)

Figure 4.12: The average delivery ratio under different mobility scenarios for RRP and QRT, respectively

around 128MB, with a high variability of around 52%. This high memory consumption is attributed to having more packets looping in the network, while the high variability indicate difficulty in converging to the global optimum. For each simulation run, QRT converges to a local optimum, which causes inconsistent memory consumption between

different simulation runs. On the other hand, RRP is a memory-conservative protocol. It consumes a decent amount of memory, around 42MB, which saves around 67% of the memory consumed by QRT. Moreover, RRP shows almost no variability, indicating that RRP did not experience any converging difficulties thanks to its novel updating mechanisms.



(a) (b)

Figure 4.13: The average processing time and memory consumption

4.5.9 Hyperparameters Tuning

Optimizing the used hyperparameters lead to faster convergence and overall better performance. In previous experiments, the benchmark parameters setting has been adopted to ensure fair compression between both protocols. In this experiment, the learning rate η and the discount factor γ are tuned using the grid search approach. As both parameters are continuous in the domain $[0, 1]$, a step of 0.1 has been used for each parameter. This involves a combination of 121 simulation parameters. Each one of them has been run for 30 times. The simulation has been run for blackhole attacks where 50% of the nodes are malicious and the traffic rate is set to $1p/s$. Fig. 4.14 shows the heatmap of the average delivery ratio of these simulations. Closer inspection of the figure shows poor delivery ratio for $\eta = 0$ and $\gamma = 0$. When $\eta = 0$, the algorithm only uses its current observation and does not learn from previous experience, which causes poor performance. The values between $[0.2, 0.4]$ show the highest performance, although the heatmap also shows good performance for $\eta = 0.8$; however, consulting the hop counts results shows that the algorithms take slightly more hops to reach the destination, indicating that the algorithm did not converge to the global optimum.

On the other hand, the discount factor plays a significant role also. What stands out in the figure is myopic learning agent performs poorly. For instance, when the discount factor $\gamma = 0$, the learning agent only considers its direct observations from the observable environment to choose the optimal path, which likely leads to losing the packet due to existing a malicious node in the in-observable path. Interestingly, the maximum value of γ also shows poor performance as the learning agent weighs current and expected future rewards equally, which influences the routing decision negatively based on in-observable future rewards. Thus, the optimal value for the discount factor is around 0.8, which could efficiently balance the current and future expected rewards.

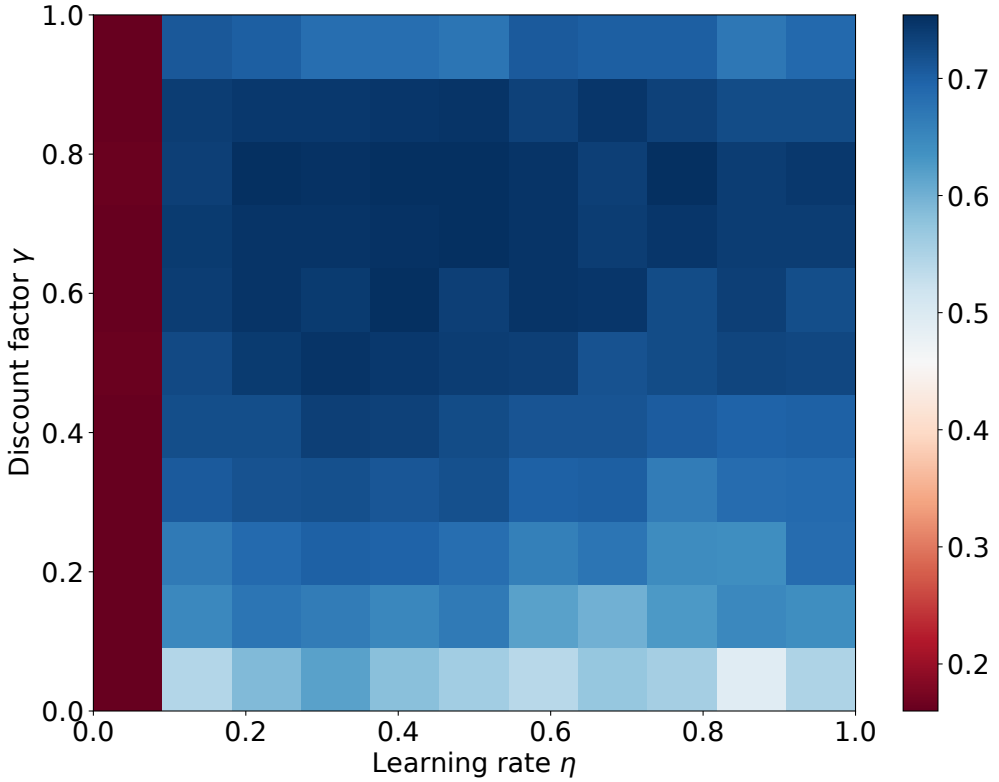


Figure 4.14: Hyperparameters optimization heatmap

4.5.10 Exploration Exploitation Optimisation

Exploration exploitation trade-off is a critical component in RL model. The learning agent needs to explore the stochastic environment in order to maximize the long-term reward. During the exploration phase, the learning agent tries to discover the most rewardable actions. However, taking an action at a state s may affect the immediate

reward as well as the subsequent rewards, while insufficient exploration may lead to converging to a sub-optimal solution. ε-greedy and Softmax exploration methods are the most used algorithms in the literature to balance exploration-exploitation. ε-greedy has been used in previous experiments. However, the reported values in our benchmark have been adopted to ensure a fair comparison. In this experiment, we optimize the value ε under blackhole attacks where 50% of the nodes are malicious. ε is a continuous value in the range [0, 1]. Therefore, a step of 0.01 has been chosen. The simulation has been run for values in the range $[0, 0.2]$ as higher values over-explore the environment and shows poor performance. Fig. 4.15 shows the results of only some ε values for clarity. The value $\varepsilon = 0.06$ achieves the highest reward and is able to converge faster than other values. This means that the learning agent randomly explores the environment with a probability of 6%.

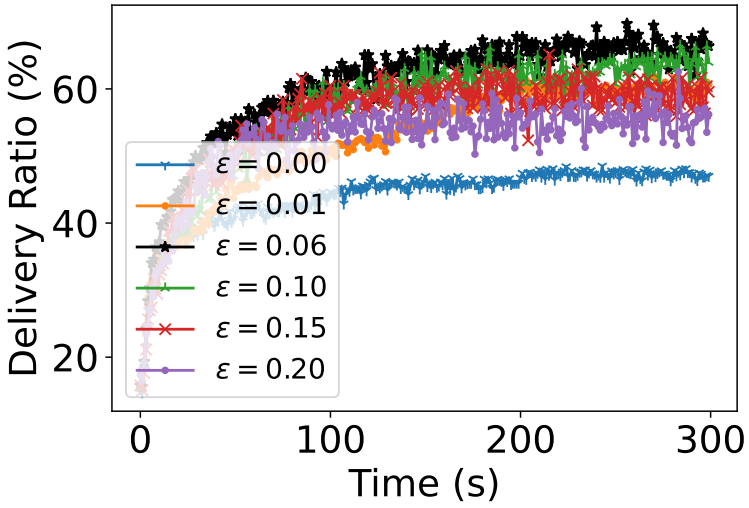


Figure 4.15: Optimizing ε-greedy exploration algorithm

Softmax exploration algorithm is a value-based approach to exploring the environment in which the learning agents make informed routing decisions based on its Q table. Softmax algorithm is modeled using Gibbs distribution as shown in Eq. 4.5.

$$\pi(a|s) = Pr\{a_t = a|s_t = s\} = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_1^n e^{\frac{Q(s,a)}{\tau}}} \tag{4.5}$$

where τ is called the temperature parameter, which is used to control the probability of choosing the greedy action. Decreasing the value of τ, increases the probability of

choosing the greedy action. Moreover, ε-greedy could be derived from Softmax algorithm when $\tau \rightarrow \infty$ as all possible actions will have the same probability. Optimizing the temperature value of the Gibbs distribution is not straightforward [204] as any change in reward function can influence the probabilities of available actions. In this experiment, the temperature parameter will be optimized under the same conditions. As $\tau \in \mathbb{R}^+$, we start at $\tau = 0.01$ and then increase the value by 0.05. Fig. 4.16 shows the results of some temperature values for clarity. The performance starts poorly at $\tau = 0.01$ and enhanced gradually to reach the peak at $\tau = 0.05$, and further increase beyond 0.05 decreases the delivery ratio.

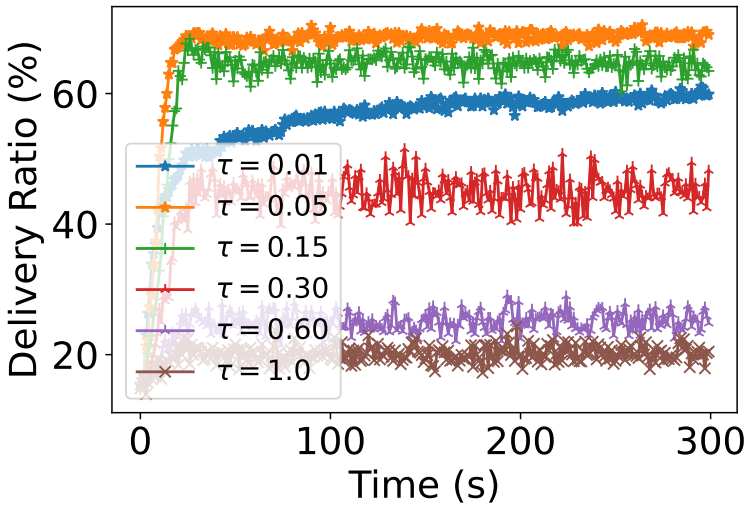
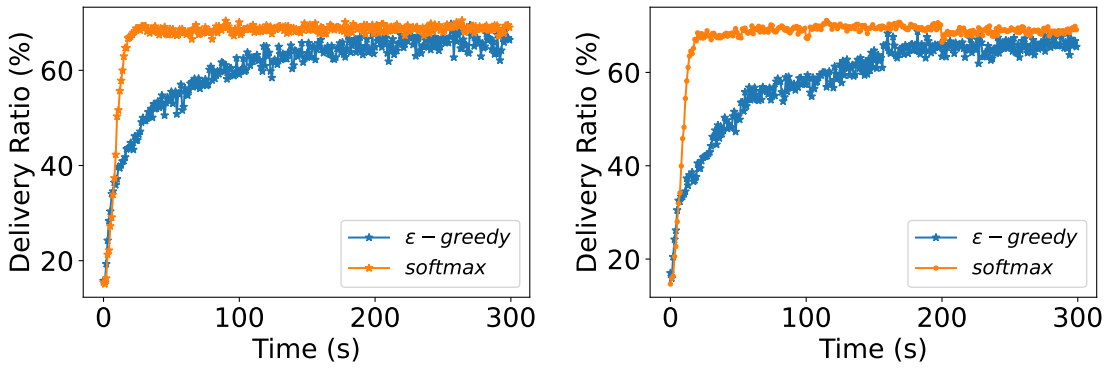


Figure 4.16: Optimizing softmax exploration algorithm

In the third experiment, we compare both algorithms using optimized values for stationary and non-stationary environments. Fig. 4.17a shows the results for a stationary environment. Softmax shows superior performance. The algorithm converges fast to the global optimum, while ε-greedy needs more time to converge. This could be attributed to the mechanism of making routing decisions. ε-greedy exploration could be regarded as blind exploration as actions are chosen randomly during the exploration, while the Softmax algorithm takes informative actions based on the current estimations. On the other hand, the results of the non-stationary environment in Fig. 4.17b show that both algorithms are able to re-converge fast after movement detection.



(a) (b)

Figure 4.17: Comparing ε-greedy and softmax exploration algorithms for stationary and non-stationary environment

4.6 Conclusion

There is still a persistent need for a lightweight and secure routing protocol for WMSN. Although RL is regarded as a promising approach to building a routing protocol for WMSN, the widely used RL model is a resource-consuming model. Moreover, reliable data delivery cannot be achieved using only routing metrics as this information cannot deal with the free will of other relay nodes inside the network. Realizing the aforementioned problems open the way to re-design a lightweight RL model and integrate a security tool with the routing decision engine to ensure lightweight and reliable data transfer for WMSN. The proposed RL model does not necessitate updating the Q table after each sent/forwarded packet, but rather it updates it periodically after receiving a reward for a set of actions within one time unit. The performance results show that the proposed RL model can significantly reduce computational overhead. Furthermore, integrating TMS with the routing engine enables reliable data delivery and avoids malicious paths that cannot be achieved using traditional routing metrics. Finally, the experimental results prove the robustness of our proposed method in defeating well-known dropping attacks.

Chapter 5

Double Q-Learning Energy-Aware Routing

In this chapter, we propose DQR, a double Q-learning routing protocol to meet WMSN requirements and overcome the positive bias estimation problem of the Q-learning-based routing protocols. DQR extends the proposed lightweight RL model in Chapter 4 in order to reduce the computational and communication overheads, coupled with the effective trust management scheme proposed in Chapter 3. Moreover, an energy model is also integrated with DQR to enhance the network lifetime and balance relaying overhead across the network. The experimental results demonstrate robust performance under various attacks with minimal resource footprint and efficient energy consumption.

The main findings of this chapter has been accepted to publish in the 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm).

5.1 Introduction

Reinforcement Learning (RL) has been used recently to solve distributed optimization problems, such as routing [205]. RL-based routing protocols rely on an existence of a learning agent that acts with the environment and receives rewards based on its actions. By interacting with the network environment, the learning agents will be able to maximize their reward by making optimal forwarding decisions. Q-learning, which is a model-free RL algorithm, is the most used algorithm for both centralized and decentralized routing protocols [40]. Although this approach is able to produce an efficient routing

protocol that can outperform other algorithms, it still has drawbacks. First, as it works without prior knowledge about the environment, it requires a series of randomly chosen actions to explore the environment before converging on the optimal solution. WMSN cannot tolerate a long learning period because of its sensitive applications. Second, Q-learning has an inborn overestimation problem which has been overlooked for a long time [206]. It uses the maximum value as an estimation for the maximum expected value. It worth noting that Q-learning is mathematically proved to converge; however, the convergence time is a critical factor, especially for sensitive applications. Q-learning uses the max operator to evaluate the next state value. This could cause huge overestimation in stochastic environments. The learning agent could get fooled by $\underset{a \in A}{max} Q_t^i(s_{t+1}^i, a_t^i)$, which is an estimate for $\mathbb{E}\{\underset{a \in A}{max} Q_t^i(s_{t+1}^i, a_t^i)\}$, and the latter approximates $\mathbb{E}\{Q_t^i(s_{t+1}^i, a_t^i)\}$. The max operator in Eq. 4.1 is prone to be larger when there is a noise and this could be clearly appear when there are no enough samples. The authors in [206] proposed double Q-leanring with two estimators to overcome this shortcoming. Hence, in this chapter, we use double Q-learning to build the routing engine as the routing performance may be impacted negatively due to this positive bias. Third, although different parameters have been considered in protocol design, ensuring reliable data transfer is still challenging as senders cannot predict the behavior of other nodes in the path to the destination. Moreover, taking into consideration more parameters may optimize the routing decisions, but it involves a significant overhead increase, especially when information must be exchanged between learning agents. Therefore, a suitable solution is needed to overcome these aforementioned shortcomings.

5.2 Contribution

The main contribution of this paper is threefold. First, double Q-learning is used with our proposed RL model in Chapter 4 to address the inborn overestimation problem of Q-learning-based routing protocols. Second, a reliable, lightweight, and energy-efficient routing protocol for WMSN has been proposed. Third, extensive analysis has been carried out to ensure the robustness of our proposed protocol under different scenarios.

5.3 Related Work

Developing a secure, reliable, and efficient routing protocol for WSN is still an open area of research. It is more challenging in WMSN due to its resource scarcity and critical applications. Abundant research has been carried out to propose an efficient routing

protocol using different metrics and methods. As mentioned in previous chapters, reinforcement learning has been widely used to find the optimal routing path with minimal overhead. Q-learning, which uses temporal difference (TD) to estimate the value of an action in a given state, is extensively used to build an efficient routing policy. However, Q-leaning suffers from an overestimation problem, which overlooks the optimal action in some cases [206]. Therefore, double Q-learning, which is an off-policy RL algorithm, is introduced to solve the overestimation problem by using double estimators to approximate the maximum expected value. To the best of our knowledge, only a few works used double Q-learning to develop a routing protocol. Authors in [46] proposed DQLR, a double Q-learning routing protocol for Delay Tolerant Networks (DTN). However, DQLR only uses the number of hops between the source and the destination as a metric. It achieved an acceptable delivery ratio under normal operation. However, considering the hop count as the only metric is insufficient to deal with complicated scenarios.

On the other hand, researchers use various metrics to build the Q-learning reward function in order to achieve an efficient routing protocol, such as delivery delay, the number of hops, remaining energy, and location information [193, 41, 207, 205]. Although this kind of metrics could produce an efficient forwarding method, it cannot deal with malicious activities launched by insiders. Therefore, the routing protocol needs a different source of information to make an informed routing decision, such as Trust Management System (TMS). According to our literature review, only two routing protocols [42, 45] are available in the literature that integrate a TMS with Q-learning as detailed in Chapter 4.

5.4 Protocol Design

The routing protocol must always choose the optimal path in order to achieve a high delivery ratio and low energy consumption. However, Q-learning-based routing protocols could suffer from poor performance due to the action-value overestimation problem. This biased estimation leads to bad routing decisions that negatively affect the packet delivery ratio. Moreover, increasing the number of transmissions aggravates the energy consumption as transmission activities account for around 80% of the total consumed energy [194]. Therefore, a new approach to achieving an efficient routing protocol is required.

In this section, the proposed routing protocol to address this problem is presented. The

design requirements are justified, and the proposed algorithms are comprehensively discussed. It is worth mentioning that the same network and threat models presented in Chapter 4 are re-used in this chapter for evaluation.

Algorithm 6: DQR routing protocol

Input:
The reward: $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j)$
The Q tables: Q_t^A & Q_t^B
The trust table: T_t
Output: Optimal next hop $a_t^{(i)}$
Initialization:

$$Q_0^{A(i)}(n^{(i)} \in N_t^{(i)}) = Q_0^{B(i)} = \begin{cases} 0 & if \quad n^{(i)} \neq S \\ 1 & if \quad n^{(i)} = S \end{cases}$$

$T_0^{(i)}(n^{(i)} \in N_t^{(i)}) = E[uni(0,1)] = 0.5$

$$a_1^{(i)} = \begin{cases} S & if \quad S \in N_1^{(i)} \\ n^{(i)} & | \quad n^{(i)} \in N_1^{(i)} \end{cases}$$

while *TRUE* **do**
  $Wait \quad \tau$
  $Broadcast \quad max(Q_t^{A(i)})$ & $max(Q_t^{B(i)})$
  if $\varepsilon - greedy > \theta$ **then**
    $a_t^{(i)} = \underset{a \in A}{argmax}(\frac{Q_t^{A(i)}(s,.) + Q_t^{B(i)}(s,.)}{2})$
    Calculate $r_{t+1}^{(i)}(s_{t+1}^{(i)}, a_t^{(i)})$ as in Eq. 5.5
    $Q_{t+1}^{A(i)}(s_t^{(i)}, a_t^{(i)})$ & $Q_{t+1}^{B(i)}(s_t^{(i)}, a_t^{(i)})$ Synchronous update as in Algorithm 7
  else
    $a_t^{(i)} \leftarrow n_t^{(i)} \mid n_t^{(i)} \in N_t^{(i)}$
    Calculate $r_{t+1}^{(i)}(s_{t+1}^{(i)}, a_t^{(i)})$ as in Eq. 5.5
    $Q_{t+1}^{A(i)}(s_t^{(i)}, a_t^{(i)})$ & $Q_{t+1}^{B(i)}(s_t^{(i)}, a_t^{(i)})$ Synchronous update as in Algorithm 7
  end
  $s_t^{(i)} \leftarrow s_{t+1}^{(i)}$
end

Algorithm 7: DQR synchronous updating

Input:
The Q Table: $Q_t^{A(i)}$ and $Q_t^{B(i)}$
The reward: $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j)$
The trust table: T_t
Output: $Q_{t+1}^{A(i)}$ and $Q_{t+1}^{B(i)}$
while *TRUE* **do**
  $Wait \quad \tau$
  foreach $j \in N_t^i$ **do**
    if $j == a_t^i$ **then**
      $\rho \leftarrow rand(0,1)$
      if $\rho > 0.5$ **then**
        Define $a^{*(i)} = \underset{a \in A}{argmax} Q_t^{A(i)}(s_{t+1}^{(i)}, a_t^{(i)})$
        $Q_{t+1}^{A(i)}(s_t^{(i)}, a_t^{(i)}) \leftarrow$
        $(1-\eta) Q_t^{A(i)}(s_t^{(i)}, a_t^{(i)}) + \eta[r_{t+1}^{(i)}(s_{t+1}^{(i)}) + \gamma Q_{t+1}^{B(i)}(s_{t+1}^{(i)}, a_t^{*(i)})]$
      else
        Define $b^{*(i)} = \underset{a \in A}{argmax} Q_t^{B(i)}(s_{t+1}^{(i)}, a_t^{(i)})$
        $Q_{t+1}^{B(i)}(s_t^{(i)}, a_t^{(i)}) \leftarrow$
        $(1-\eta) Q_t^{B(i)}(s_t^{(i)}, a_t^{(i)}) + \eta[r_{t+1}^{(i)}(s_{t+1}^{(i)}) + \gamma Q_{t+1}^{A(i)}(s_{t+1}^{(i)}, b_t^{*(i)})]$
      end
    else
      if $|O^{ij}| > \epsilon$ **then**
        $Q_{t+1}^{A(i)}(s_t^{(i)}, j) \leftarrow$
        $(1-\eta) Q_t^{A(i)}(s_t^{(i)}, j) + \eta[r_{t-\delta}^{(i)}(s_{t-\delta}^{(i)}, j) + \gamma \underset{j \in N_t^{(i)}}{max} Q_t^{A(i)}(s_{t+1}^{(i)}, j)]$
        $Q_{t+1}^{B(i)}(s_t^{(i)}, j) \leftarrow$
        $(1-\eta) Q_t^{B(i)}(s_t^{(i)}, j) + \eta[r_{t-\delta}^{(i)}(s_{t-\delta}^{(i)}, j) + \gamma \underset{j \in N_t^{(i)}}{max} Q_t^{B(i)}(s_{t+1}^{(i)}, j)]$
      else
        $Q_{t+1}^{A(ij)} \leftarrow Q_t^{A(ij)}$
        $Q_{t+1}^{B(bij)} \leftarrow Q_t^{B(ij)}$
      end
    end
  end
end

Algorithm 8: DQR asynchronous updating

Input: A packet to forward: $P_t^{(sd)}$
Output: Updated Routing
while *TRUE* **do**
  if $\forall\, i \in \mathbb{N}$ *receives* $P_{t+\delta}^{(id)}$ **then**
    if $\eta == 1$ **then**
      $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j) = -e^{\eta}(1 - T_t^{(ij)}).F_t^{(i)}$
    else
      $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j) = -(1 - T_t^{(ij)}).F_t^{(i)}$
    end
    if $RQ_{t-1}^{A(i)}(s_{t-1}^{(i)}, j) \wedge RQ_{t-1}^{B(i)}(s_{t-1}^{(i)}, j)$ **then**
      update $Q_t^{A(ij)}$ and $Q_t^{B(ij)}$ using $r_{t+1}^{(i)}$, $RQ_{t-1}^{A(i)}$ and $RQ_{t-1}^{B(i)}$
    else
      $Q_{t+1}^{(i)}(s_t^{(i)}, a_t^{(i)} = n_j) \leftarrow Q_t^{(ij)} - \zeta$
    end
    $a_t^{(i)} = \underset{n_t^{(i)} \in N_t^{(i)}}{argmax}(\frac{Q_t^{A(i)}(s,.) + Q_t^{B(i)}(s,.)}{2})$
    Update $P_t^{(id)}$
    Send $P_t^{(id)}$
  end
  if $\forall\, i \in \mathbb{N}$ *receives* $P_t^{(jd)} \wedge a_t^{(i)} = j$ **then**
    if $\eta == 1$ **then**
      $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j) = -e^{-\mu}(1 - T_t^{(ij)}).F_t^{(i)}$
    else
      $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j) = -(1 - T_t^{(ij)}).F_t^{(i)}$
    end
    if $RQ_{t-1}^{A(i)}(s_{t-1}^{(i)}, j) \wedge RQ_{t-1}^{B(i)}(s_{t-1}^{(i)}, j)$ **then**
      update $Q_t^{A(ij)}$ and $Q_t^{B(ij)}$ using $r_{t+1}^{(i)}$, $RQ_{t-1}^{A(i)}$ and $RQ_{t-1}^{B(i)}$
    else
      $Q_{t+1}^{(i)}(s_t^{(i)}, a_t^{(i)} = n_j) \leftarrow Q_t^{(ij)} - \zeta$
    end
    $a_t^{(i)} = \underset{n_t^{(i)} \in N_t^{(i)}}{argmax}(\frac{Q_t^{A(i)}(s,.) + Q_t^{B(i)}(s,.)}{2})$
    Forward $P_t^{(jd)}$
  end
end

5.4.1 DQR Protocol

DQR is designed to address the inborn bias estimation in Q-learning-based routing protocols as well as enhance the network lifetime by incorporating an energy model into the routing decision engine. The reward function is defined as punishment to ensure that the learning agent always chooses the lowest-cost path. Moreover, in order to reduce the computational overhead of the traditional RL model, DQR re-uses the proposed RL model in Chapter 4, assuming that the network will be static for a short period, which is an acceptable assumption as nodes could be regarded as stationary for a short interval. This assumption allows the learning agent to perform the same action multiple times before receiving the corresponding reward. Furthermore, DQR incorporates the proposed TMS in Chapter 3 to ensure reliable data transfer and avoid malicious paths.

DQR defines two Q functions $Q^{A(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t)$ and $Q^{B(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t)$ as the estimated future reward of an agent i at state s_t taking the action a_t as shown in Eq. 5.2 and Eq. 5.4. Each of these estimators is updated using a value from the other estimator for the next state, as shown in Eq. 5.1 and Eq. 5.3. Therefore, the actions a^*_t and b^*_t are the maximum valued actions for $Q^{A(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t)$ and $Q^{B(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t)$, respectively.

$$a^{*(i)}_t = \underset{a^{(i)}_t \in A}{argmax} Q^{A(i)}_t(s^{(i)}_{t+1}, a^{(i)}_t) \tag{5.1}$$

$$Q^{A(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t) \leftarrow (1-\eta) Q^{A(i)}_t(s^{(i)}_t, a^{(i)}_t) + \eta[r^{(i)}_{t+1}(s^{(i)}_{t+1}) + \gamma Q^{B(i)}_{t+1}(s^{(i)}_{t+1}, a^{*(i)}_t)] \tag{5.2}$$

$$b^{*(i)}_t = \underset{a^{(i)}_t \in A}{argmax} Q^{B(i)}_t(s^{(i)}_{t+1}, a^{(i)}_t) \tag{5.3}$$

$$Q^{B(i)}_{t+1}(s^{(i)}_t, a^{(i)}_t) \leftarrow (1-\eta) Q^{B(i)}_t(s^{(i)}_t, a^{(i)}_t) + \eta[r^{(i)}_{t+1}(s^{(i)}_{t+1}) + \gamma Q^{A(i)}_{t+1}(s^{(i)}_{t+1}, b^{*(i)}_t)] \tag{5.4}$$

where $\eta \in [0, 1]$ is the learning parameter where small values decelerate the learning, and large ones may prevent algorithm convergence, $\gamma \in [0, 1]$ is the future reward discount parameter where small values make the learning agent nearsighted by considering the only immediate reward.

DQR is designed to always choose the most reliable shortest path by defining the reward function as punishment, as shown in Eq. 5.5. The delivery reliability is achieved by incorporating trust information, which is evaluated using Algorithm 2, while the punishment design reduces the number of transmissions along the path to the destination to ensure an energy-efficient protocol. Moreover, energy information from the agent itself

is also considered to optimize the network lifetime, which will be discussed further in Section 5.4.3

$$r_{t+1}^{(i)}(s_{t+1}^{(i)}, j) = \begin{cases} -(1 - T_t^{(ij)}).F_t^{(i)} & if \quad O_t^{(ij)} \neq \{\phi\} \\ -(1 - T_{t-\delta}^{(ij)}).F_t^{(i)} & if \quad O_t^{(ij)} = \{\phi\} \wedge |O^{(ij)}| > \epsilon \\ 0 & Otherwise \end{cases} \tag{5.5}$$

where $r_{t+1}^{(i)}(s_{t+1}^{(i)}, j)$ is the received reward by node i for taking the action $a_t^{(i)} = j$ forwarding the traffic to the neighbor j at time window $[t, t + \tau]$, $T_t^{(ij)}$ is the trust value of node j that maintained by node i at time window t and is evaluated using Algorithm 2, $O_t^{(ij)}$ is the direct observations maintained by node i for node j at time window t, δ is a time lag used to obtain the last trust value, ϵ is a threshold to identify the minimum required evidence where higher values means more historical data is required to use the evaluated trust value.

The learning process must be continual due to network dynamicity and distributed as no agent has a full view of the network. DQR is a decentralized protocol where the learning agents exchange their best estimations with their neighbors, as illustrated in Algorithm 6. The received estimations are then used to update the Q^A and Q^B tables and specify the most optimal next hop. As the goal of the learning agent is to maximize the received reward in the long run, greedy action should not always be taken as routing task is a continual online task, and exploiting the greedy action all the time prevents the convergence to the global optimum. Therefore, DQR uses ε-greedy method to balance between exploration and exploitation. The learning agent explores the environment with a probability of θ and exploits it with a probability of $(1 - \theta)$. Initially, the learning agents have no evidence from the network; hence their Q values are initialized to zeros, which is more practical to motivate the agents to explore the environment and does not require any hardware or positioning information like in [45, 44].

5.4.2 Synchronous and Asynchronous Updating

Both synchronous and asynchronous updating methods are used in DQR. The synchronous Q table updating method is used to produce a lightweight routing protocol, while asynchronous is used to ensure fast convergence. Each action-value function is updated with the outcome of the other action-value function as shown in Algorithm 7. The actions $a^{*(i)}$ and $b^{*(i)}$ are the maximum value action in state s_{t+1} for $Q^{A(i)}$ and $Q^{B(i)}$,

respectively. Therefore, both Q tables are updated for the same problem but with a different set of evidence to produce an unbiased estimate for all action-value(s). Although the obtained experience is divided between two action-value functions, the algorithm is still data-efficient as selecting the optimal action is computed based on the average Q tables as illustrated in Algorithm 6. As the learning agents collaborate with each other by broadcasting their best estimation to a destination, this information is then used to keep the Q tables updated. However, the learning agent forwards the traffic to only one adjacent node during the time window t, and thus it can only calculate the reward for this action. For instance, node i take the action $a_t^{(i)} = j$ during the time window t and receives two updates from nodes j and k. Consequently, DQR updates the action-value of j with the calculated reward using double Q-learning, while it checks if there is enough evidence about node k to update each action-value separately using Q-learning or keep it unchanged in case of no enough evidence. This method allows DQR to react quickly to any environment change, and at the same time, it immunizes DQR against utilizing false updates from malicious nodes.

On the other hand, although the synchronous updating method is computationally efficient, it may decelerate the convergence as the learning agent, especially in the exploration phase, may make wrong decisions and thus keep forwarding the traffic to the wrong next hop. Furthermore, in traditional RL mode, the learning agent risks losing one packet each time to update the Q tables. However, using only synchronous updating, more packets may be lost before updating the Q tables. This usually happens when loops occur. Therefore, DQR uses an asynchronous updating method to step up the learning process and make the algorithm converge swiftly. Once a loop is detected or expected, such as when forwarding the packet to its source again, the asynchronous updating method is triggered to penalize both corresponding action-value(s) and allow the learning agent to take the appropriate action accordingly, as detailed in Algorithm 8.

5.4.3 Energy Model

Optimizing the network lifetime is still a challenging concern in WSN and WMSN in particular. Due to the critical applications of WMSN, dead nodes may have catastrophic consequences. Moreover, in some cases, replacing the battery may need surgical intervention. Considering the residual energy of the adjacent nodes is widely used to maximize the overall network lifetime [208, 209]. However, exchanging energy information between adjacent nodes is neither energy nor computational efficient. In contrast, DQR only

uses local energy information with a view to reducing the computational overhead and avoiding filtering out false second-hand information. Moreover, it uses two sources of energy information with a view to load balancing energy consumption across the network. When the residual energy percentage is greater than a threshold ϑ, this parameter does not contribute in evaluating the consumed energy ratio $E_t^{(i)} \in [0,1]$ as shown in Eq. 5.6. In that case, SNs choose the most reliable shortest path, which in turn makes some nodes overloaded due to their trustworthiness and positions. Therefore, DQR defines the energy consumption ratio $C_t^{(i)}$ to evaluate the extra burden incurred by nodes due to relaying activities, as shown in Eq. 5.7. The weighted average of $E_t^{(i)}$ and $C_t^{(i)}$ is calculated in Eq. 5.8. As integrating the energy into the reward function may influence the nodes routing decision to choose a malicious path, the energy factor is bounded by $\lambda \in [0,1]$ as shown in Eq. 5.9.

$$E_t^{(i)} = \begin{cases} 0 & if \quad \frac{e_{res}(t)}{e_{init}} > \vartheta \\ 1 - \frac{e_{res}(t)}{e_{init}} & Otherwise \end{cases} \tag{5.6}$$

$$C_t^{(i)} = 1 - \frac{c_n(t)}{c_a(t)} \tag{5.7}$$

$$\psi_t^{(i)} = \omega E_t^{(i)} + (1-\omega) C_t^{(i)} \tag{5.8}$$

$$F_t^{(i)} = e^{\lambda \psi_t^{(i)}} \tag{5.9}$$

where $e_{res}(t)$ is the remaining energy at time t, e_{init} is the initial energy, ϑ is the residual energy threshold, $c_n(t)$ is the node normal energy consumption rate, $c_a(t)$ is the overall energy consumption rate, ω is the average weight, λ is the bound parameter where $\lambda = 0$ is used to disable the energy module.

5.5 Evaluation and Performance Results

In this section, our proposed DQR is analyzed under different conditions. Various simulation scenarios have been run to prove the merit of DQR.

5.5.1 Experimental Setup

A WMSN for a ward in a field hospital, as discussed in Chapter 4, has been adopted. The SNs have been distributed randomly over an area of $50m \times 10m$. A total number of 64 SNs has been used where one of them acts as a sink. The traffic is randomly generated using an exponential distribution density function.

DQR is benchmarked with QRT [45] routing protocol. QRT has been chosen as a benchmark for the several reasons. First, it is a learning-based routing protocol mainly proposed for WMSN. It uses Q-learning to choose the optimal path to the destination. Second, it is designed to avoid non-cooperative and misbehaving SNs by incorporating a trust management scheme to evaluate the trustworthiness of available routing paths. Most importantly, QRT is the only re-producible routing protocol, which is learning based and uses trust management to deal with dropping attacks as discussed earlier in Section 4.3. To ensure fair comparisons, we adopted the reported parameters setting of QRT as shown in Table 5.1. The experiments have been run using a discrete event simulator based on Simpy [200]. The simulation time is set to $200s$ where the first $50s$ represents the learning time. The exploration-exploitation rate is controlled by ε-greedy strategy and set to 10% as in QRT. Each experiment has been run 30 times to ensure the Gaussian distribution. The results are then averaged out and reported with one standard deviation.

Table 5.1: DQR Simulation Parameters

Parameter	**Value**
Application	Poisson random traffic
Traffic rate μ	1, 2, 4, 8
Radio range	5m
Propagation loss model	Range propagation loss
Number of SNs	64
Time unit	1s
Simulation time	200s
Learning period	50s
Learning rate η	0.5
Discount factor γ	0.5
$\varepsilon-$greedy	0.1
The average weight ω	0.5
Residual energy threshold ϑ	0.7

5.5.2 Double Q-learning Convergence

In this experiment, the convergence speed of both Q-learning and double Q-learning is compared. The delivery performance has been compared between our work (RRP) in Chapter 4 and DQR routing protocol. The energy model of DQR has been deactivated by setting $F_t^{(i)}$ always to one since RRP does not take into account energy parameter. This modification is necessary to ensure a fair comparison between both algorithms as enabling the energy model influences the routing decisions with a view to enhancing the overall network lifetime. Fig. 5.1 shows how both algorithms converge under blackhole attack where 50% of the nodes are malicious and the traffic rate is $\mu = 1$. The most important finding is both algorithms converge to the same value, and this is expected as both algorithms are proven to converge [210, 206]. However, these results validate our work in Chapter 4. The second important finding is that DQR shows better performance before converging, which is attributed to using double estimators, ensuring unbiased routing decisions.

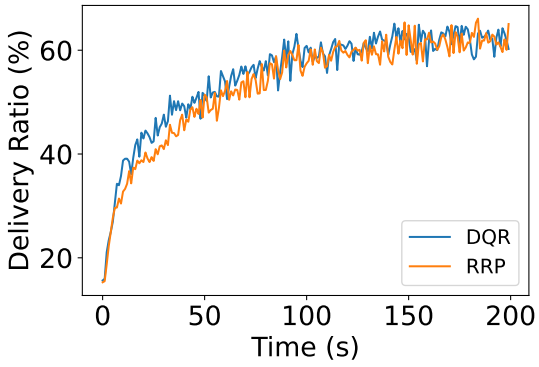


Figure 5.1: Double Q-learning Convergence

5.5.3 Delivery Reliability Analysis

In these experiments, the delivery performance is evaluated under different network conditions ranging from normal operation to under complicated attacks. In addition, the packet delivery ratio and hop counts are considered to compare the optimality of the routing decisions made by both protocols, DQR and QRT.

The first experiment studies the performance under normal operation with variable traffic rates, starting at $1p/s$ and doubling it each time. No malicious SNs are considered in this experiment. Benign nodes may drop randomly 1% of the traffic. Fig. 5.2a and 5.2d show the delivery ratio and the hop counts for both protocols under normal

operation, respectively. DQR shows superior data delivery performance with optimum routing decisions. On the other hand, QRT shows high variability in terms of delivery ratio and hop count, which indicates that QRT does not converge to the optimum action-value(s) all the time. Moreover, it finds difficulty working under low traffic rates.

In the second experiment, blackhole and selective forwarding attacks are launched during the simulation to study the robustness of both protocols. Both attacks may disrupt the network operation. Therefore, nodes should always choose the most reliable path to destinations. The performance has been evaluated for a variable number of malicious nodes, starting from 1 malicious nodes and up to 50% of the total number of SNs. Fig. 5.2b, 5.2c, 5.2e and 5.2f show the delivery ratio and the hop counts under blackhole and selective forwarding attacks, respectively. Across all scenarios, DQR chooses the most optimal reliable paths, as illustrated in the hop counts results. When the number of malicious nodes increases, DQR avoids malicious paths and tends to choose longer but reliable paths. On the other hand, QRT is not able to detect malicious paths, as shown in the decreasing hop counts when introducing more malicious nodes. This means that packets end up in malicious nodes, which explains the low delivery ratio.

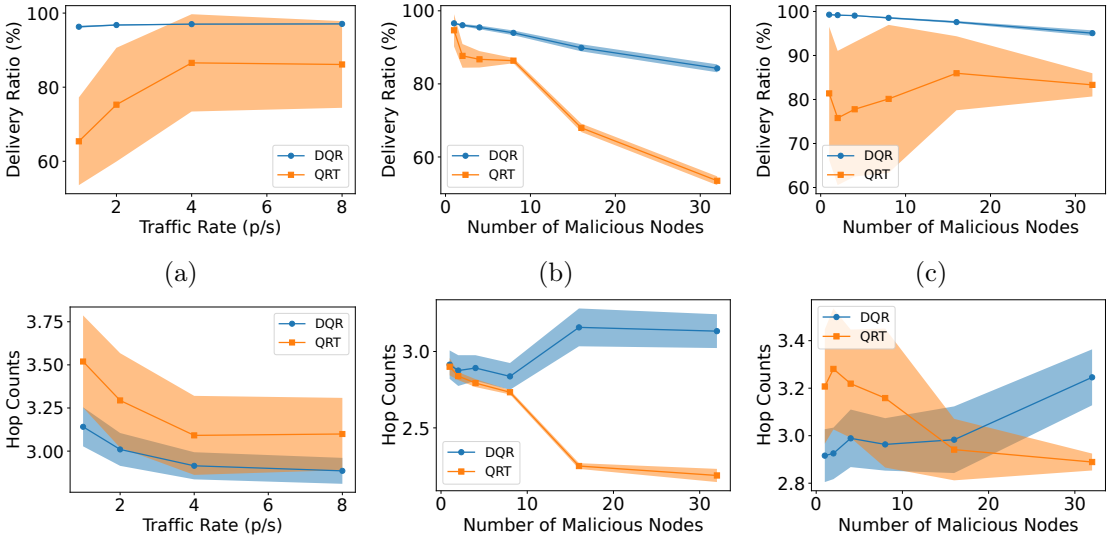


Figure 5.2: The average delivery and hop counts ratios under different conditions

In the third scenario, sinkhole, which is a route poisoning attack, is launched to study the impact of receiving dishonest updates from other agents on routing decisions. Different levels of poisoning are evaluated starting by increasing the updates by 25% and doubling

it up to 100%. The delivery and hop counts ratios are illustrated in Fig. 5.3a - Fig. 5.3f. The results show that DQR is robust under different poisoning levels and can achieve a high delivery ratio. It is worth noting that in the worst-case scenario, when malicious SNs advertise zeros, DQR takes slightly longer paths as the received false updates influence not only the SN itself but also its neighbors. However, this behavior does not affect the delivery ratio.

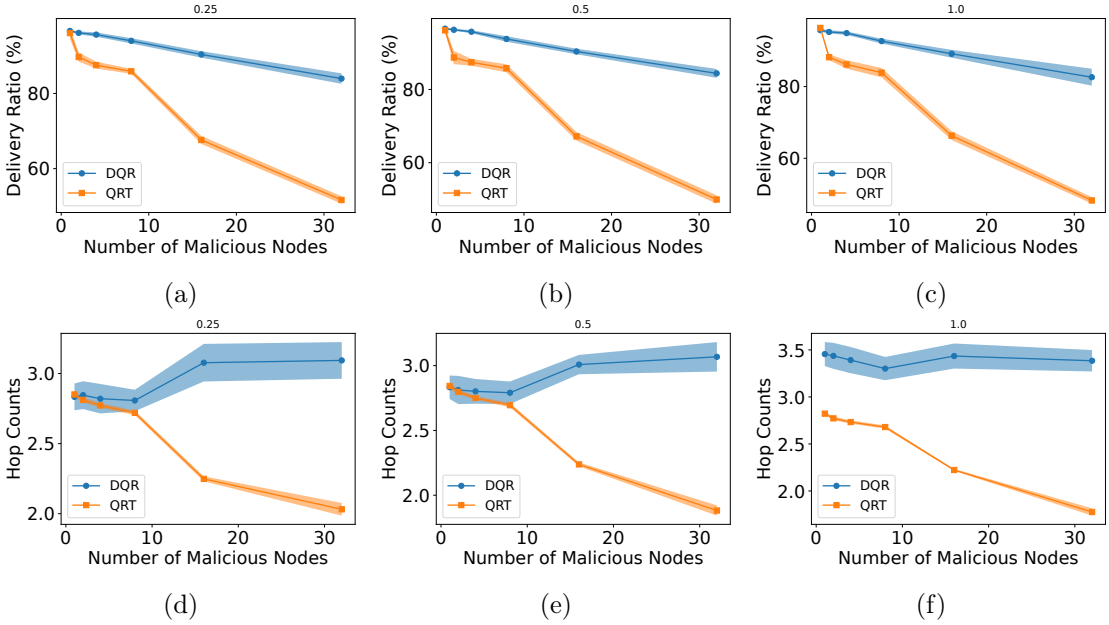
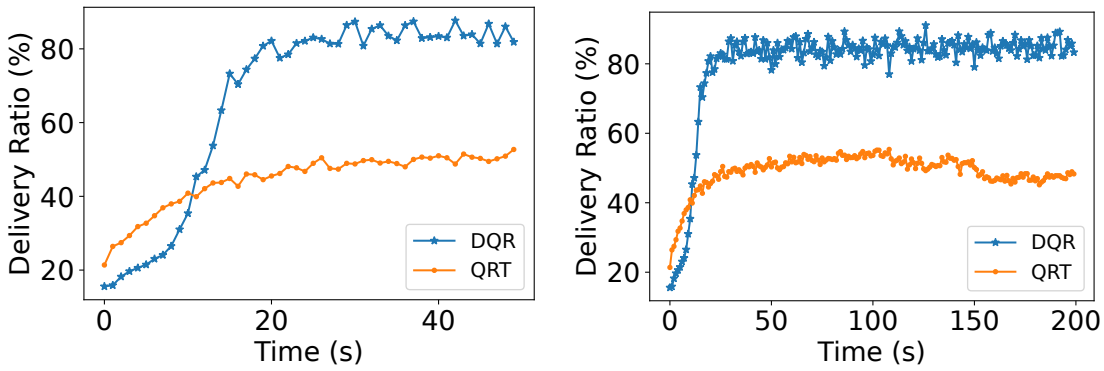


Figure 5.3: The average delivery and hop counts ratios under sinkhole attack

5.5.4 Convergence and Mobility

Q-learning is proved to converge to the optimum action-value(s) [210], as is double Q-learning [206]. However, convergence time is a crucial factor. A longer time to converge implies risking more packets to lose and consuming extra resources. In this experiment, the convergence time is evaluated in two scenarios, at the beginning of the simulation and when patients change their locations. Fig. 5.4a demonstrates the convergence time at the beginning of the simulation, where SNs have no information about the environment and need to explore the environment in order to converge. DQR converges faster than QRT thanks to its asynchronous updating algorithm. It takes less than 50% time to converge compared to QRT. It is worth noting that QRT performs well at the early start because it is provided by positional information, while DQR works without any prior knowledge. In the second scenario, patient mobility is considered within the hospital ward. Assuming



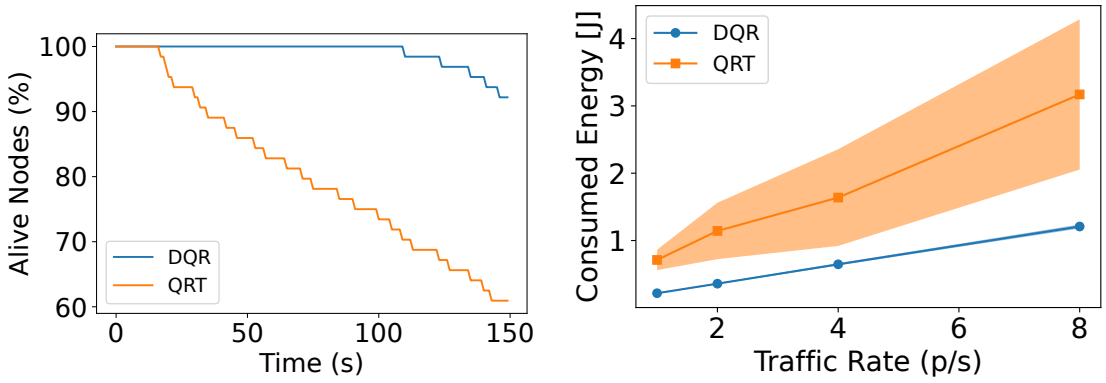
(a) (b)

Figure 5.4: The average convergence time at the begining of the simulation is shown in 5.4a, while in 5.4b the average delivery ratio is shown when three patients change their locations inside the ward at 100s and 150s

that the patient could have up to three SNs. Thus, the simulation is run for 1, 2, or 3 randomly chosen SNs at a time. Two patient's movements have been simulated at times 100s and 150s. The simulation has been run for a hostile environment where 50% of the nodes are launching blackhole attacks. Fig. 5.4b shows mobility results for only three SNs for clarity as it represents the worst case. The results show a fast convergence without any noticeable performance degradation for the DQR protocol, which proves the robustness of our methods. On the other hand, QRT suffers from difficulty in re-converging, especially after the second movement.

5.5.5 Energy Efficiency

The energy efficiency has been evaluated in two experiments, by modelling the energy consumption as explained in Section 5.4.3 in our simulations. In the first, the network lifetime has been compared between both protocols. The second scenario shows the average consumed energy by a node for different traffic rates. Network lifetime could be defined as the running time until a node dies [193]. Both simulation scenarios have been carried out under normal operation without introducing any attack. Fig. 5.5a shows the percentage of alive nodes during the simulation. QRT has a very short network lifetime compared to DQR. The first node dies after around 16s on average. This deficiency could be attributed to two reasons. First, QRT does not take any energy-related factor into account to choose the optimal path, and most importantly, the excessive information exchanging increases the RF activities significantly, which is responsible for 80% of the



(a) (b)

Figure 5.5: The average alive SNs ratio is shown in 5.5a, while the average consumed energy for variable traffic rates is shown in 5.5b

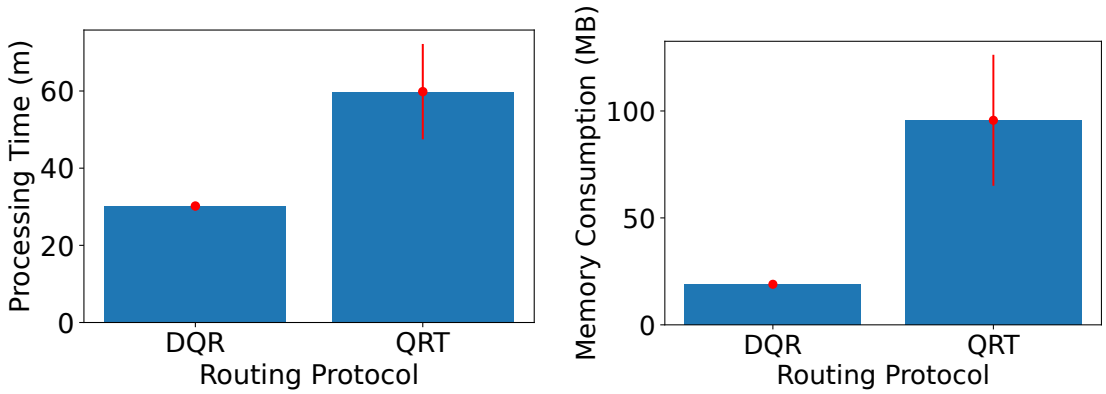
consumed energy. On the other hand, DQR shows superior performance because of its resource-conservative design, which is clearly reflected in consuming less energy for all traffic rates, as obviously seen in Fig. 5.5b.

5.5.6 Computational Overhead

In this experiment, both protocols' processing and memory overheads are evaluated. The experiment was carried out on an Intel Core i9-10885H at 2.4GHz and 32GB RAM. The computational overhead has been evaluated for traffic rate $\mu = 4p/s$ as QRT performs poorly for traffic rates less than $4p/s$ as shown in Fig. 5.2a. No attacks were launched during the simulation. The simulation was repeated 30 times, and then the mean with one standard deviation was reported.

The average processing time of both protocols is illustrated in Fig. 5.6a. The results show a minimum processing overhead of DQR. It saves around 50% processing overhead compared to QRT. This proves that our novel RL model is resource-efficient. Moreover, unlike QRT, DQR has minimum variability. The low variability of DQR indicates that DQR is always able to converge swiftly without any difficulties, proving its robustness.

Memory consumption is another crucial factor for constrained devices, such as SNs. Fig. 5.6b depicts the average consumed memory of both protocols. During the simulation, the memory allocations were traced using a memory allocation module called tracemalloc [203]. The results show that DQR is able to save up to 80% of QRT consumed memory. Moreover, unlike QRT, DQR shows almost no variability, which indicates its robustness.



(a) (b)

Figure 5.6: The average processing time and memory consumption

The results of this experiment show that DQR has a minimum footprint in terms of processing time and consumed memory. This lightweight computational overhead could be attributed to its resource-efficient design represented in synchronous and asynchronous Q tables updating algorithms. Furthermore, this novel design is able to converge swiftly with minimum variability allowing the packets to reach their destinations efficiently.

5.6 Conclusion

The resource scarcity and the sensitive applications have brought enormous challenges to WMSN routing protocols. The existing routing protocols for WSN cannot be directly adopted for WMSN due to overlooking some imperative requirements. In this chapter, a lightweight, reliable, and energy-efficient routing protocol for WMSN has been proposed. DQR is a double Q-learning routing protocol that uses the proposed RL model in Chapter 4. It uses two updating methods combined with trust management and energy models to ensure lightweight, reliable, and resource-efficient data delivery. The experimental results show superior performance with minimal resource footprint.

Chapter 6

A Trust Management Module for NS3 Simulator

Trust management offers a further level of defense against internal attacks in WMSN and ad hoc networks in general. Deploying an effective trust management scheme can reinforce the overall network security and enhance routing decisions. Despite limitations, however, security researchers often use numerical simulations to prove the merits of novel methods. This is due to the lack of an adequate testbed to evaluate the proposed trust schemes. Therefore, there is a demanding need to develop a generic testbed that can be used to evaluate the trust relationship for different networks and protocols. In this chapter, an NS-3 module consisting of three main components to evaluate the different trust relationships: direct trust, uncertainty, and indirect trust, is proposed. It is designed to meet usability, generalisability, flexibility, scalability, and high-performance requirements. A series of experiments involving 1680 simulations were performed to prove the design and implementation accuracy of TrustMod. The performance results show that TrustMod's resource footprint is minimal, even for very large networks.

The main findings of this chapter were presented at the IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) [211].

6.1 Introduction

Abundant research is put forward to evaluate the trust relationship using different approaches and techniques [212]. However, the majority of these trust schemes are evaluated using numerical simulations due to the shortage of a dedicated simulator or a testbed to simulate the trust relationship. Trust evaluation using numerical computing environments such as MATLAB can only provide an abstract mathematical and numerical analysis for the proposed trust schemes. However, these unrealistic experiment setups cannot reflect the actual network operation where tens of protocols from different stack layers work collaboratively. For instance, traffic rates, packet re-transmissions, collisions, and routing protocols are all examples of network operating conditions and protocols that cannot be fulfilled using existing numerical analysis tools, including MATLAB. This chapter, therefore, presents TrustMod, a new testbed environment for NS3 [185] to realistically simulate the trust relationship on ad hoc networks, which WMSN is part of.

Among a wide range of network simulators, NS-3 [185] has been chosen to build this testbed for several reasons. First, it is an open-source network simulator designed essentially for research. It provides a robust core written in C++ with high compatibility and scalability characteristics. Unlike NS-2, which uses OTcl to write the simulation scenario, NS-3 uses C++ with Python bindings that allow researchers to import NS-3 libraries as Python modules. Moreover, the performance analysis of NS-3 shows an optimal trade-off between memory consumption and simulation run-time among different network simulators [213].

6.2 Contribution

The main contribution of this chapter is introducing a trust management module for NS-3. Our proposed NS-3 module, TrustMod, consists of three main components which are direct trust evaluation, uncertainty evaluation, and indirect trust evaluation. It has been designed to be resource efficient and easily integrated as other NS-3 modules. Furthermore, the code of our proposed trust management module is made available on GitHub.

6.3 Related Work

Many different trust schemes are proposed in the literature for different wireless networks and applications. However, due to the unavailability of more realistic trust evaluation simulators or testbeds, MATLAB is still the first choice for researchers to evaluate their schemes' effectiveness. Few tools are introduced in the literature to evaluate trust schemes, such as TOSim [214] and TRMSim-WSN [215]. In [214], the authors introduced a tool to evaluate various types of trust and reputation schemes targeting overlay networks with four threat models. It is a scope-specific tool that is not applicable to other networks. In [215], the authors introduced a simple java-based trust simulator for Wireless Sensor Networks (WSNs). This simulator allows the researchers to tune several parameters, such as the number of nodes and delay, to simulate different kinds of malicious activities. However, as the protocol stack is not implemented in TRMSim-WSN, it is regarded as a conceptual simulator that cannot reflect the realistic network's behavior. The main shortcomings of using the aforementioned tools are the incapability of simulating the targeted network operating conditions and the inability to integrate with other stack layers; therefore, the proposed trust schemes cannot be evaluated under different stack protocols, such as routing protocols. Table 6.1 shows the most used tools in the literature to simulate proposed trust schemes.

On the other hand, TrustMod is a generic trust management testbed built as a module for NS-3. It works as a cross-layer module in the TCP/IP stack. Therefore, to the best of our knowledge, it is the first trust management testbed fully integrated with the protocol stack and can be used with different networks and under different network conditions and parameters. It can be added like any other built-in modules to the nodes, which are the conceptual computing devices in NS-3. Researchers can use the NS-3 attributes system to configure the required trust management properties and then write their scheme implementation.

6.4 Network Simulator 3

For decades, NS-2 [136] was regarded as the de-facto standard simulator for research. Countless published research papers are evaluated using NS-2. In 2006, a project to develop a new network simulator to replace NS-2 was begun. NS-3 is built from scratch using C++, although some models are ported from predecessor simulators such as NS-2, YANS [216] and GTNetS [217]. The initial release of NS-3 was available in 2008, while the development and maintenance of NS-2 stopped in 2010 [218]. The main goal was to

Table 6.1: The used simulators in the literature to evaluate trust schemes

Simulator	Trust Module	Networks	Comments
MATLAB	No	N/A	It has been widely used in the literature. Not a network simulator. It provides numerical analysis.
TRMSim-WSN [215]	Yes	WSN	A java based trust simulator to evaluate trust schemes for WSN. It is a scope specific simulator with a few parameters to adjust. The protocol stack is not implemented in TRMSim-WSN.
TOSim [214]	Yes	Overlay networks	A scope specific simulator that cannot be used for other networks.
NS-2 [136]	No	Generic	An old simulator. Development has been discontinued. It does not have trust module. Researchers need to implement their schemes and analyses the trace files to obtain the results.
NS-3 [185]	No	Generic	A research oriented simulator. It does not have a trust module. Researchers need to implement their schemes and analyses the trace files to obtain the results.

enhance the NS-2 models' realism by making it closer to how real computers operate. For instance, NS-3 adopts the Linux architecture for sockets and internal interfaces. Moreover, NS-3 supports emulations by incorporating real network devices to form a real testbed.

NS-3 is built as software libraries that can be linked to the simulation scenario statically or dynamically. Fig. 6.1 describes how the NS-3 modules are organized. Module dependencies may usually exist with other underneath modules. The core module of NS-3 consists of C++ classes that provide time services, smart pointers, callbacks, debugging facilities, and other significant services. These services are used by all kinds of hardware, protocols, and environmental models. The list below contains other important NS-3 components:

- Network module, which models the network packet, packet tags, and packet headers. Moreover, the node class and the abstract base class netdevice are defined in this module, in addition to addressing types such as IPv4 and MAC.
- Mobility module, which has different mobility models such as static, random, and walk.

- Helper API contains classes and methods that provide high-level wrappers to encapsulate low-level API calls. It is widely used when scripting a simulation scenario.

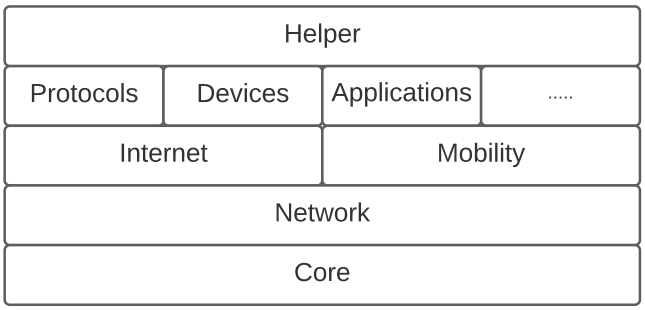


Figure 6.1: NS-3 Modules Architecture

NS-3 allows researchers to evaluate their simulation models using different kinds of tools and systems. A logging facility is used to monitor and debug the execution of the simulation scenario. It can be enabled from the simulation script for different NS-3 components using the environment variable NS_LOG. Logging messages are classified into different severity classes, where the user can set to see the log of a specified severity class for a specified NS-3 component. The tracing system is another tool to allow users to gather information and statistics to evaluate their simulation models or modifications. NS-3 defines two independent components, tracing sources and tracing sinks, where the connection between the sources and the sinks can be established with the help of the attributes and the callback mechanisms. NS-3 provides two kinds of tracing output. The first is similar to NS-2, where all events associated with their properties, like timestamps, are output to text files. The second tracing output is PCAP binary files for capturing live traffic. PCAP files can be analyzed by TCPDUMP or Wireshark applications. FlowMonitor is an NS-3 module for monitoring traffic flows contributed by authors in [219]. It provides an easy tool to monitor the traffic across the network. It allows researchers to measure the performance of their methods by providing different kinds of statistics such as packet loss rate, delays, and bit rates. Furthermore, although NS-3 does not have a built-in graphical animation tool, an offline animator toolkit can be used for visualization. NetAnim is an example of those tools [220]. It uses the generated XML trace file for animation. Animation includes but is not limited to transmitted packets over different links, packets timeline with filtering capabilities, IP and MAC information, and routing tables.

6.5 The Proposed Trust Management Module

In this section, our trust management module for NS-3, termed as TrustMod, is introduced. The design requirements are first identified, and then the module structure and its components are presented.

6.5.1 Design Requirements

Different objectives are taken into account when designing our trust management module for NS-3. These objectives cover various aspects such as usability, generalisability, flexibility, scalability, and performance.

Usability is the first objective of our TrustMod module. Writing the simulation scripts is already a time-consuming and complicated task. Long times are spent on writing simulation scenarios and analyzing trace files. Therefore, the trust management module must be easy to use. This gives the researchers more time to spend on their proposed methods. The researchers can enable the trust management module for any node by adding a few code lines. Moreover, they can instantiate different trust instances for different nodes. The module design should also comply with other NS-3 modules; thus, the researchers can easily configure and set the module instance attributes using the same way as other modules.

Trust management systems may generally rely on two components, direct and indirect trust evaluation. In direct trust evaluation, the trustor evaluates the trustee based on direct observations. However, when the trustor does not have sufficient observation history, it can ask for recommendations from other network entities. This approach is widely adopted in the literature [169, 170, 145, 221]. However, obtaining recommendations is proved to be a time, and resource-consuming process [174]. Therefore, the trust evaluation must comprise three main components. The direct trust evaluation process is based on direct observations. The indirect trust evaluation, which is based on received recommendations from other entities in the network, in addition to introducing an uncertainty evaluation component that can manage the process of asking for recommendations with a view to preserve resources.

The trust module must work with different protocols and applications. Therefore, researchers can evaluate their methods for different networks and under different conditions. In addition, the distributed approach provides more flexibility to support other services and protocols within the node itself, such as secured routing protocols and access control

mechanisms.

Obviously, deploying a trust module will consume some resources. However, memory and processing overhead must be as low as possible to significantly maintain high simulation performance when increasing the number of nodes. Therefore, the simulation time and the memory footprints are expected to be minimal.

It is imperative that the proposed trust module can output all the results and statistics in a readable and easy-to-use format. This output data has to be stored by the end of the simulation process and retrieved later by researchers for results analysis. Several output format candidates are available such as binary files, ASCII files, XML/JSON files, or databases. Choosing the appropriate file format that allows the results to be stored and retrieved efficiently is essential. TurstMod uses ASCII format for output, which provides an effortless and readable way to obtain the simulation results. The trust results are formatted in a way that allows researchers to import them to Excel readily.

6.5.2 Design Overview

NS-3 network entities are defined using Node class, which is a conceptual model that other objects can be aggregated to it. The implementations of TCP/IPv4, TCP/IPv6, and other related protocols are available in the internet stack module, which can be installed into each node using the helper class. Sending and receiving packets using this module goes through different layers from NetDevice to Application classes. Trust management aims to monitor the behavior of others. Therefore, cross-layer information should be received from different sources. This cross-layer architecture allows TrustMod to receive all the needed information from other stack protocols. The received information is then processed and stored in a dynamic data structure, which will be presented in the next section. Fig. 6.2 provides an overview of the TrustMod structure showing the implementation of receiving the cross-layer information from both sending and receiving paths.

The TrustHelper class is used to initialize the trust module and aggregate it to the node. Two cross-layer information sources from layers two and three are used to evaluate the trust relationship. The first is NetDevice from the packets sending path. The SentInput method is called, and the transmitted packet and the destination MAC address are passed to it. In order to receive cross-layer information of the forwarded packets from layer 3, the promiscuous mode must be enabled. Therefore, the SetPromiscReceiveCallback has to be registered on the NetDevice in addition to writing the implementation of the method

PromiscReceive. This allows layer 3 to receive the sniffed packet from lower layers, which in turn passes it with the source MAC address to the ForwardedInput method.

The observed interactions form sequences of discrete-time data. This period is set by a predefined attribute called TimeUnitAttribute, which can be set from the simulation script. This attribute timely controls all the operations inside the Trust module. Therefore, trust evaluation is scheduled at the end of each time unit by calling the method TrustSchemeManager in the TrustScheme class. Researchers have to write the implementations of their trust schemes in the following methods:

- DirectTrustEvalaution method: This method is used to evaluate the trust value based on direct observations.
- UncertaintyEvalaution method: This method is introduced to specify when to consider second-hand information from other neighbors.
- RecommendationsEvalaution method: This method is used to evaluate the received recommendations and filter out those untrustworthy ones.
- OverallTrustEvalaution: This method is called once having all the required information in order to evaluate the overall trust value.

TrustMod inherits from the Application class to allow sending and receiving recommendations. Once TrustMod is installed using TrustHelper, it initializes the recommendations listener in order to receive both recommendation requests from other neighbors and the expected recommendation responses for the recommendation requests sent by the node itself. The simulation script specifies the listening port by setting the predefined attribute *RecommendationListenerPortAttribute*. A receive callback is defined for the receiving socket. Therefore, received packets are processed in the *RecommendationReceive* method in order to differentiate between requests and responses. Based on the received packet type, a recommendation processor is called from the *TrustScheme* class. Fig. 6.3 and Fig. 6.4 show the recommendation packet format for both request and response types. Both have packet type, trustee MAC, trustor IP address, and TrustNode (TN) sequence number. The packet type field is used to differentiate between request and response packets, while the trustee MAC field is used to specify the node in question. Trustor IP is set to the IP address of the recommendation sender. The TN sequence number represents the sequence number of the TN when the trustor is uncertain about the trustee's trustworthiness and needs second-hand information to evaluate the overall trust value. It is used to indicate to the TN object where the second-hand information

is needed, and its value will be used in the recommendation response packet. On the other hand, the recommendation response has two more fields. The first contains the trust value, while the second represents how certain the recommender is about this trust value.

6.5.3 Trust Module Data Structure

TrustMod receives cross-layer information and recommendation responses during the simulation process. This information is usually processed at the end of the time unit. Moreover, researchers expect detailed statistics and results at the end of the simulation. Therefore, the aforementioned information needs to be saved efficiently during the simulation. Two classes are defined for this purpose. The first is the TrustNode (TN) class to save the observations, while the second is the ReceivedRecommendations (RR) class to save the received second-hand information.

TrustMod instantiates a TN object for each time unit to save the observations and other related information. Therefore, an efficient data structure is required to store a series of TN objects for each trustee. In TrustMod, a map of vector pointers is adopted as shown in Fig. 6.5. The map key is set to be the trustee MAC address, while the mapped value is a pointer to a vector of TN objects. Each TN has different attributes to save observations, such as m_forwardedPackets and m_droppedPackets. In addition to the primitive attributes, there are two objects. The first is used to track the trust scheme parameter changes, while the second is used to store the received recommendations. As the recommendation request is broadcasted over the network, it is expected to receive recommendations from multiple nodes. These recommendations are stored for further processing to filter out dishonest ones. Therefore, TrustMod uses a vector of ReceivedRecommendations object pointers to store the received recommendations. Adopting this data structure, as mentioned earlier, allows a dynamic memory allocation. Moreover, using pointers makes the management of data structures more efficient by saving memory and reducing the module's complexity.

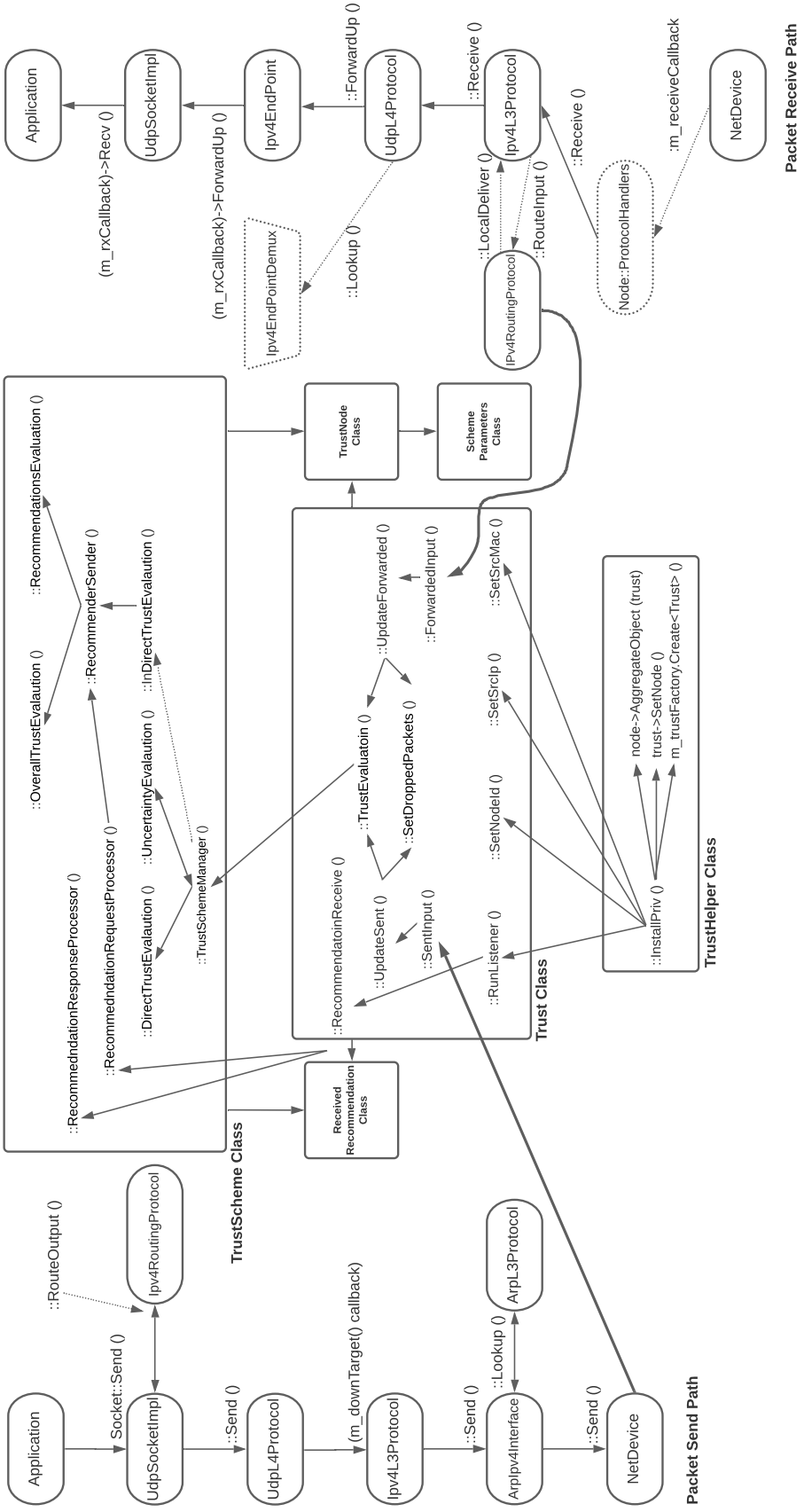


Figure 6.2: Trust Module Overview

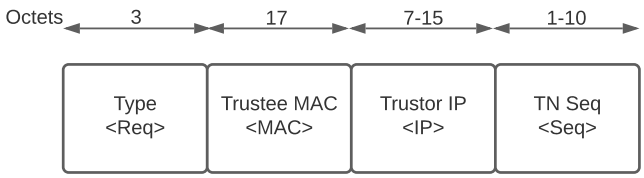


Figure 6.3: Recommendation Request Format

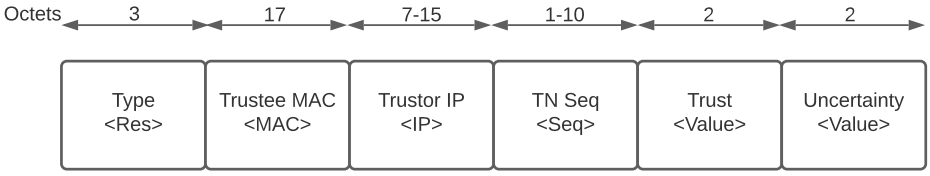


Figure 6.4: Recommendation Response Format

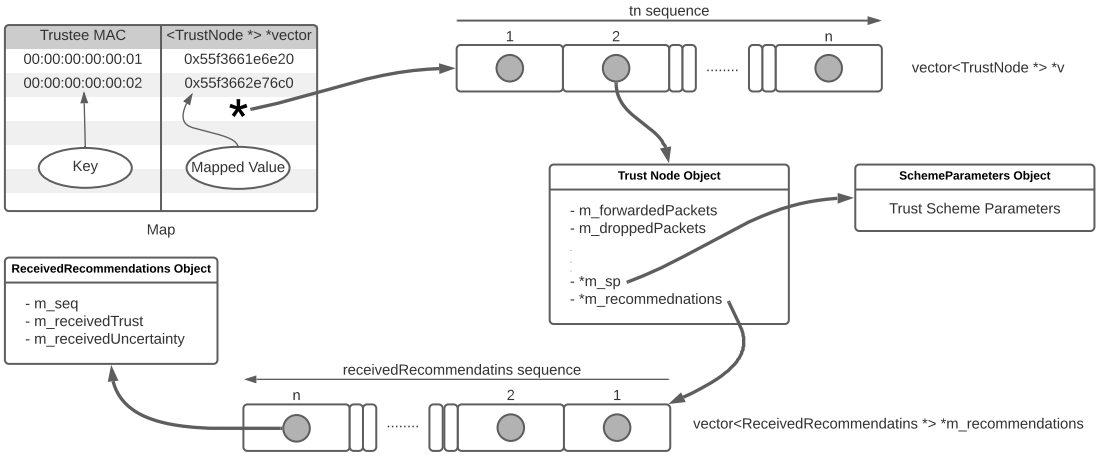


Figure 6.5: Trust Module Data Structure

6.5.4 Trust Module Implementation

The proposed trust management module is implemented using six classes. The Trust class represents the core of the module. It is responsible for initializing the trust module for each node, linking with other stack layers, listening for recommendation requests, and processing the observed information. TrustHelper class provides a separate API on top of the core NS-3 API. This helper class makes the use of the TrustMod module easier by creating and configuring the trust module effortlessly. Moreover, TrustMod has two additional classes for modeling the data structure of the module. The first is TrustNode class, which stores all the statistics and trust information for a one time

unit during the simulation. It provides the necessary attributes and methods to store and evaluate the trust relationship. The second is ReceivedRecommendations class, which stores and processes the received recommendations for each sent recommendation request. On the other hand, researchers propose different methods to evaluate the trust relationship, which usually introduce new parameters and mechanisms. Therefore, two classes are used to integrate the researchers' methods into the trust module. The first is SchemeParameters class, which can be used to define scheme proprietary parameters. The second is TrustScheme, which provides the core of trust evaluation. This class is designed to have all the evaluation operations, such as processing the two kinds of recommendation requests and response packets. Moreover, four methods are provided for researchers to implement their trust scheme as detailed in Section 6.5.2. All the required information is passed as arguments to these methods with a view to providing all the necessary information for any proposed trust scheme. These arguments are passed as pointers to minimize memory and processing overhead.

All NS-3 modules are located in the src directory, where the directory's name is the name of the module. TrustMod is organized into six directories in addition to a wscript file, as illustrated in Fig. 6.6. The TrustHelper class source code is available in the helper directory. The model directory contains Trust, TrustNode, ReceivedRecommendations, SchemeParameters, and TrustScheme classes. The bindings directory contains files related to Python bindings, while the test directory includes required module test files. Simulation examples are provided in the examples directory, while manual and useful instructions are provided in the doc directory. Finally, as all NS-3 modules depend on core modules, these dependencies are defined in the wscript file. Moreover, all module source and header files must be defined there. The wscript file can be regarded as a Makefile.

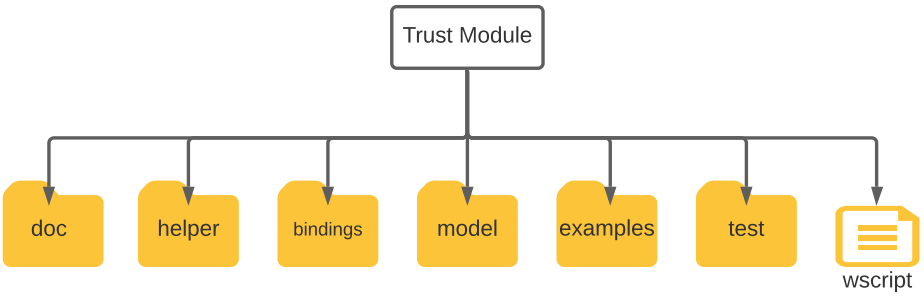


Figure 6.6: TrustMod Organization

6.6 Module Validation

In this section, we validate the results obtained by the TrustMod module. The proposed trust scheme in Chapter 3 is adopted for the validation process. Simulation parameters of LTMS have been adopted to validate our trust management module by comparing the results obtained using TrustMod module with those obtained by analyzing trace files. Researchers usually evaluate the trust relationship offline after running the simulation scripts to simulate their network scenarios. At the end of the simulation, the output will be a trace file that shows all the events and logs during the simulation. By analysing the trace file, they can evaluate the trustworthiness of the network's nodes. However, in TrustMod, everything is evaluated online during the simulation process. Therefore, the evaluated trust data could be used as input to other processes, such as routing protocols. Moreover, the researchers do not need to process the trace files to obtain any further results. Therefore, in order to validate TrustMod results, we run the same simulation scripts and compare the results obtained automatically by TrustMod by those obtained by analysing the trace files after the simulation.

The traffic is generated using the parameterized exponential density function shown in Eq. 6.1, which makes the number of packets different each time.

$$p(x;b) = \begin{cases} \mu e^{-\mu x} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad x \in [0, b] \tag{6.1}$$

where μ is the traffic rate parameter, and b is the bound parameter which can be used to make the generated values bounded over the interval $[0, b]$ as the exponential distribution is theoretically unbounded.

Moreover, we adopted a randomized attack model, where malicious nodes decide to forward or drop each received packet randomly. Different factors impact the dynamic evaluation of TrustMod, such as scheduled events processing and observation recording. Therefore, in order to validate the TrustMod module, the simulation scenarios will be run multiple times, and then the results will be averaged out. We run the simulation 30 times for each simulation parameters setting, followed by a statistic test. It is worth mentioning here that when the sample size is 30, the sampling distribution approximates the Gaussian distribution [201]. The two-sample Kolmogorov-Smirnov (K-S) test, a nonparametric hypothesis test, is applied to ensure that the two distributions obtained from the results using TrustMod module and trace file analysis are the same [222]. The

K-S test rejects the null hypothesis at the 5% significance level. The malicious activities are integrated into the RouteInput method of the AODV protocol. The indirect trust evaluation has been manually stopped before the simulations in order to neutralize the recommendation exchanging messages on the results. The trust scheme's robustness against on-off attacks is evaluated using the on-off Attack Detection Metric proposed in Chapter 3. The validation process is carried out for the following scenarios.

6.6.1 Variable Traffic Rates

In this experiment, we validate our proposed trust module for variable traffic rates ranging from low to high. The traffic rate has doubled each time, starting at $\mu = 5$. The obtained results using TrustMod and the offline analysis are averaged out and reported with one standard deviation in Fig. 6.7a-6.7d, which show the detection performance for variable on-off attack cycles. The figures show an identical behavior without any noticeable difference between using TrustMod and without using it. This ensures the design and implementation accuracy of TrustMod. The detection performance ranges between around 74% and 97%. Table 6.2 shows the two-sample K-S test results for each parameters setting, including the test decision for the 30 runs of simulation, P-value, and test statistic. The minimum, maximum, mean, and standard deviation values have been reported. It is worth mentioning that due to the high randomness level of the simulation settings, the results show some low minimum P-values. Therefore, the mean and standard deviation of the P-value have been reported to show that even though some low minimum values are shown in the table, the averaged P-value is still high, proving that the obtained trust values by the two methods are almost identical. The test statistic represents the maximum absolute difference between the two cumulative distribution functions as shown in Eq. 6.2.

$$D^* = \max_x(|\hat{F}_1(x) - \hat{F}_2(x)|) \tag{6.2}$$

where D^* represents the test statistic, $\hat{F}_1(x)$ is the cumulative distribution function for the trust values obtained using TrustMod modules, and $\hat{F}_2(x)$ is the cumulative distribution function for the trust values obtained using trace file analysis.

The results show that the average test statistic ranges between 0.029 and 0.064, which indicates that both cumulative distribution functions are very similar.

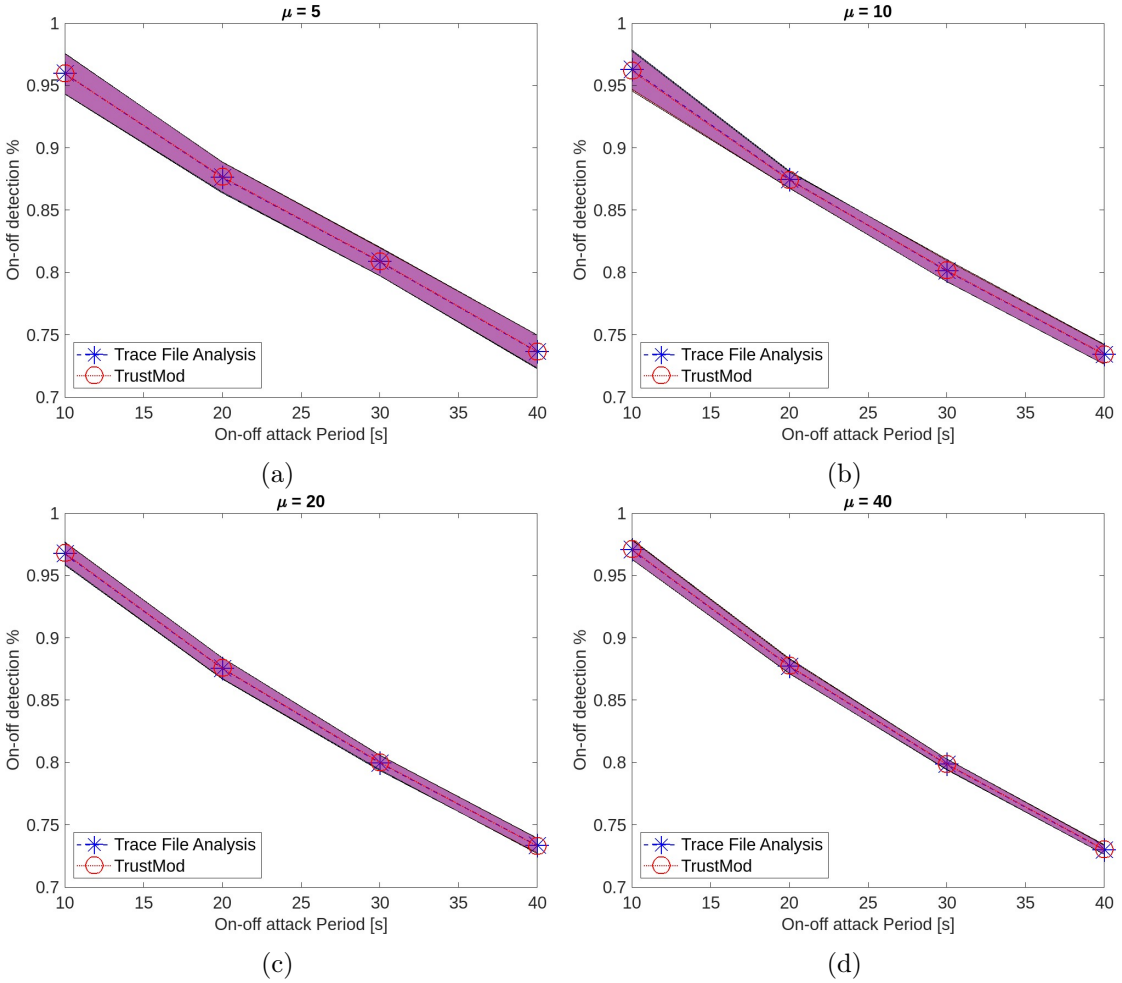


Figure 6.7: The detection performance for variable traffic rates

6.6.2 Variable Drop Rates

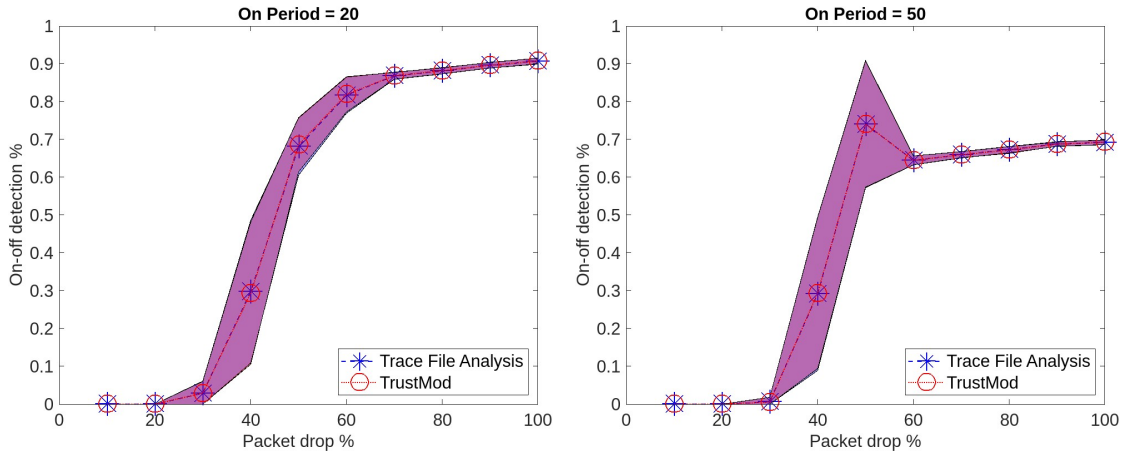
In this experiment, we use on-off attacks with variable drop rates to evaluate the detection performance using TrustMod and without using it. The drop rate varies between 10% and 100% during the on period. This experiment has been conducted for two different on periods, 20 and 50 time units, using the same traffic rate $\mu = 10$. Fig. 6.8a and Fig. 6.8b show the detection performance with one standard deviation for both methods. It is obvious that the reported detection performance for TrustMod and trace file analysis are identical, which means that the two implementations follow the same behavior in attack detection.

On the other hand, a further investigation is done using the two-sample K-S test to ensure

Table 6.2: K-S test for variable traffic rates

Simulation parameters	K-S test decision	P-value				Test statistic			
		Min	Max	Mean	Std	Min	Max	Mean	Std
$\mu = 5$, $period = 10$	30/30	0.524	1.0	0.952	0.124	0.005	0.080	0.032	0.020
$\mu = 5$, $period = 20$	30/30	0.601	1.0	0.973	0.078	0.005	0.076	0.031	0.018
$\mu = 5$, $period = 30$	30/30	0.687	1.0	0.966	0.082	0.005	0.071	0.032	0.018
$\mu = 5$, $period = 40$	30/30	0.847	1.0	0.986	0.039	0.015	0.061	0.032	0.013
$\mu = 10$, $period = 10$	30/30	0.524	1.0	0.918	0.135	0.015	0.08	0.042	0.020
$\mu = 10$, $period = 20$	30/30	0.776	1.0	0.985	0.049	0.01	0.065	0.029	0.014
$\mu = 10$, $period = 30$	30/30	0.607	1.0	0.980	0.081	0.010	0.075	0.029	0.013
$\mu = 10$, $period = 40$	30/30	0.693	1.0	0.977	0.062	0.015	0.070	0.034	0.014
$\mu = 20$, $period = 10$	30/30	0.374	1.0	0.963	0.117	0.010	0.090	0.036	0.016
$\mu = 20$, $period = 20$	30/30	0.445	1.0	0.946	0.137	0.010	0.085	0.038	0.017
$\mu = 20$, $period = 30$	30/30	0.693	1.0	0.967	0.073	0.010	0.070	0.038	0.014
$\mu = 20$, $period = 40$	30/30	0.310	1.0	0.924	0.154	0.015	0.095	0.043	0.019
$\mu = 40$, $period = 10$	30/30	0.205	1.0	0.748	0.254	0.030	0.106	0.064	0.020
$\mu = 40$, $period = 20$	30/30	0.312	1.0	0.798	0.193	0.025	0.095	0.059	0.018
$\mu = 40$, $period = 30$	30/30	0.205	1.0	0.793	0.229	0.030	0.106	0.061	0.018
$\mu = 40$, $period = 40$	30/30	0.203	1.0	0.776	0.246	0.030	0.106	0.061	0.020

that the two obtained trust series come from the same distribution. Thirty simulation runs have been carried out for each simulation parameters setting. This results in a total of 600 simulation runs. Table 6.3 shows the two-sample K-S test for the variable drop rates experiment. The null hypothesis has been accepted in 596 out of 600 simulation runs, which makes up 99.3% of all simulation runs. The four rejected cases have been intensely investigated to find out the reason for the rejection. The comparison of the two trust series shows very close trust values over time, following the same trend. Moreover, the detection performance for both simulations was identical, which proves that both trust series are almost identical. Furthermore, the averaged P-value of the parameter settings that reported one rejection shows a high probability of accepting the null hypothesis up to 92.5%. Therefore, this trivial difference in the trust values obtained from TrustMod module and trace file analysis can be attributed to the high simulation randomness and the system design. All generated traffics and packet dropping are randomly achieved during the simulation, as discussed earlier. Moreover, we obtain an efficient method to store and process the observations during the simulations. All TN objects are created just after the watchdog unit reports new observations during a specific time unit. Afterward, all required processing is scheduled to be run at the end of the current time unit. This efficient mechanism may lead to having few observations reported in the next time unit as all processing is running online, in contrast to trace file analysis where the processing is running offline after the simulation.



(a) (b)

Figure 6.8: The detection performance for variable drop rates

Table 6.3: K-S test for variable drop rates

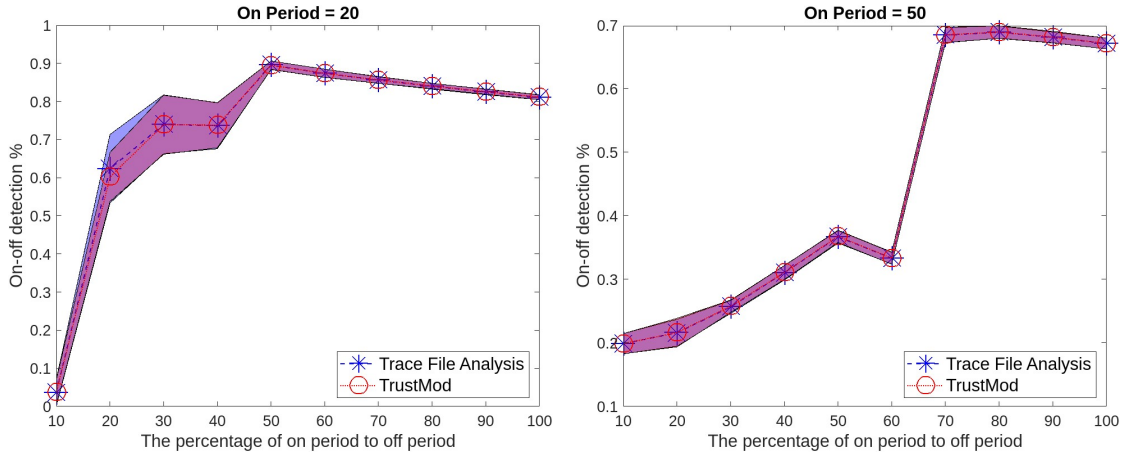
Simulation parameters	K-S test decision	P-value				Test statistic			
		Min	Max	Mean	Std	Min	Max	Mean	Std
$\mu = 10$, $period = 20$, $drop = 10\%$	30/30	0.102	1.0	0.590	0.305	0.035	0.121	0.077	0.023
$\mu = 10$, $period = 20$, $drop = 20\%$	30/30	0.445	1.0	0.906	0.138	0.035	0.085	0.051	0.013
$\mu = 10$, $period = 20$, $drop = 30\%$	29/30	0.002	1.0	0.925	0.223	0.010	0.186	0.044	0.032
$\mu = 10$, $period = 20$, $drop = 40\%$	29/30	0.007	1.0	0.868	0.289	0.015	0.166	0.046	0.034
$\mu = 10$, $period = 20$, $drop = 50\%$	30/30	0.061	1.0	0.938	0.186	0.015	0.131	0.037	0.023
$\mu = 10$, $period = 20$, $drop = 60\%$	30/30	0.165	1.0	0.940	0.200	0.015	0.11	0.036	0.022
$\mu = 10$, $period = 20$, $drop = 70\%$	30/30	0.445	1.0	0.949	0.116	0.010	0.085	0.033	0.021
$\mu = 10$, $period = 20$, $drop = 80\%$	30/30	0.693	1.0	0.981	0.061	0.005	0.070	0.0271	0.016
$\mu = 10$, $period = 20$, $drop = 90\%$	30/30	0.607	1.0	0.969	0.099	0.005	0.075	0.030	0.017
$\mu = 10$, $period = 20$, $drop = 100\%$	30/30	0.310	1.0	0.943	0.150	0.005	0.095	0.033	0.021
$\mu = 10$, $period = 50$, $drop = 10\%$	30/30	0.080	1.0	0.582	0.279	0.030	0.126	0.077	0.022
$\mu = 10$, $period = 50$, $drop = 20\%$	30/30	0.445	1.0	0.884	0.155	0.025	0.085	0.052	0.015
$\mu = 10$, $period = 50$, $drop = 30\%$	30/30	0.524	1.0	0.964	0.094	0.020	0.080	0.040	0.013
$\mu = 10$, $period = 50$, $drop = 40\%$	29/30	0.000	1.0	0.744	0.330	0.025	0.307	0.067	0.053
$\mu = 10$, $period = 50$, $drop = 50\%$	29/30	0.034	1.0	0.847	0.296	0.010	0.141	0.049	0.033
$\mu = 10$, $period = 50$, $drop = 60\%$	30/30	0.524	1.0	0.953	0.107	0.010	0.080	0.040	0.015
$\mu = 10$, $period = 50$, $drop = 70\%$	30/30	0.696	1.0	0.984	0.056	0.020	0.070	0.036	0.011
$\mu = 10$, $period = 50$, $drop = 80\%$	30/30	0.849	1.0	0.976	0.040	0.015	0.061	0.038	0.013
$\mu = 10$, $period = 50$, $drop = 90\%$	30/30	0.524	1.0	0.961	0.098	0.015	0.080	0.037	0.015
$\mu = 10$, $period = 50$, $drop = 100\%$	30/30	0.776	1.0	0.973	0.056	0.009	0.065	0.037	0.014

6.6.3 Non-Identical Periods

In this experiment, we run more sophisticated on-off attacks to validate the obtained results of our TrustMod module. The ratio of the on-to-off period varies, starting from 10% to 100%, in order to make the attack harder to detect. The traffic rate μ is set to 10, and two on periods have been considered, 20 and 50. A total of 600 simulation runs for 20 different parameters setting are carried out. Fig. 6.9a and Fig. 6.9b show the

detection performance with one standard deviation for both methods, TrustMod module and trace file analysis. The two figures show an almost identical performance for the two methods in both periods.

The same nonparametric hypothesis test has been used to ensure that both trust series come from the same distribution. Table 6.4 shows the two-sample K-S test results. The test decision has accepted the null hypothesis in the 600 simulation runs, which makes up a 100% accepting rate. The averaged P-value ranges between 0.862 and 0.992, with a maximum P-value of 1 for all parameters setting. Moreover, the averaged test statistic shows very low values ranging from 0.027 to 0.050. These reported values, along with the detection performance results, prove that both trust series are almost identical, which ensures the design and implementation accuracy of TrustMod.



(a) (b)

Figure 6.9: The detection performance for different on-off ratios

6.7 Computational Overhead

In this section, we evaluate the performance overhead introduced by TrustMod module. A number of simulations are carried out to measure the computational time and the used memory by increasing the network size.

6.7.1 Simulation Scenario

The network topology plays a significant role when evaluating the performance. For instance, choosing randomized parameters, such as nodes' positions and traffic rates,

Table 6.4: K-S test for non-identical periods

Simulation parameters	K-S test decision	P-value				Test statistic			
		Min	Max	Mean	Std	Min	Max	Mean	Std
$\mu = 10$, $period = 20$, $ratio = 10\%$	30/30	0.165	1.0	0.862	0.221	0.020	0.111	0.050	0.022
$\mu = 10$, $period = 20$, $ratio = 20\%$	30/30	0.254	1.0	0.877	0.217	0.020	0.101	0.047	0.022
$\mu = 10$, $period = 20$, $ratio = 30\%$	30/30	0.374	1.0	0.936	0.161	0.015	0.090	0.040	0.019
$\mu = 10$, $period = 20$, $ratio = 40\%$	30/30	0.524	1.0	0.934	0.134	0.015	0.080	0.039	0.019
$\mu = 10$, $period = 20$, $ratio = 50\%$	30/30	0.061	1.0	0.920	0.205	0.015	0.131	0.038	0.025
$\mu = 10$, $period = 20$, $ratio = 60\%$	30/30	0.524	1.0	0.957	0.109	0.015	0.080	0.035	0.017
$\mu = 10$, $period = 20$, $ratio = 70\%$	30/30	0.852	1.0	0.992	0.028	0.010	0.060	0.027	0.012
$\mu = 10$, $period = 20$, $ratio = 80\%$	30/30	0.852	1.0	0.992	0.028	0.010	0.060	0.028	0.013
$\mu = 10$, $period = 20$, $ratio = 90\%$	30/30	0.607	1.0	0.965	0.096	0.010	0.075	0.034	0.017
$\mu = 10$, $period = 20$, $ratio = 100\%$	30/30	0.776	1.0	0.986	0.043	0.015	0.065	0.031	0.014
$\mu = 10$, $period = 50$, $ratio = 10\%$	30/30	0.445	1.0	0.890	0.149	0.020	0.085	0.050	0.016
$\mu = 10$, $period = 50$, $ratio = 20\%$	30/30	0.205	1.0	0.909	0.204	0.024	0.106	0.046	0.020
$\mu = 10$, $period = 50$, $ratio = 30\%$	30/30	0.445	1.0	0.921	0.153	0.015	0.085	0.045	0.017
$\mu = 10$, $period = 50$, $ratio = 40\%$	30/30	0.693	1.0	0.935	0.096	0.030	0.070	0.046	0.013
$\mu = 10$, $period = 50$, $ratio = 50\%$	30/30	0.310	1.0	0.886	0.191	0.020	0.095	0.049	0.019
$\mu = 10$, $period = 50$, $ratio = 60\%$	30/30	0.131	1.0	0.868	0.246	0.020	0.116	0.050	0.023
$\mu = 10$, $period = 50$, $ratio = 70\%$	30/30	0.374	1.0	0.932	0.147	0.020	0.090	0.043	0.017
$\mu = 10$, $period = 50$, $ratio = 80\%$	30/30	0.607	1.0	0.961	0.092	0.012	0.075	0.037	0.016
$\mu = 10$, $period = 50$, $ratio = 90\%$	30/30	0.607	1.0	0.952	0.107	0.010	0.075	0.037	0.017
$\mu = 10$, $period = 50$, $ratio = 100\%$	30/30	0.607	1.0	0.976	0.076	0.015	0.075	0.034	0.014

affects the comparison process of simulations. Therefore, in order to make the comparison fair, the grid position allocation has been adopted, and the Constant Bit Rate (CBR) is used to generate traffic. Table 6.5 shows the used simulation parameters. The network topology consists of rows of nodes, where one of them acts as a sink, and other nodes act as clients. The propagation delay is constant, and the propagation loss model depends on the distance between the transmitter and receiver with a view to minimizing their impacts on the performance measurements.

Table 6.5: TrustMod Simulation Parameters

Parameter	Value
Application	CBR
Interval	1s
Packet size	264B
Routing Protocol	AODV
Radio Range	10m
Propagation delay model	Constant speed propagation delay
Propagation loss model	Range propagation loss
Time unit	10s
Simulation Time	500s

6.7.2 Performance Results

The performance overhead introduced by TrustMod is evaluated for different network sizes. The number of nodes in the network is increased from 5 to 200 nodes. The processing time to run the simulation scenario, in addition to the memory usage, including physical and virtual memory, are evaluated before and after enabling the TrustMod module. The experiments are carried out on an Intel Core i5-8500T processor at 2.1GHz and 4GB RAM. For each parameters setting, we run the simulations ten times. This took around five days of simulation running on the aforementioned computer configuration. The results are then averaged out and reported in the following figures considering one standard deviation. Fig. 6.10 shows the processing time of using the TrustMod module for different network sizes. It shows a minimal increase in processing time for large network sizes along with unnoticeable overhead for small and medium networks, ensuring the lightweight design of TrustMod.

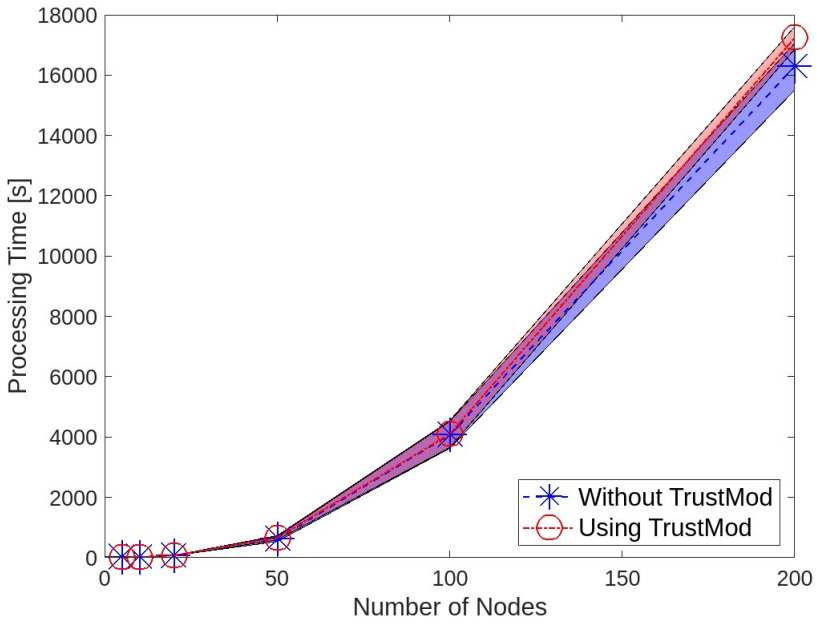
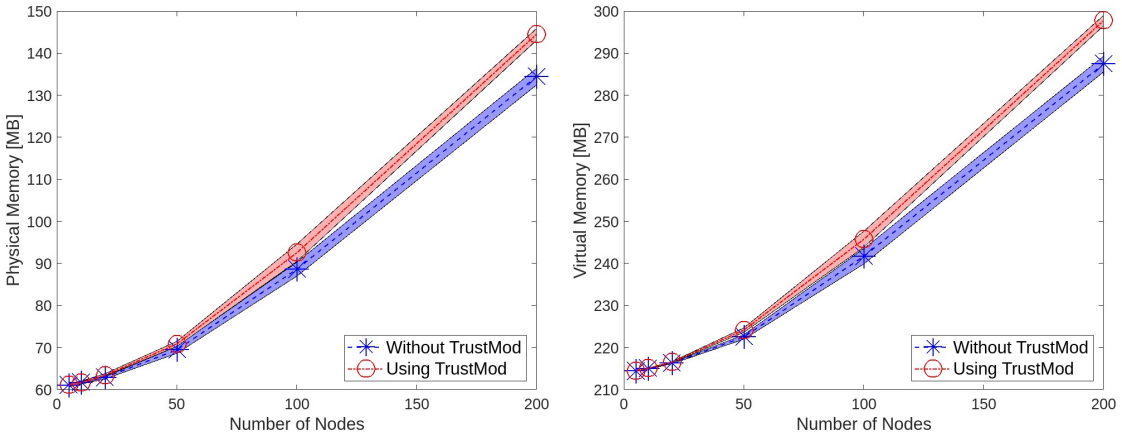


Figure 6.10: The average processing time

The second important metric is memory consumption. The average virtual and physical memory consumption are evaluated for a different number of nodes. Fig. 6.11a and Fig. 6.11b show the average consumed physical and virtual memory, respectively. The memory consumption is monitored during the simulation by scheduling five events with a lag that represents 20% of the total simulation time. TrustMod module provides a memory conservative trust management testbed. The memory overhead is unobtrusive

for small and medium network sizes. Moreover, it shows an increase of around 7% and 4% for physical and virtual memory for large networks, respectively. This slight increase is expected by increasing the network size because each node maintains the whole observations and trust evaluations for all nodes that were communicated with during the simulation.



(a) (b)

Figure 6.11: Memory Consumption

6.8 Conclusion

Trust management schemes provide a powerful tool to reinforce the overall security. Our proposed trust management module, TrustMod, provides a lightweight and accurate testbed to evaluate the effectiveness of the proposed trust schemes. TrustMod is designed to be an NS-3 module and can be easily integrated and used with any network protocol. Validation results prove the high accuracy of the trust evaluations for a wide range of simulation parameters setting with minimum resource overhead.

Chapter 7

Conclusion

This chapter summarizes the key outcomes of the preceding chapters and discusses the limitations and the future directions of this body of work.

7.1 Summary

This work contributes to the state-of-the-art advancement in developing an efficient, lightweight, and reliable routing protocol for WMSN. The methods proposed in this thesis address several challenges to ensure delivery reliability in a resource-conservative way.

Firstly, ensuring reliable data delivery requires integrating a security countermeasure to deal with dropping attacks because traditional routing inputs, such as hop counts, cannot be used to make trustworthy routing decisions that can avoid malicious paths between the source and the destination. Although trust management is a promising candidate, there are three main challenges. Any proposed TMS must be lightweight to fit the tough resource constraints. Moreover, it should be effective and attack resistant. In Chapter 3, we proposed an effective, lightweight and attack-resistant trust management scheme for WMSN. The trust relationship between two SNs is modeled using a beta probability distribution. Beta distribution-based trust management schemes suffer from high detection time due to the fact that the evaluated trust value represents the long-term expected value. As adversaries may take advantage of this drawback, a novel updating mechanism is proposed to enable fast detection. The proposed updating mechanism adopts the asymmetry trust principle to automatically penalize continuous bad behavior

over time. The results show that the proposed updating methods are more sensitive to bad behavior change and enhance the detection speed. Moreover, it immunizes the TMS from being manipulated by on-off attacks.

Two trust evaluation methods have been proposed for different kinds of SNs to accommodate their resource-constrained nature. The first is based only on the proposed updating algorithm and is proposed mainly for in-body SNs. However, it could be easily used by other researchers to enhance the robustness of their proposed trust management schemes. Moreover, in order to address the security concerns of complicated on-off attacks launched by smart adversaries, a further level of protection is proposed in the second method for on- and off-body SNs. This on-off protection module is designed to detect repeated behavior and penalize malicious trustees increasingly when they continue their malicious activities. Thus, bad behaviour is becoming more difficult to be forgotten.

As discussed in Chapter 4, lightweight and reliable data delivery are the most crucial requirements for designing a routing protocol for WMSN. Unfortunately, these two essential requirements have been overlooked by the existing proposals in the literature. The RL-based routing proposals are regarded as a resource-consuming process. To the best of our knowledge, all routing proposals use the same traditional RL model, which is also used in all RL applications. To bridge this gap, in Chapter 4, a lightweight RL model has been proposed to deal with the resource constraints of SNs. However, it could be adopted for other RL applications. The proposed RL model assumes a stationary environment for a short time unit. This assumption allows the learning agent to repeat the same action during a time window and afterward receive one accumulated reward from the environment to update its estimators. This new approach in designing RL-based applications reduces the number of estimators updating and hence can cut the computational overhead significantly.

The synchronous updating method used with the proposed RL model may delay the convergence as the learning agent takes longer time to tackle wrong routing decisions. This shortcoming has been addressed by an asynchronous updating method. The asynchronous updating mechanism can immediately update the estimators once a wrong routing decision is made and allow the algorithm to converge faster. Moreover, it helps the routing decision engine to recover and reflect any environment change fast.

To address the reliable data delivery requirement, the proposed TM scheme in Chapter 3 has been integrated with the reward function to protect the routing protocol from dropping attacks. Moreover, the reward function is defined as a punishment function

to induce the algorithm to always choose the most reliable shortest cost path to the destination. This design approach is adopted to reduce the RF activities, which are considered the main contributor to energy consumption [194].

The routing proposal in Chapter 4 has two shortcomings. The first is the inborn positive bias of the Q-learning-based routing protocols, while the second is the need to integrate an energy model to enhance the overall network lifetime and load balance the energy consumption across the network.

To address the positive bias of our routing proposal in Chapter 4, double Q-learning has been used to tackle the overestimation problem caused by using a single estimator to approximate the maximum action values using the maximum estimated action values. Two estimators have been used to evaluate the state-action value to protect the routing engine from this overestimation problem.

On the other hand, an energy model is integrated with the reward function to ensure energy-aware routing decisions. The proposed energy model is developed to only consider local information with a view to addressing two main issues. First, broadcasting energy updates increase the RF activities significantly, thus increasing the energy consumption incurred by the SNs. Second, considering second-hand information without pre-processing and filtering out false information may influence the routing decisions negatively. Therefore, the proposed energy model has only considered two local parameters to avoid routing paths through overloaded or energy-depleted SNs.

During this research project, we realized that there is a need for a realistic testbed to evaluate the proposed TMS schemes. Researchers are still using numerical analysis to show the robustness of their proposal despite its limitations. This applies to all kinds of networks, not just WMSN in particular. Therefore, in Chapter 6, a generic trust management testbed has been proposed as a module to be aggregated to the conceptual entity "node" in the NS-3 simulator. The proposed module can be used like any other available protocol or module in NS-3. This allows the researchers to instantiate the trust module and configure its attributes easily. The generic design of TrustMod gives more flexibility to simulate different kinds of trust management schemes. Moreover, the proposed trust module has a resource-efficient design to minimize the computational overhead. It stores the simulation outputs and statistics efficiently. Furthermore, it outputs them in a readable format to help the researcher to extract the required information at the end of the simulation readily.

7.2 Objectives Revisited

In this section, We revisit the research objectives set out in Section 1.3 and summarise how each of which has been addressed in this doctoral research.

- **Developing an effective, lightweight and attack-resistant trust management scheme**. This objective has been addressed in Chapter 3. The results showed that the proposed updating mechanism for probability-based TMS is able to evaluate the trust relationship accurately and defeat complicated on-off attacks with minimum resource footprint.
- **Developing a novel RL model for routing applications**. This objective has been addressed in Chapter 4. The results showed that the proposed RL model along with the synchronous and asynchronous methods are able to cut the computational overhead significantly and reflect any behaviour change swiftly.
- **Developing an energy-efficient reliable routing protocol for WMSN**. This objective has been addressed in Chapter 5, incorporating the trust scheme proposed in Chapter 3 and the RL model proposed in Chapter 4. The results showed that DQR is able to ensure high delivery ratio even in hostile environment with minimum energy and computation overhead.
- **Developing a trust management testbed**. This objective has been addressed in Chapter 6. The research outcome is a realistic, generic, lightweight, and scalable trust module for NS-3 simulator.

7.3 Future Directions

WMSN is relatively a young field with numerous favorable applications. As a result, it has attracted a great deal of academic research and industrial attention. Throughout this work, we have identified several potential improvements and future directions.

7.3.1 Multinomial Trust Management

The proposed trust management system in Chapter 3 is based on the beta probability density function to represent events in the binary space where the observations are regarded as a sequence of trials with binomial outcomes. Our novel method to update the beta probability distribution levels, in addition to our method to protect the trust evaluation engine from on-off attacks, could be generalized to represent events with multinomial

outcomes. Dirichlet probability distribution could be used to model the trust relationship for some kinds of networks and applications. For instance, in Delay Tolerant Network (DTN), the packet forwarding service could be modeled to have three outcomes instead of two. For instance, forwarded on time, delayed, and dropped.

7.3.2 Evaluate the proposed RL for other applications

The proposed RL model in Chapter 4 could be regarded as a generalization of the traditional RL model. Therefore, it is worth evaluating the proposed method for other RL applications, especially those with limited resources. Although adopting the proposed RL model incurs more design burdens to build updating methods, it can reduce the computational overhead significantly.

7.3.3 Develop a more efficient exploration strategy

In RL, the learning agent needs to discover the environment to make informed decisions. More exploration may delay or even prevent the learning agent from converging to the global optimum, while insufficient exploration may lead to converging to a local optimum. Moreover, randomly taken actions given a state affects not only the immediate reward but could affect all future actions and rewards. Therefore, the exploration-exploitation tradeoff is a critical part of RL-based routing protocol. In Chapter 4, two methods to balance exploration-exploitation have been evaluated with acceptable convergence speed. However, there is still a need for a robust exploration strategy that can deal with a stochastic environment. Although the exploration-exploitation tradeoff is a fundamental dilemma that is still unresolved [223], a well-tailored exploration strategy for routing applications may enhance the overall performance.

7.3.4 Hyperparameters tuning

Hyperparameters optimization is an essential part of achieving faster convergence and better overall performance. In Chapter 4, an attempt to tune the learning rate and the discount factor of the RL model was made. The process has been done through grid search by defining a search space as a grid of parameters' values and evaluating the network performance for each position in this space. Looking at the problem holistically to find the optimized parameters for the whole network resulted in enhancing the overall performance. However, as each learning agent has a partial view of the environment with a different number of adjacent nodes and distance from the sink, the optimized hyperparameters values could be variable and depending on the sub-environment of each

learning agent. This hypothesis is worth further investigation to determine if there is a relation between the hyperparameters and the other environment parameters.

7.3.5 TrustMod further validation experiments

In Chapter 6, we proposed a generic testbed to simulate the trust relationship for different networks and protocols. Generally speaking, there are two primary components for trust evaluation in the literature, direct and indirect trust. Indirect trust evaluation is a resource-consuming process and may involve security concerns. Therefore, we designed our testbed to consist of three components, direct trust, indirect trust, and uncertainty. The latter is proposed to control when the trustor needs to request recommendations from others. Although the functionality of all the components has been validated through several tests, the direct trust component is the only one that has been validated using a trust management scheme. Therefore, further investigation for uncertainty and indirect trust components following the same approach in Chapter 6 will be helpful to ensure the accuracy of the obtained results.

7.3.6 Evaluation using real hardware

The merit of the proposed methods in this doctoral research have been evaluated using simulation. However, the implementation of the proposed methods using real world SNs is essential to evaluate the computational overhead in runtime, such as CPU load, memory usage, and I/O operations.

Bibliography

[1] IEEE. IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks. IEEE Std 802156-2012. 2012 Feb:1-271.

[2] Latré B, Braem B, Moerman I, Blondia C, Demeester P. A survey on wireless body area networks. Wireless networks. 2011;17(1):1-18.

[3] Population Division. World Population Prospects. United Nations; 2017. Last accessed 13 May 2019. Available from: https://esa.un.org/unpd/wpp/Publications/Files/WPP2017_KeyFindings.pdf.

[4] Hajar MS, Al-Kadri MO, Kalutarage H. LTMS: A lightweight trust management system for wireless medical sensor networks. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE; 2020. p. 1783-90.

[5] Hajar MS, Al-Kadri MO, Kalutarage HK. A survey on wireless body area networks: Architecture, security challenges and research opportunities. Computers & Security. 2021;104:102211.

[6] IEEE. IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802154-2015 (Revision of IEEE Std 802154-2011). 2016 April:1-709.

[7] Yazdandoost K, Sayrafian-Pour K. TG6 channel model ID: 802.15-08-0780-12-0006. IEEE submission, Nov. 2010.

[8] Smith DB, Miniutti D, Lamahewa TA, Hanlen LW. Propagation models for body-area networks: A survey and new outlook. IEEE Antennas and Propagation Magazine. 2013;55(5):97-117.

[9] Office of National Statistics. National Population Projections: 2016-based statistical bulletin. Office of National Statistics; 2016. Accessed: 14-05-2019. Available from: https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationprojections/bulletins/nationalpopulationprojections/2016basedstatisticalbulletin/pdf.

[10] World Health Organization. Global status report. World Health Organization; 2010. Accessed: 14-05-2019. Available from: https://www.who.int/nmh/publications/ncd_report_full_en.pdf.

[11] Yoo HJ. Wireless body area network and its healthcare applications. In: 2013 Asia-Pacific Microwave Conference Proceedings (APMC). IEEE; 2013. p. 89-91.

[12] Kwak KS, Ullah S, Ullah N. An overview of IEEE 802.15.6 standard. In: 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010). IEEE; 2010. p. 1-6.

[13] Barakah DM, Ammad-uddin M. A survey of challenges and applications of wireless body area network (WBAN) and role of a virtual doctor server in existing architecture. In: 2012 Third International Conference on Intelligent Systems Modelling and Simulation. IEEE; 2012. p. 214-9.

[14] Bharathi KS, Venkateswari R. Security Challenges and Solutions for Wireless Body Area Networks. In: Computing, Communication and Signal Processing. Springer; 2019. p. 275-83.

[15] Paul PC, Loane J, Regan G, McCaffery F. Analysis of Attacks and Security Requirements for Wireless Body Area Networks-A Systematic Literature Review. In: European Conference on Software Process Improvement. Springer; 2019. p. 439-52.

[16] Al-Janabi S, Al-Shourbaji I, Shojafar M, Shamshirband S. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. Egyptian Informatics Journal. 2017;18(2):113-22.

[17] Challa S, Wazid M, Das AK, Khan MK. Authentication protocols for implantable medical devices: taxonomy, analysis and future directions. IEEE Consumer Electronics Magazine. 2017;7(1):57-65.

[18] Kompara M, Hölbl M. Survey on security in intra-body area network communication. Ad Hoc Networks. 2018;70:23-43.

[19] Khernane N, Potop-Butucaru M, Chaudet C. BANZKP: A secure authentication scheme using zero knowledge proof for WBANs. In: 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE; 2016. p. 307-15.

[20] Toorani M. Security analysis of the IEEE 802.15. 6 standard. International Journal of Communication Systems. 2016;29(17):2471-89.

[21] Toorani M. On vulnerabilities of the security association in the IEEE 802.15. 6 standard. In: International conference on financial cryptography and data security. Springer; 2015. p. 245-60.

[22] Qu Y, Zheng G, Ma H, Wang X, Ji B, Wu H. A survey of routing protocols in WBAN for healthcare applications. Sensors. 2019;19(7):1638.

[23] Cavallari R, Martelli F, Rosini R, Buratti C, Verdone R. A survey on wireless body area networks: Technologies and design challenges. IEEE Communications Surveys & Tutorials. 2014;16(3):1635-57.

[24] Karmakar K, Biswas S, Neogy S. MHRP: A novel mobility handling routing protocol in Wireless Body Area Network. In: 2017 international conference on wireless communications, signal processing and networking (WiSPNET). IEEE; 2017. p. 1939-45.

[25] Quwaider M, Biswas S. DTN routing in body sensor networks with dynamic postural partitioning. Ad Hoc Networks. 2010;8(8):824-41.

[26] Samanta A, Misra S. Energy-Efficient and Distributed Network Management Cost Minimization in Opportunistic Wireless Body Area Networks. IEEE Transactions on Mobile Computing. 2018;17(2):376-89.

[27] Tang Q, Tummala N, Gupta SK, Schwiebert L. TARA: thermal-aware routing algorithm for implanted sensor networks. In: International conference on distributed computing in sensor systems. Springer; 2005. p. 206-17.

[28] Ahmad A, Javaid N, Qasim U, Ishfaq M, Khan ZA, Alghamdi TA. RE-ATTEMPT: a new energy-efficient routing protocol for wireless body area sensor networks. International Journal of Distributed Sensor Networks. 2014;10(4):464010.

[29] Javaid N, Abbas Z, Fareed M, Khan ZA, Alrajeh N. M-ATTEMPT: A new energy-efficient routing protocol for wireless body area sensor networks. Procedia Computer Science. 2013;19:224-31.

[30] Rafatkhah O, Lighvan MZ. M2E2: A novel multi-hop routing protocol for wireless body sensor networks. Int J Comput Netw Commun Secur. 2014;2(8):260-7.

[31] Monowar MM, Mehedi Hassan M, Bajaber F, Hamid MA, Alamri A. Thermal-aware multiconstrained intrabody QoS routing for wireless body area networks. International Journal of Distributed Sensor Networks. 2014;10(3):676312.

[32] Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications. 2002;1(4):660-70.

[33] Al-Shalabi M, Anbar M, Wan TC, Khasawneh A. Variants of the low-energy adaptive clustering hierarchy protocol: Survey, issues and challenges. Electronics. 2018;7(8):136.

[34] Abidi B, Jilbab A, Mohamed EH. An energy efficiency routing protocol for wireless body area networks. Journal of medical engineering & technology. 2018;42(4):290-7.

[35] Peng Y, Zhang S. A Power Optimization Routing Algorithm for Wireless Body Area Network. Electron Sci Technol. 2018;31:38-41.

[36] Yun WK, Yoo SJ. Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks. IEEE Access. 2021;9:10737-50.

[37] Künzel G, Indrusiak LS, Pereira CE. Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing. IEEE Transactions on Industrial Informatics. 2019;16(8):5617-25.

[38] Yun WK, Yoo SJ. Q-Learning-Based Data-Aggregation-Aware Energy-Efficient Routing Protocol for Wireless Sensor Networks. IEEE Access. 2021;9:10737-50.

[39] Maivizhi R, Yogesh P. Q-learning based routing for in-network aggregation in wireless sensor networks. Wireless Networks. 2021;27(3):2231-50.

[40] Künzel G, Indrusiak LS, Pereira CE. Latency and Lifetime Enhancements in Industrial Wireless Sensor Networks: A Q-Learning Approach for Graph Routing. IEEE Transactions on Industrial Informatics. 2020;16(8):5617-25.

[41] Maivizhi R, Yogesh P. Q-learning based routing for in-network aggregation in wireless sensor networks. Wireless Networks. 2021;27(3):2231-50.

[42] Liu G, Wang X, Li X, Hao J, Feng Z. ESRQ: An efficient secure routing method in wireless sensor networks based on Q-learning. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom). IEEE; 2018. p. 149-55.

[43] Upreti K, Kumar N, Alam MS, Verma A, Nandan M, Gupta AK. Machine learning-based Congestion Control Routing strategy for healthcare IoT enabled wireless sensor networks. In: 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE; 2021. p. 1-6.

[44] Liang X, Balasingham I, Byun SS. A reinforcement learning based routing protocol with QoS support for biomedical sensor networks. In: 2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies. IEEE; 2008. p. 1-5.

[45] Naputta Y, Usaha W. RL-based routing in biomedical mobile wireless sensor networks using trust and reputation. In: 2012 International Symposium on Wireless Communication Systems (ISWCS). IEEE; 2012. p. 521-5.

[46] Yuan F, Wu J, Zhou H, Liu L. A double Q-learning routing in delay tolerant networks. In: ICC 2019-2019 IEEE international conference on communications (ICC). IEEE; 2019. p. 1-6.

[47] Jin Z, Ma Y, Su Y, Li S, Fu X. A Q-learning-based delay-aware routing algorithm to extend the lifetime of underwater sensor networks. Sensors. 2017;17(7):1660.

[48] Jafarzadeh SZ, Moghaddam MHY. Design of energy-aware QoS routing algorithm in wireless sensor networks using reinforcement learning. In: 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE; 2014. p. 722-7.

[49] Abdel-Fattah F, Farhan KA, Al-Tarawneh FH, AlTamimi F. Security Challenges and Attacks in Dynamic Mobile Ad Hoc Networks MANETs. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). IEEE; 2019. p. 28-33.

[50] Al Ameen M, Liu J, Kwak K. Security and privacy issues in wireless sensor networks for healthcare applications. Journal of medical systems. 2012;36(1):93-101.

[51] Masdari M, Ahmadzadeh S. Comprehensive analysis of the authentication methods in wireless body area networks. Security and Communication Networks. 2016;9(17):4777-803.

[52] Shen J, Chang S, Shen J, Liu Q, Sun X. A lightweight multi-layer authentication protocol for wireless body area networks. Future Generation Computer Systems. 2018;78:956-63.

[53] Liu R, Wang Y. A new Sybil attack detection for wireless body sensor network. In: 2014 Tenth International Conference on Computational Intelligence and Security. IEEE; 2014. p. 367-70.

[54] Vadlamani S, Eksioglu B, Medal H, Nandi A. Jamming attacks on wireless networks: A taxonomic survey. International Journal of Production Economics. 2016;172:76-94.

[55] Jo M, Han L, Tan ND, In HP. A survey: energy exhausting attacks in MAC protocols in WBANs. Telecommunication Systems. 2015;58(2):153-64.

[56] Segovia M, Grampín E, Baliosian J. Analysis of the applicability of wireless sensor networks attacks to body area networks. In: Proceedings of the 8th International Conference on Body Area Networks; 2013. p. 509-12.

[57] Osanaiye OA, Alfa AS, Hancke GP. Denial of service defence for resource availability in wireless sensor networks. IEEE Access. 2018;6:6975-7004.

[58] Diaz A, Sanchez P. Simulation of attacks for security in wireless sensor network. Sensors. 2016;16(11):1932.

[59] Rughiniş R, Gheorghe L. Storm Control Mechanism in Wireless Sensor Networks. In: 9th RoEduNet IEEE International Conference; 2010. p. 430-5.

[60] Ndoye E, Jacquet F, Misson M, Niang I. Evaluation of rts/cts with unslotted csma/ca algorithm in linear sensor networks. NICST 2013. 2013.

[61] Barbi M, Sayrafian K, Alasti M. Using RTS/CTS to enhance the performance of IEEE 802.15.6 CSMA/CA. In: 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE; 2016. p. 1-5.

[62] Shakhov VV. Protecting wireless sensor networks from energy exhausting attacks. In: International Conference on Computational Science and Its Applications. Springer; 2013. p. 184-93.

[63] Javadi SS, Razzaque M. Security and privacy in wireless body area networks for health care applications. In: Wireless networks and security. Springer; 2013. p. 165-87.

[64] Paul S, Chakraborty A, Banerjee JS. A fuzzy AHP-Based relay node selection protocol for wireless body area networks (WBAN). In: 2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix). IEEE; 2017. p. 1-6.

[65] Niksaz P, Branch M. Wireless body area networks: attacks and countermeasures. International Journal of scientific and engineering research. 2015;6(19):565-8.

[66] Hajar MS, Al-Kadri MO, Kalutarage H. ETAREE: An Effective Trend-Aware Reputation Evaluation Engine for Wireless Medical Sensor Networks. In: 2020 IEEE Conference on Communications and Network Security (CNS). IEEE; 2020. p. 1-9.

[67] Labraoui N, Gueroui M, Sekhri L. On-off attacks mitigation against trust systems in wireless sensor networks. In: IFIP International Conference on Computer Science and its Applications. Springer; 2015. p. 406-15.

[68] Polai M, Mohanty S, Sahoo SS. A Lightweight Mutual Authentication Protocol for Wireless Body Area Network. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE; 2019. p. 760-5.

[69] Shen J, Gui Z, Ji S, Shen J, Tan H, Tang Y. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. Journal of Network and Computer Applications. 2018;106:117-23.

[70] Xiong H, Qin Z. Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks. IEEE transactions on information forensics and security. 2015;10(7):1442-55.

[71] Li X, Ibrahim MH, Kumari S, Sangaiah AK, Gupta V, Choo KKR. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Computer Networks. 2017;129:429-43.

[72] He D, Zeadally S. Authentication protocol for an ambient assisted living system. IEEE Communications Magazine. 2015;53(1):71-7.

[73] Li X, Peng J, Kumari S, Wu F, Karuppiah M, Choo KKR. An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. Computers & Electrical Engineering. 2017;61:238-49.

[74] Omala AA, Kibiwott KP, Li F. An efficient remote authentication scheme for wireless body area network. Journal of medical systems. 2017;41(2):25.

[75] Xiong H. Cost-effective scalable and anonymous certificateless remote authentication protocol. IEEE Transactions on Information Forensics and Security. 2014;9(12):2327-39.

[76] Zhang Z, Wang H, Vasilakos AV, Fang H. ECG-cryptography and authentication in body area networks. IEEE Transactions on Information Technology in Biomedicine. 2012;16(6):1070-8.

[77] Rajasekaran RT, Manjula V, Kishore V, Sridhar T, Jayakumar C. An efficient and secure key agreement scheme using physiological signals in body area networks. In: Proceedings of the

International Conference on Advances in Computing, Communications and Informatics. ACM; 2012. p. 1143-7.

[78] Hu C, Cheng X, Zhang F, Wu D, Liao X, Chen D. OPFKA: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks. In: 2013 Proceedings IEEE INFOCOM. IEEE; 2013. p. 2274-82.

[79] Sammoud A, Chalouf MA, Hamdi O, Montavont N, Bouallegue A. A new biometrics-based key establishment protocol in WBAN: energy efficiency and security robustness analysis. Computers & Security. 2020:101838.

[80] Li M, Yu S, Guttman JD, Lou W, Ren K. Secure ad hoc trust initialization and key management in wireless body area networks. ACM Transactions on Sensor Networks (TOSN). 2013;9(2):18.

[81] Cai L, Zeng K, Chen H, Mohapatra P. Good Neighbor: Ad hoc Pairing of Nearby Wireless Devices by Multiple Antennas. In: NDSS; 2011. .

[82] Shi L, Yuan J, Yu S, Li M. MASK-BAN: Movement-aided authenticated secret key extraction utilizing channel characteristics in body area networks. IEEE Internet of Things Journal. 2015;2(1):52-62.

[83] Shi L, Li M, Yu S, Yuan J. BANA: body area network authentication exploiting channel characteristics. IEEE Journal on selected Areas in Communications. 2013;31(9):1803-16.

[84] Mathur S, Miller R, Varshavsky A, Trappe W, Mandayam N. Proximate: proximity-based secure pairing using ambient wireless signals. In: Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM; 2011. p. 211-24.

[85] Kaliski B. The Mathematics of the RSA Public-Key Cryptosystem. RSA Laboratories. 2006.

[86] Wang H, Li Q. Efficient implementation of public key cryptosystems on mote sensors (short paper). In: International Conference on Information and Communications Security. Springer; 2006. p. 519-28.

[87] Liu A, Ning P. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In: Proceedings of the 7th international conference on Information processing in sensor networks. IEEE Computer Society; 2008. p. 245-56.

[88] Shamir A. Identity-based cryptosystems and signature schemes. In: Workshop on the theory and application of cryptographic techniques. Springer; 1984. p. 47-53.

[89] Cao X, Zeng X, Kou W, Hu L. Identity-based anonymous remote authentication for value-added services in mobile networks. IEEE Transactions on Vehicular Technology. 2009;58(7):3508-17.

[90] Yang JH, Chang CC. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Computers & security. 2009;28(3-4):138-43.

[91] Islam SH, Biswas G. A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Journal of Systems and Software. 2011;84(11):1892-8.

[92] Al-Riyami SS, Paterson KG. Certificateless public key cryptography. In: International conference on the theory and application of cryptology and information security. Springer; 2003. p. 452-73.

[93] Zhang L, Liu J, Sun R. An efficient and lightweight certificateless authentication protocol for wireless body area networks. In: 2013 5th International Conference on Intelligent Networking and Collaborative Systems. IEEE; 2013. p. 637-9.

[94] Kasyoka P, Kimwele M, Mbandu Angolo S. Certificateless pairing-free authentication scheme for wireless body area network in healthcare management system. Journal of Medical Engineering & Technology. 2020;44(1):12-9.

[95] Singh U, Narwal B. A Novel Authentication Scheme for Wireless Body Area Networks with Anonymity. In: Progress in Advanced Computing and Intelligent Engineering. Springer; 2021. p. 295-305.

[96] Shim KA. Comments on "Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks". IEEE Transactions on Information Forensics and Security. 2018.

[97] Khan H, Dowling B, Martin KM. Highly Efficient Privacy-Preserving Key Agreement for Wireless Body Area Networks. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE; 2018. p. 1064-9.

[98] Suh GE, Devadas S. Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference. IEEE; 2007. p. 9-14.

[99] Wang W, Shi X, Qin T. Encryption-free Authentication and Integrity Protection in Body Area Networks through Physical Unclonable Functions. Smart Health. 2018.

[100] Xie L, Wang W, Shi X, Qin T. Lightweight mutual authentication among sensors in body area networks through Physical Unclonable Functions. In: 2017 IEEE International Conference on Communications (ICC). IEEE; 2017. p. 1-6.

[101] Zhang W, Qin T, Mekonen M, Wang W. Wireless Body Area Network Identity Authentication Protocol Based on Physical Unclonable Function. In: 2018 International Conference on Sensor Networks and Signal Processing (SNSP). IEEE; 2018. p. 60-4.

[102] Tan X, Zhang J, Zhang Y, Qin Z, Ding Y, Wang X. A PUF-based and cloud-assisted lightweight authentication for multi-hop body area network. Tsinghua Science and Technology. 2020;26(1):36-47.

[103] Djenouri D, Khelladi L, Badache N. Security issues of mobile ad hoc and sensor networks. In: IEEE Communications Surveys Tutorials. vol. 7. IEEE Communications Society; 2005. p. 2-28.

[104] Mainanwal V, Gupta M, Upadhayay SK. A survey on wireless body area network: Security technology and its design methodology issue. In: 2015 international conference on innovations in information, embedded and communication systems (ICIIECS). IEEE; 2015. p. 1-5.

[105] Dworkin MJ. Sp 800-38c. recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. 2004.

[106] Padmavathi B, Kumari SR. A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution.

[107] Nadeem A, Javed MY. A performance comparison of data encryption algorithms. In: 2005 international conference on information and communication technologies. IEEE; 2005. p. 84-9.

[108] McKay K, Bassham L, Sönmez Turan M, Mouha N. Report on lightweight cryptography. National Institute of Standards and Technology; 2016.

[109] Singh S, Sharma PK, Moon SY, Park JH. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Journal of Ambient Intelligence and Humanized Computing. 2017:1-18.

[110] CMT. MICAz;. Accessed: 07-11-2019. Available from: http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf.

[111] Yang G, Zhu B, Suder V, Aagaard MD, Gong G. The simeck family of lightweight block ciphers. In: International Workshop on Cryptographic Hardware and Embedded Systems. Springer; 2015. p. 307-29.

[112] Lim CH, Korkishko T. mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors. In: International Workshop on Information Security Applications. Springer; 2005. p. 243-58.

[113] Suzaki T, Minematsu K, Morioka S, Kobayashi E. Twine: A lightweight, versatile block cipher. In: ECRYPT Workshop on Lightweight Cryptography; 2011. p. 1-24.

[114] Beaulieu R, Treatman-Clark S, Shors D, Weeks B, Smith J, Wingers L. The SIMON and SPECK lightweight block ciphers. In: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE; 2015. p. 1-6.

[115] Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences. 2015;58(12):1-15.

[116] Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knezevic M, Knudsen LR, et al. Prince–a low-latency block cipher for pervasive computing applications. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer; 2012. p. 208-25.

[117] Engels D, Saarinen MJO, Schweitzer P, Smith EM. The Hummingbird-2 lightweight authenticated encryption algorithm. In: International Workshop on Radio Frequency Identification: Security and Privacy Issues. Springer; 2011. p. 19-31.

[118] Toprak S, Akbulut A, Aydın MA, Zaim AH. LWE: An Energy-Efficient Lightweight Encryption Algorithm for Medical Sensors and IoT Devices. Electrica. 2020;20(1):71-81.

[119] Qiao K, Hu L, Sun S. Differential Security Evaluation of Simeck with Dynamic Key-guessing Techniques. IACR Cryptology ePrint Archive. 2015;2015:902.

[120] Hao Y, Bai D, Li L. A meet-in-the-middle attack on round-reduced mCrypton using the differential enumeration technique. In: International Conference on Network and System Security. Springer; 2015. p. 166-83.

[121] Çoban M, Karakoç F, Boztaş Ö. Biclique cryptanalysis of TWINE. In: International Conference on Cryptology and Network Security. Springer; 2012. p. 43-55.

[122] Chen H, Wang X. Improved linear hull attack on round-reduced Simon with dynamic key-guessing techniques. In: International Conference on Fast Software Encryption. Springer; 2016. p. 428-49.

[123] Soleimany H, Blondeau C, Yu X, Wu W, Nyberg K, Zhang H, et al. Reflection cryptanalysis of PRINCE-like ciphers. Journal of Cryptology. 2015;28(3):718-44.

[124] Canteaut A, Naya-Plasencia M, Vayssiere B. Sieve-in-the-middle: improved MITM attacks. In: Annual Cryptology Conference. Springer; 2013. p. 222-40.

[125] Chai Q, Gong G. A Cryptanalysis of HummingBird-2: The Differential Sequence Analysis. IACR Cryptology ePrint Archive. 2012;2012:233.

[126] Dinur I. Improved differential cryptanalysis of round-reduced speck. In: International Conference on Selected Areas in Cryptography. Springer; 2014. p. 147-64.

[127] Alaparthy VT, Morgera SD. A Multi-Level Intrusion Detection System for Wireless Sensor Networks Based on Immune Theory. IEEE Access. 2018;6:47364-73.

[128] Butun I, Morgera SD, Sankar R. A survey of intrusion detection systems in wireless sensor networks. IEEE communications surveys & tutorials. 2014;16(1):266-82.

[129] Huang Ya, Lee W. A cooperative intrusion detection system for ad hoc networks. In: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks. ACM; 2003. p. 135-47.

[130] Thamilarasu G. iDetect: an intelligent intrusion detection system for wireless body area networks. International Journal of Security and Networks. 2016;11(1-2):82-93.

[131] Thamilarasu G, Ma Z. Autonomous mobile agent based intrusion detection framework in wireless body area networks. In: 2015 IEEE 16th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM). IEEE; 2015. p. 1-3.

[132] Odesile A, Thamilarasu G. Distributed intrusion detection using mobile agents in wireless body area networks. In: 2017 Seventh International Conference on Emerging Security Technologies (EST). IEEE; 2017. p. 144-9.

[133] Newaz AI, Sikder AK, Babun L, Uluagac AS. Heka: A novel intrusion detection system for attacks to personal medical devices. In: 2020 IEEE Conference on Communications and Network Security (CNS). IEEE; 2020. p. 1-9.

[134] Hady AA, Ghubaish A, Salman T, Unal D, Jain R. Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study. IEEE Access. 2020;8:106576-84.

[135] Dunkels A. Cooja;. Accessed: 25-04-2019. Available from: http://www.contiki-os.org.

[136] DARPA. ns-2;. Accessed: 25-04-2019. Available from: https://www.isi.edu/nsnam/ns/.

[137] Zhang Y, Li W. MobiEmu;. Accessed: 25-04-2019. Available from: http://mobiemu.sourceforge.net/.

[138] Levis P, Lee N, Welsh M, Culler D. TOSSIM;. Accessed: 25-04-2019. Available from: http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM.

[139] Boulis T, Tselishchev Y, Pediaditakis D. Castalia Simulator;. Accessed: 23-04-2019. Available from: https://github.com/boulis/Castalia.

[140] Sundararajan T, Shanmugam A. A novel intrusion detection system for wireless body area network in health care monitoring. Journal of Computer Science. 2010;6(11):1355.

[141] Technologies SN. QualNet;. Accessed: 25-04-2019. Available from: https://www.scalable-networks.com/qualnet-network-simulation.

[142] Ishmanov F, Malik AS, Kim SW, Begalov B. Trust management system in wireless sensor networks: design considerations and research challenges. Transactions on Emerging Telecommunications Technologies. 2015;26(2):107-30.

[143] Misra S, Vaish A. Reputation-based role assignment for role-based access control in wireless sensor networks. Computer Communications. 2011;34(3):281-94.

[144] Zhan G, Shi W, Deng J. Design and implementation of TARF: A trust-aware routing framework for WSNs. IEEE Transactions on dependable and secure computing. 2012;9(2):184-97.

[145] He D, Chen C, Chan S, Bu J, Vasilakos AV. ReTrust: Attack-resistant and lightweight trust management for medical sensor networks. IEEE transactions on information technology in biomedicine. 2012;16(4):623-32.

[146] Khan T, Singh K, Abdel-Basset M, Long HV, Singh SP, Manjul M, et al. A Novel and Comprehensive Trust Estimation Clustering Based Approach for Large Scale Wireless Sensor Networks. IEEE Access. 2019;7:58221-40.

[147] Han G, Jiang J, Shu L, Niu J, Chao HC. Management and applications of trust in Wireless Sensor Networks: A survey. Journal of Computer and System Sciences. 2014;80(3):602-17.

[148] Hossein J, Mohammad R, et al. A fuzzy fully distributed trust management system in wireless sensor networks. International Journal of Electronics and Communications. 2016;9(17):1-10.

[149] Kazmi F, Khan MA, Saeed A, Saqib NA, Abbas M. Evaluation of trust management approaches in wireless sensor networks. IEEE; 2018. p. 870-5.

[150] Almogren A, Mohiuddin I, Din IU, Al Majed H, Guizani N. Ftm-iomt: Fuzzy-based trust management for preventing sybil attacks in internet of medical things. IEEE Internet of Things Journal. 2020.

[151] Zhao J, Huang J, Xiong N. An Effective Exponential-Based Trust and Reputation Evaluation System in Wireless Sensor Networks. IEEE Access. 2019;7:33859-69.

[152] Fang W, Zhu C, Chen W, Zhang W, Rodrigues JJ. BDTMS: Binomial distribution-based trust management scheme for healthcare-oriented wireless sensor network. In: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE; 2018. p. 382-7.

[153] Ganeriwal S, Srivastava MB. Reputation-based Framework for High Integrity Sensor Networks. In: Proceedings of the 2Nd ACM Workshop on Security of Ad Hoc and Sensor Networks. ACM; 2004. p. 66-77.

[154] Ganeriwal S, Balzano LK, Srivastava MB. Reputation-based framework for high integrity sensor networks. ACM Transactions on Sensor Networks (TOSN). 2008;4(3):15.

[155] Fang W, Zhang X, Shi Z, Sun Y, Shan L. Binomial-based trust management system in wireless sensor networks. Chin J Sens Actuat. 2015;28(5):703-8.

[156] Labraoui N. A reliable trust management scheme in wireless sensor networks. In: 2015 12th International Symposium on Programming and Systems (ISPS). IEEE; 2015. p. 1-6.

[157] Labraoui N, Gueroui M, Sekhri L. A risk-aware reputation-based trust management in wireless sensor networks. Wireless Personal Communications. 2016;87(3):1037-55.

[158] Hussain SA, Raza I, Mehdi MM. A cluster based energy efficient trust management mechanism for medical wireless sensor networks (MWSNs). In: 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE). IEEE; 2018. p. 433-9.

[159] Mármol FG. TRMSim-WSN;. Accessed: 2019-05-02. https://sourceforge.net/projects/trmsim-wsn/.

[160] Bangash J, Abdullah A, Anisi M, Khan A. A survey of routing protocols in wireless body sensor networks. sensors. 2014;14(1):1322-57.

[161] Johny B, Anpalagan A. Body area sensor networks: Requirements, operations, and challenges. IEEE Potentials. 2014;33(2):21-5.

[162] Cunha A, Koubaa A, Severino R, Alves M. Open-ZB: an open-source implementation of the IEEE 802.15. 4/ZigBee protocol stack on TinyOS. In: 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems. IEEE; 2007. p. 1-12.

[163] Yibo C, Hou KM, Zhou H, Shi Hl, Liu X, Diao X, et al. 6LoWPAN stacks: A survey. In: 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE; 2011. p. 1-4.

[164] Ma X, Luo W. The analysis of 6LoWPAN technology. In: 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application. vol. 1. IEEE; 2008. p. 963-6.

[165] Mulligan G. The 6LoWPAN architecture. In: Proceedings of the 4th workshop on Embedded networked sensors. ACM; 2007. p. 78-82.

[166] Kohvakka M, Kuorilehto M, Hännikäinen M, Hämäläinen TD. Performance analysis of IEEE 802.15. 4 and ZigBee for large-scale wireless sensor network applications. In: Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks. ACM; 2006. p. 48-57.

[167] Stelios Y, Papayanoulas N, Trakadas P, Maniatis S, Leligou HC, Zahariadis T. A distributed energy-aware trust management system for secure routing in wireless sensor networks. In: International Conference on Mobile Lightweight Wireless Systems. Springer; 2009. p. 85-92.

[168] Fang W, Zhang W, Yang Y, Liu Y, Chen W. A resilient trust management scheme for defending against reputation time-varying attacks based on BETA distribution. Science China Information Sciences. 2017;60(4):040305.

[169] Reddy VB, Venkataraman S, Negi A. Communication and data trust for wireless sensor networks using D–S theory. IEEE Sensors Journal. 2017;17(12):3921-9.

[170] Labraoui N, Gueroui M, Sekhri L. A risk-aware reputation-based trust management in wireless sensor networks. Wireless Personal Communications. 2016;87(3):1037-55.

[171] Yao Z, Kim D, Doh Y. PLUS: Parameterized and localized trust management scheme for sensor networks security. In: 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems. IEEE; 2006. p. 437-46.

[172] Islam MN, Yuce MR. Review of medical implant communication system (MICS) band and network. Ict Express. 2016;2(4):188-94.

[173] Alrahhal H, Jamous R, Ramadan R, Alayba AM, Yadav K. Utilising Acknowledge for the Trust in Wireless Sensor Networks. Applied Sciences. 2022;12(4):2045.

[174] Zhao J, Huang J, Xiong N. An effective exponential-based trust and reputation evaluation system in wireless sensor networks. IEEE Access. 2019;7:33859-69.

[175] Fang W, Zhu C, Chen W, Zhang W, Rodrigues JJ. BDTMS: Binomial distribution-based trust management scheme for healthcare-oriented wireless sensor network. In: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE; 2018. p. 382-7.

[176] Fang W, Zhang C, Shi Z, Zhao Q, Shan L. BTRES: Beta-based Trust and Reputation Evaluation System for wireless sensor networks. Journal of Nework and Computer Applications. 2016;59:88-94.

[177] Fang W, Zhang W, Liu Y, Yang W, Gao Z. BTDS: Bayesian-based trust decision scheme for intelligent connected vehicles in VANETs. Transactions on Emerging Telecommunications Technologies. 2020.

[178] Moe ME, Helvik BE, Knapskog SJ. Comparison of the beta and the hidden markov models of trust in dynamic environments. In: IFIP International Conference on Trust Management. Springer; 2009. p. 283-97.

[179] Ishmanov F, Kim SW, Nam SY. A robust trust establishment scheme for wireless sensor networks. Sensors. 2015;15(3):7040-61.

[180] Wei Z, Tang H, Yu FR, Wang M, Mason P. Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. IEEE Transactions on Vehicular Technology. 2014;63(9):4647-58.

[181] Josang A, Ismail R. The beta reputation system. In: Proceedings of the 15th bled electronic commerce conference; 2002. p. 2502-11.

[182] Ishmanov F, Kim SW, Nam SY. A secure trust establishment scheme for wireless sensor networks. Sensors. 2014;14(1):1877-97.

[183] Morady F. Electrophysiologic interventional procedures and surgery. In: Goldman's Cecil Medicine. Elsevier; 2012. p. 369-73.

[184] Poortinga W, Pidgeon NF. Trust, the asymmetry principle, and the role of prior beliefs. Risk Analysis: An International Journal. 2004;24(6):1475-86.

[185] Open source. NS-3 a discrete-event network simulator for internet systems;. Accessed: 01-04-2020. Available from: https://www.nsnam.org/releases/.

[186] Das S, Perkins C, Royer E. Ad hoc on demand distance vector (AODV) routing. IETF RFC3561, July. 2003.

[187] Fu H, Liu Y, Dong Z, Wu Y. A data clustering algorithm for detecting selective forwarding attack in cluster-based wireless sensor networks. Sensors. 2020;20(1):23.

[188] Khanna N, Sachdeva M. A comprehensive taxonomy of schemes to detect and mitigate blackhole attack and its variants in MANETs. Computer Science Review. 2019;32:24-44.

[189] Prathapchandran K, Janani T. A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST. Computer Networks. 2021;198:108413.

[190] Altowaijri SM. Efficient Next-Hop Selection in Multi-Hop Routing for IoT Enabled Wireless Sensor Networks. Future Internet. 2022;14(2):35.

[191] Ullah Z, Ahmed I, Khan FA, Asif M, Nawaz M, Ali T, et al. Energy-efficient harvested-aware clustering and cooperative routing protocol for WBAN (E-HARP). IEEE Access. 2019;7:100036-50.

[192] Bedi P, Das S, Goyal S, Shukla PK, Mirjalili S, Kumar M. A novel routing protocol based on grey wolf optimization and Q learning for wireless body area network. Expert Systems with Applications. 2022;210:118477.

[193] Guo W, Yan C, Lu T. Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing. International Journal of Distributed Sensor Networks. 2019;15(2).

[194] Azdad N, Elboukhari M. Wireless Body Area Networks for Healthcare: Application Trends and MAC Technologies. International Journal of Business Data Communications and Networking (IJBDCN). 2021;17(2):1-20.

[195] Li T, Zhu K, Luong NC, Niyato D, Wu Q, Zhang Y, et al. Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey. IEEE Communications Surveys & Tutorials. 2022.

[196] Zhang K, Yang Z, Liu H, Zhang T, Basar T. Fully decentralized multi-agent reinforcement learning with networked agents. In: International Conference on Machine Learning. PMLR; 2018. p. 5872-81.

[197] Kumar A, Matam R, Shukla S. Impact of packet dropping attacks on RPL. In: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC); 2016. p. 694-8.

[198] Chede SD, Kulat KD. Design Overview Of Processor Based Implantable Pacemaker. Journal of Computers. 2008;3(8):49-57.

[199] Tokic M, Palm G. Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In: Annual conference on artificial intelligence. Springer; 2011. p. 335-46.

[200] Matloff N. Introduction to discrete-event simulation and the simpy language. Davis, CA Dept of Computer Science University of California at Davis Retrieved on August. 2008;2(2009):1-33.

[201] Chang HJ, Wu CH, Ho JF, Chen Py. On sample size in using central limit theorem for gamma distribution. Information and Management Sciences. 2008;19(1):153-74.

[202] Sahoo RR, Sarkar S, Ray S. Defense against On-Off attack in trust establishment scheme for wireless sensor network. In: 2019 2nd International Conference on Signal Processing and Communication (ICSPC). IEEE; 2019. p. 153-60.

[203] TRACEMALLOC - Trace memory allocations - Python 3.10.2 documentation;. Accessed: 2022-02-08. Available from: https://docs.python.org/3/library/tracemalloc.html.

[204] Amin S, Gomrokchi M, Satija H, van Hoof H, Precup D. A survey of exploration methods in reinforcement learning. arXiv preprint arXiv:210900157. 2021.

[205] Mammeri Z. Reinforcement learning based routing in networks: Review and classification of approaches. IEEE Access. 2019;7:55916-50.

[206] Hasselt H. Double Q-learning. Advances in neural information processing systems. 2010;23.

[207] Al-Rawi HA, Ng MA, Yau KLA. Application of reinforcement learning to routing in distributed wireless networks: a review. Artificial Intelligence Review. 2015;43(3):381-416.

[208] Jiang J, Zhu X, Han G, Guizani M, Shu L. A dynamic trust evaluation and update mechanism based on C4. 5 decision tree in underwater wireless sensor networks. IEEE Transactions on Vehicular Technology. 2020;69(8):9031-40.

[209] Krishnaswamy V, Manvi SS. Trusted node selection in clusters for underwater wireless acoustic sensor networks using fuzzy logic. Physical Communication. 2021;47:101388.

[210] Melo FS. Convergence of Q-learning: A simple proof. Institute Of Systems and Robotics, Tech Rep. 2001:1-4.

[211] Hajar MS, Kalutarage H, Al-Kadri MO. TrustMod: A Trust Management Module For NS-3 Simulator. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE; 2021. p. 51-60.

[212] Sirisala S, Ramakrishna S. Survey: Enhanced trust management for improving QoS in MANETs. In: First International Conference on Artificial Intelligence and Cognitive Computing. Springer; 2019. p. 255-63.

[213] Pal D. A comparative analysis of modern day network simulators. In: Advances in Computer Science, Engineering & Applications. Springer; 2012. p. 489-98.

[214] Zhang Y, Wang W, Lü S. Simulating trust overlay in p2p networks. In: Int. Conf. on Computational Science. Springer; 2007. p. 632-9.

[215] Mármol FG, Pérez GM. TRMSim-WSN, trust and reputation models simulator for wireless sensor networks. In: 2009 IEEE International Conference on Communications. IEEE; 2009. p. 1-5.

[216] Lacage M, Henderson TR. Yet another network simulator. In: Proceeding from the 2006 workshop on ns-2: the IP network simulator; 2006. .

[217] Riley GF. Large-scale network simulations with GTNetS. In: Winter Simulation Conference. vol. 1; 2003. p. 676-84.

[218] Lakkakorpi J, Ginzboorg P. ns-3 module for routing and congestion control studies in mobile opportunistic dtns. In: 2013 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). IEEE; 2013. p. 46-50.

[219] Carneiro G, Fortuna P, Ricardo M. FlowMonitor: a network monitoring framework for the network simulator 3 (NS-3). In: Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools; 2009. p. 1-10.

[220] Open source. NetAnim;. Accessed: 30-08-2022. Available from: https://www.nsnam.org/wiki/NetAnim.

[221] Zhang W, Zhu S, Tang J, Xiong N. A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks. The Journal of Supercomputing. 2018;74(4):1779-801.

[222] Xiao Y. A fast algorithm for two-dimensional Kolmogorov–Smirnov two sample tests. Computational Statistics & Data Analysis. 2017;105:53-8.

[223] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.

Appendix A

Publications

1. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "RRP: A Reliable Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks." IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023.

2. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "3R: A Reliable Multi Agent Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks." Submitted for review to IEEE Internet of Things Journal. 2022.

3. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "DQR: A Double Q Learning Multi Agent Routing Protocol for Wireless Medical Sensor Network." 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm). 2022.

4. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "Security Challenges in Wireless Body Area Networks for Smart Healthcare." Artificial Intelligence for Disease Diagnosis and Prognosis in Smart Healthcare, CRC Press, 2022.

5. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "A Robust Exploration Strategy in Reinforcement Learning Based on Temporal Difference Error." 35th Australasian Joint Conference on Artificial Intelligence. 2022.

6. Muhammad Shadi Hajar, Harsha Kalutarage, and M. Omar Al-Kadri. "TrustMod:

A Trust Management Module For NS-3 Simulator." 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2021.

7. Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kumara Kalutarage. "A survey on wireless body area networks: Architecture, security challenges and research opportunities." Computers & Security 104 (2021): 102211.

8. Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kalutarage. "LTMS: A lightweight trust management system for wireless medical sensor networks." 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2020.

9. Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kalutarage. "ETAREE: An effective trend-aware reputation evaluation engine for wireless medical sensor networks." 2020 IEEE Conference on Communications and Network Security (CNS). IEEE, 2020.