Trust-aware and Cooperative Routing Protocol for IoT Security

Nabil Djedjig[a,b,*], Djamel Tandjaoui[a], Faiza Medjek[a,b], Imed Romdhani[c]

[a]*Research Center on Scientific and Technical Information, 03, Rue des Frres Aissou, Ben Aknoun, Algiers, Algeria*
[b]*Departement Informatique, Facult des Sciences Exactes, Universit de Bejaia, 06000 Bejaia, Algeria*
[c]*Edinburgh Napier University, School of Computing, 10 Colinton Road, EH10 5DT, Edinburgh, UK*

Abstract

The resource-constrained nature of IoT objects makes the Routing Protocol for Low-power and Lossy Networks (RPL) vulnerable to several attacks. Although RPL specification provides encryption protection to control messages, RPL is still vulnerable to internal attackers and selfish behaviours. To address the lack of robust security mechanisms in RPL, we designed a new Metric-based RPL Trustworthiness Scheme (MRTS) that introduces trust evaluation for secure routing topology construction. Extensive simulations show that MRTS is efficient and performant with respect to packet delivery ratio, energy consumption, and nodes' rank changes. In addition, a mathematical modelling analysis shows that MRTS meets the requirements of consistency, optimality, and loop-freeness, and that the proposed trust-based routing metric has the isotonicity and monotonicity properties required for a routing protocol. By using game theory concepts, we formally describe MRTS as a strategy for the iterated Prisoner's Dilemma and demonstrate its cooperation enforcement characteristic. Both mathematical analysis and evolutionary simulation results show clearly that MRTS, as a strategy is an effective approach in promoting the stability and the evolution of the Internet of Things network.

Keywords: RPL, Secure Routing, Internet of Things, Trust Management, Game Theory, Cooperation Enforcement.

1. Introduction

The Internet of Things (IoT) is a new communication paradigm that affects our daily lives in many domains, such as healthcare, home and building automation, automobiles, urban, and industrial appliances. The IoT-based networks are more likely formed of Low-power and Lossy Networks (LLNs), composed of various heterogeneous wireless technologies (objects), such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, etc. In these technologies, computing and communication systems are seamlessly embedded [1]. IoT's objects are characterised both by their strong resource constraints and by their lossy communication links. Indeed, these objects have limited processing power, memory, and energy supply, in addition to a high loss rate, a low throughput, a limited frame size, and short communication ranges [2][3]. Such limitations raised several challenges for industry and academic research community, for example, scalability, routing, and security.

This last decade, several routing solutions for LLNs were suggested. Finally, the Internet Engineering Task Force (IETF) ROLL (Routing Over Low power and Lossy networks) [4] working group has developed and standardised the Routing Protocol for Low-power and Lossy Networks (RPL) [5]. One major issue for the IoT is the routing security that researchers consider as a critical requirement [3] [6]. In the spate of the RPL specification defined cryptography-based mechanisms to ensure control messages integrity and confidentiality against outsider attackers [5], nonetheless, RPL still vulnerable to various known and new internal threats, that have been extensively studied in the literature [7][8].

Because trusting the objects participating in the routing process is crucial for the well-functioning of the network, we focus our research study on addressing RPL weaknesses in terms of routing security and proposing a security scheme for RPL based on trustworthiness between nodes. In this paper, we propose a Metric-based RPL Trustworthiness Scheme (MRTS) that enables secure routing by avoiding malicious nodes, and calculating and choosing the most trusted path from the source

*Corresponding author
Email address: djedjig_nabil@cerist.dz (Nabil Djedjig)

node to the root. We introduced MRTS initially in [9]. Firstly, this paper is a revision[1] of our previous work [9] where new elements and components are added to enhance MRTS performance in term of security, lifetime and routing requirements. Secondly, this paper extends the work in [9] with a simulation validation and a mathematical analysis.

Cooperation and collaboration are considered critical in the development of trust relationships among participating nodes for secure operations of the network [10]. According to Buttyan et al. [10], cooperation reinforces trust because trust is about the ability to predict the behaviour of another party, where the cooperation makes predictions more reliable. Surely, MRTS demands nodes to cooperate to improve the detection of untrusted nodes, and thus to enforce the routing security. Thus, MRTS can be seen as a strategy in which punishment Mechanism (i.e., isolation of untrusted node) is introduced to motivate nodes to be cooperative. In this paper, we demonstrate that in cooperation enforcement, the MRTS strategy is as good as other famous strategies like the tit-for-tat and the Spiteful strategies.

We summarise our contributions as follows.

1. We present a revision of MRTS [9]. We added ETX as a new parameter for trust calculation. We changed the functionality of the trust metric flag to make MRTS more flexible (i.e., secure and non-secure modes). We modified the parent selection process to extend nodes' energy.
2. We evaluate MRTS performances and give simulation results.
3. We provide a mathematical analysis of MRTS routing and its ERNT metric.
4. We perform mathematical analysis and a simulation study of MRTS as a strategy for cooperation enforcement, using game theory concepts.

This paper is an extension of our previous works [9] and [11] by adding simulation and mathematical validations. The rest of this paper is organised as follows.Section 2 presents a background of trust definition, RPL protocol, and its vulnerabilities. Section 3 sketches the related works for securing RPL. Section 4 gives a presentation of the components and the functioning of MRTS. Section 5 reports simulation-based performance study, a mathematical analysis, and a discussion on MRTS and its security features. Section 6 is devoted to mathematical analysis and a simulation study of MRTS as a strategy in the iterated (repeated) Prisoner's Dilemma game for cooperation enforcement. Finally, the last section concludes the paper and gives future works and perspectives.

[1]Throughout this paper, we use the terms enhancement and revision interchangeably.

2. Background

2.1. Trust and Trust Management

Trust is a subject of strong theoretical significance and real meaning. It is a very complicated concept that researchers saw and interpreted in many different ways under different contexts. One definition of trust is the relationship between different actors such as persons, entities, objects or actions. Thus, the trustor evaluates the trustee to assess its trustworthiness to perform some actions on its behalf. The evaluation process takes into consideration the history of the trustee's behaviour towards parties with which the trustee interacted previously. In the context of IoT, the trustworthiness (trust value) of a node is a scalar that defines the observed experiences of the node over a period. It quantifies the positive and negative interaction of each node towards its neighbours based on specified properties. In the literature, there exist different models and methods for trust management and calculation for IoT. Djedjig et al. [12] presented more details about trust management properties and models for IoT based networks.

2.2. The Routing Protocol for Low-Power and Lossy Networks

2.2.1. RPL Overview

RPL [5] is the first standardised routing protocol specially designed for LNN networks. RPL is a proactive distance-vector routing protocol that constructs a logical representation of the network topology as a set of Destination Oriented Directed Acyclic Graphs (DODAGs) through which data packets are routed. As depicted in Figure 1, in each DODAG, nodes are connected to the Border Router (BR) - edge router/gateway. A backbone link connects the BR to the Internet and other BRs. To construct the topology, RPL uses DIO (DODAG Information Object), DIS (DODAG Information Solicitation) and DAO (DODAG Destination Advertisement Object) control messages and a Trickle timer. To support routing optimisation and calculate the best paths to route traffic, RPL uses an Objective Function (OF) [13] and node, link metrics and constraints, or both [14]. In RPL, each object has a Rank (R), which determines the individual position of a node relative to the BR and other nodes within a DODAG. The Rank rule states that the Rank should be monotonic; thus, Rank values should

increase from the BR towards the leaf nodes, and vice-versa. If inconsistencies happen involving changes in the topology, the nodes reset the Trickle timer to a lower value, and thus, control messages transmission rate will be fastened. Nodes use Global Repair (GR) and Local Repair (LR) mechanisms to fix links and nodes failures, and other inconsistencies. Once GR or LG triggered, the nodes reset their trickle timers and update their respective parents' lists and Ranks.

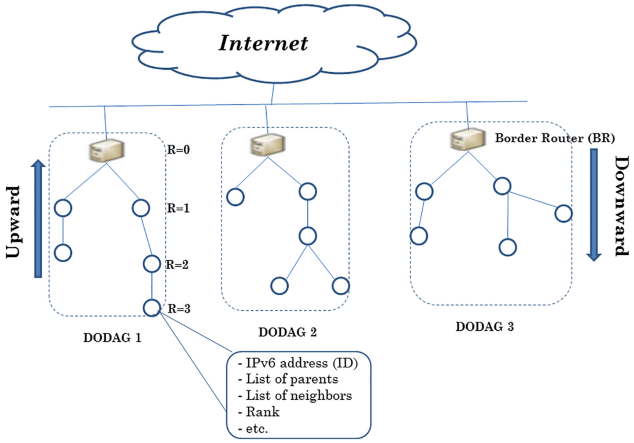


Figure 1: RPL Topology and Components.

2.2.2. *Routing Attacks against RPL*

RPL is vulnerable to a large variety of attacks, which researchers have treated in the literature [7][8]. Nowadays, several classifications for RPL threats exist. In our earlier study [7], we proposed two main classes: the Novel RPL Specification-based attacks and the Existing routing attacks tailored to the context of RPL. The first class includes the rank, neighbour, and version number attacks, while the second one includes the hello flooding, selective forwarding, Sybil, wormhole, and Blackhole attacks. In the following, we give the definitions of the two attacks addressed in this paper.

1. Rank Attacks: there exist several variants of the Rank attack, namely Decreased Rank Attack [15], Rank Attack [16], Worst Parent Attack [17], and Increased Rank Attack [18]. These attacks lead to generate loops in the network, to exhaust node resources, and to congest the network. In this paper, we give particular attention to the decreased rank attack. In this attack, the malicious node illegitimately advertises a better Rank equal to a lower Rank value inducing other nodes to select it as a parent.
2. Blackhole Attack [19][20][21]: in this attack, the malicious node drops all packets (control and data packets) routed through it. In the literature, researchers consider this attack as a DoS attack. Indeed, the Blackhole attack is more dangerous if combined with Rank or sinkhole attacks since the attacker is in a position where normal nodes route colossal traffic through it. This attack increases the number of exchanged DIO messages, which leads to instability of the network, data packets delay, and thus resources exhausting.

3. Related Works

Attacks take place at different levels in the IoT infrastructure. Although there exist several state-of-the-art works that address the IoT security, in this section, we focus on IoT's networking security, especially the solutions to secure the routing protocol RPL. Several works have presented in-depth analysis and classifications of RPL's vulnerabilities regarding different attacks, such as Rank, version number, neighbour, Sybil and CloneID, sinkhole, Blackhole, selective-forwarding, DIS, and DAO attacks [7][8][18]. Other studies have introduced several security measures to counter such attacks. Following we classify the proposed security efforts for the RPL protocol onto two classes: The IDS-based and the trust-based.

3.1. *IDS-based Security Solutions*

A growing number of works proposed Intrusion Detection Systems (IDS) as a technique to detect or isolate attacks against RPL. For instance, Raza et al. [22] proposed SVELTE, a hybrid IDS for IP-based IoT, where IDS modules were placed both in the BR and in constrained nodes. SVELTE targets spoofed or altered information (Rank attack), sinkhole, and selective-forwarding attacks. Furthermore, the authors proposed a distributed mini-firewall to protect the network against external attackers. However, SVELTE main drawbacks are the high false detection rate and the lack of DIO synchronisation. Pongle et al. [23] proposed a hybrid anomaly-based IDS to detect wormhole attacks. Authors used Neighbour Discovery/Verification Based techniques for the detection of these attacks. In this solution, the monitoring in-network nodes gather information about their respective neighbours and changes on the network (RSSI) and send them to the BR. This later analyses received data to detect intruders and then make decisions. Le et al. [16] proposed a hybrid specification-based IDS for securing RPL against topology attacks (Rank, sinkhole, and neighbour attacks). In

this approach, nodes monitor routing information conveyed in control messages to detect attackers. They used an Extended Finite State Machine (EFST) with statistic pieces of information about transitions and states for RPL. In this IDS, a cluster head requests its members to report its topology information periodically and process this information using the EFST. Napiah et al. [24] proposed a compression header analyser based IDS (CHA-IDS) to detect HelloFlood, Sinkhole, and Wormhole attacks in a 6LoWPAN. They used the Best First Search (BFS), Greedy Stepwise (GS), and the Correlation-based Features Selection (CFS) algorithm for feature engineering and selection. Simulation results showed that the J48 Machine Learning (ML) algorithm performs better than other classifiers for that specific configuration. Even though this approach presents a good background for IoT ML-based IDS, still authors considered a very small network of 8 nodes. Furkan et al. [25] proposed a Deep Learning (DL) model as IDS to detect routing attacks against an IoT RPL-based network. Simulation results showed good performance in term of IDS accuracy. However, the fitting time of the DL model was too long. Indeed, the machine and deep learning methods are too greedy in term of computation and storage for IoT devices.

The proposed IDS solutions for IoT depend on the availability of information conveyed within some packets or in datasets generated from exchanged packets. However, these packets can be lost if an attack forces nodes to drop their packets or other nodes' packets, thus making the overall IDS disturbed. From another side, if the malicious node is smart and does not trigger the attack continuously, the IDS might not detect it, and thus the attacker could participate readily in the network operations.

3.2. Trust-based Security Solutions

These last years, more research works are addressing the problem of trust management for different networks of the IoT. For instance, authors in [26] and [27] are considered the pioneers of the concept of distributed IoT trust management. In the proposed hierarchical trust management protocol for IoT, each node calculates the trust level of other nodes using social relationships metrics: honesty, cooperativeness, and community-interest, and relying on both direct service experiences and indirect recommendations, where recommendations are collected at the time nodes encounter each other through social contacts. One drawback of the suggested protocol is that the metrics are calculated using the energy of the node as a parameter. As a consequence, if a normal node is surrounded by selfish nodes, it will consume more energy, and it can be considered as non-trusted while it is trusted. Furthermore, a node may not collect enough recommendations to make informed decisions about other nodes. Chen et al. [28] proposed Community of Interest dynamic hierarchical trust management (COI-HiTrust) protocol that integrates mobility in trust evaluation, and where trust protocol parameter settings can be dynamically adjusted in response to changing environments. Authors used COI-HiTrust in the context of a MANET network. Authors in [29] proposed a 3-tier cloud-cloudlet-device hierarchical trust-based service management protocol (IoT-HiTrust) that eliminates the problems of the protocol in [26] and [27]. Authors used cloud servers to stock many recommendations toward each trustee node.

The above-cited works present robust trust solutions to address security in service-oriented and social-oriented IoT systems, i.e., for the aim of securing service composition and management in the context of social IoT applications. The proposed schemes dealt with misbehaving owners of IoT devices that provide services to other IoT devices in the system. Furthermore, the authors focused on Wireless Sensor Networks (WSNs) or Mobile Ad hoc NETworks (MANETs). These protocols are based on the work of Bao et al. [30]. Nevertheless, the authors did not address the trust-management to secure the RPL routing protocol against insider attackers. [28] and [29] works differ from our work as follows.

- Our work aims to propose a solution to secure routing in RPL-based networks, and not to select trusted devices according to their services as in [29].
- In our solution, the IDS is a part of the trust mechanism where it plays the role of a detector for the calculation of honesty component, while in [28] COI-HiTrust is used as a technique for the detection of intrusions in a community of interest.
- MRTS differs from the work of Bao et al. [30] and Chen et al. [28] [29] by the use of the recommendations. Indeed Bao et al. and Chen et al. use recommendations in the case of n-hop neighbours ($n > 1$), while MRTS uses recommendations even for 1-hop neighbours. This is because the more the information around a given node, the more other nodes can judge the certainty about it.

Almost all existing works for trust management in IoT are based on social IoT networks. Only a few works applied the trust concept to secure RPL. For instance,

Karkazis et al. [31] introduced the Packet Forwarding Indication (PFI) metric to build trust knowledge as a trust-related metric for RPL. In this approach, each node transmits a packet to one of its neighbours and listens whether this neighbour forwards the packet or not. Then, it calculates the probability for this packet to travel along the path successfully. The drawback of this method is the fact that each node takes a decision based only on its knowledge. Thus, if this node misbehaves, it will choose a failing path rather than a trusted one. Djedjig et al. [11] introduced a new trust-based metric for the construction of the RPL topology. In this approach, nodes cooperate to calculate trust metric of their respective neighbours based on nodes behaviours and some trust components (energy, honesty). One drawback of the solution is that the authors did not consider the trust value along the path (trust inference problem), which induce to not selecting the most secure paths. Khan et al. [32] proposed a centralised trust-based model for managing the reputation of every node participating in RPL-based network. In this model, each node relies on packets routed across the network to calculate direct trust for other nodes, thus elaborating positive and negative experiences with other nodes. The gathered trust information is then transmitted to a central entity, which evaluates the interactions between network nodes and gives them a global reputation. This solution uses direct trust exclusively, which makes it vulnerable to attacks related to trust mechanisms, such as bad-mouthing and good-mouthing attacks. Airehrour et al. [33][34] proposed SecTrust-RPL: a trust-aware RPL routing protocol to secure RPL from routing attacks. In SecTrust, the trust calculation process evaluates the trustworthiness of a node based on direct and indirect packet forwarding behaviour between linked and 2-hops nodes, respectively. Although SecTrust uses indirect trust observation, a node recommendation depends only on the neighbour of its indirectly linked neighbours (the parent of its parent). In other words, the indirect trust of a node is calculated based only on one recommendation of the intermediate neighbour, which makes it vulnerable to Bad-mouthing and Good-mouthing attacks. Seyyed and Fereidoon [35] proposed DCTM-IoT, a dynamic and comprehensive trust model for IoT which have a multi-dimensional vision of trust. The authors integrated several parameters in trust calculation, such as packet forwarding indicator, ETX, energy, and mobility. Besides, the nodes calculate trust using direct and indirect observations from neighbours. The authors claim that their programmed code size (48.28kbyte) is less than the objects' (Tmote Sky) available memory (48kbyte) which is not valid. From one side, too much information (historical) are used in this model and need to be stored and handled, thus making the solution not lightweight for constrained objects. From another side, the authors did not present nor how they integrated the model to RPL (recommendations, trust propagation, new objective function), neither how they used the detection of attacks with the trust calculation process. Lahbib et al. [36] proposed LT-RPL, Link reliable and Trust aware model for RPL protocol. In this approach, periodically, nodes send the node ID, neighbour ID, remaining energy percentage, the packet forwarding ratio, the Packet Reception Ratio(PRR), the Packet Error Rate (PER), the Expected Transmission Count (ETX), the transmission delay as well as the entity time to a trust manager. The trust manager stores trust-related data and evaluates the trust of each node. Authors did not explain how they calculated the recommendations. The proposed trust mechanism can counter grey-hole and black-hole attacks; however, there is not an IDS or a mechanism to detect other attacks such as version number and Sybil attacks. Besides, since the trust manager handles the storage and computation tasks, this solution is vulnerable to a one-point of failure. Kiran et al. [37] proposed a trust-based DDOS attack detection approach. The authors used packet frequency within a time interval as a trust indicator. The root node calculates the data frequency rate and maintains lists of nodes that crossed this rate for several intervals. Each time a node appears in a list, its trust value diminishes. When the trust value is under a threshold, the node will be classified as malicious, and the root node sends its identity to other nodes. The nodes receiving the malicious node identity, discard it from the routing operation. From one side, this approach can only detect DDOS attack. From another side, it is centralised making it vulnerable to a one-point of failure.

As highlighted above, the proposed security models for RPL focus either on attack detection or on trust management. In the case of IDS-based solutions, malicious (i.e., untrusted) nodes could divert the IDS, and thus could be selected in the routing process. Also, even though the IDS can detect and isolate attackers, still the selected path is not the most secure since some nodes could be, for example, selfish. In the case of trust-based solutions, some of the proposed solutions use only packet forwarding (or packets rate) as trust indicator parameter[2], which is not enough to assess the node's trustworthiness. Other solutions do not take into consideration the quality of a path in term of QoS, while

[2]Throughout this paper, we use the terms parameters, criteria, and components interchangeably in the context of trust evaluation.

others do not use mechanisms to detect several types of attacks to secure the routing paths strongly.

To benefit from the advantages of both IDS-based and trust-based solutions, avoid their disadvantages, and solve both network QoS and security requirements for an RPL-based network, we propose the Metric-based RPL Trustworthiness Scheme (MRTS); an RPL secure scheme that integrates both trust-management and IDS techniques and uses both QoS metrics and selfishness and honesty[3] parameters in trust computation. The MRTS trust-mechanism cooperates with an IDS, which monitors and detects malicious nodes, and thus translates it to an honesty parameter.

4. Metric-based RPL Trustworthiness Scheme

MRTS is a cooperation based trust mechanism in which each node evaluates the behaviour and calculates the trustworthiness of its neighbouring nodes relying not only on its direct observations but also on its neighbours' indirect observations, as presented in [9]. The more the information around a given node, the more other nodes can judge the certainty about it.

One issue of MRTS as it is defined in [9] is that it relies only on node metrics (energy, honest and selfishness) to select the best bath to route traffic. If nodes are honest and are not selfish then, the energy will be the principal metric to select the parent, and thus some nodes along the selected trusted path will consume more energy than other nodes, which results in unbalanced energy consumption, thus reducing the lifetime of those nodes. Furthermore, MRTS does not consider link metrics, thus reducing the routing quality, such as the packet delivery ratio. Nevertheless, link metrics are essential to assess the reliability of the route. This routing property is essential for IoT applications since reliable routes provide a high delivery ratio. Researchers proposed several methods to estimate the reliability of a route, such as the Received Signal Strength (RSS), the Link Quality Level(LQL), and the expected number of retransmissions (ETX) [38]. ETX metric estimates the average number of transmissions and retransmissions required to send a data packet to a neighbour. ETX is one of the widely used link metrics to enhance RPL performances, where the lower the ETX, the better the quality of the link. In addition, there exist several lightweight implementations of ETX for the RPL routing protocol.

For the above-presented arguments, and to balance the energy consumption of the nodes while promoting routes with higher packets delivery ratio, we extended MRTS with the ETX metric.

Figure 2 resumes how MRTS has integrated ERNT -the new trust-based metric- to DIO messages. In the present work, we changed the use of the (1 bit) flag T (See Figure 2) to indicate the security status (policy) of the network. Thus, when T is set to 1, the active mode is enabled, and the nodes perform the security check. Whereas, when T is set to 0, the passive mode is enabled, and the nodes do not perform any security check. For an in-depth understanding of MRTS and ERNT object, readers can refer to [9]. Algorithm 1 summarises the overall functioning of MRTS using ERNT, while Table 1 presents different notations used to describe MRTS.

4.1. Trust Metric Parameters

MRTS uses a combination of four parameters to evaluate nodes' trustworthiness: selfishness, honesty, ETX, and energy. Still, MRTS is flexible and adjustable by adding or removing behavioural components specific for a given IoT application.

4.1.1. Energy

The energy of the node is a QoS trust component. It refers to the level of expectation of node i that the node j has sufficient energy to achieve its functionalities. The energy trust between node i and node j is the remaining energy (ER) percentage of the node j estimated by the node i and vice versa. In IoT, the nodes consume mainly their energy while receiving and sending packets. There exist different approaches to calculate the energy. According to the energy model in [40], the energy consumed by a node i sending k bits data to the node j, denoted by E_{mt}, is calculated according to equation 1. E_{elec} is the electronics energy (i.e. the energy required for the transmitter as well as the receiver circuitry), E_{amp} is the energy dissipation for transmitting amplifier, and d is the distance from node i to node j. The energy consumed by node j receiving the k bits data, denoted by E_{mr}, is calculated according to equation 2. In RPL topology, every node communicates with its neighbours and sends data with the power level corresponding to the communication range of the node. Therefore, d is equal to the communication range.

$$E_{mt}(i) = k * (E_{elec} + E_{amp} * d^2) \quad (1)$$

$$E_{mr}(i) = k * E_{elec} \quad (2)$$

Initially, $ER(i)(t)$ is equal to the maximum energy E_{max}, i.e., at $t = 0, ER(i)(0) = E_{max}$. The energy spent by a

[3]The honesty parameter indicates if a node is malicious or not.

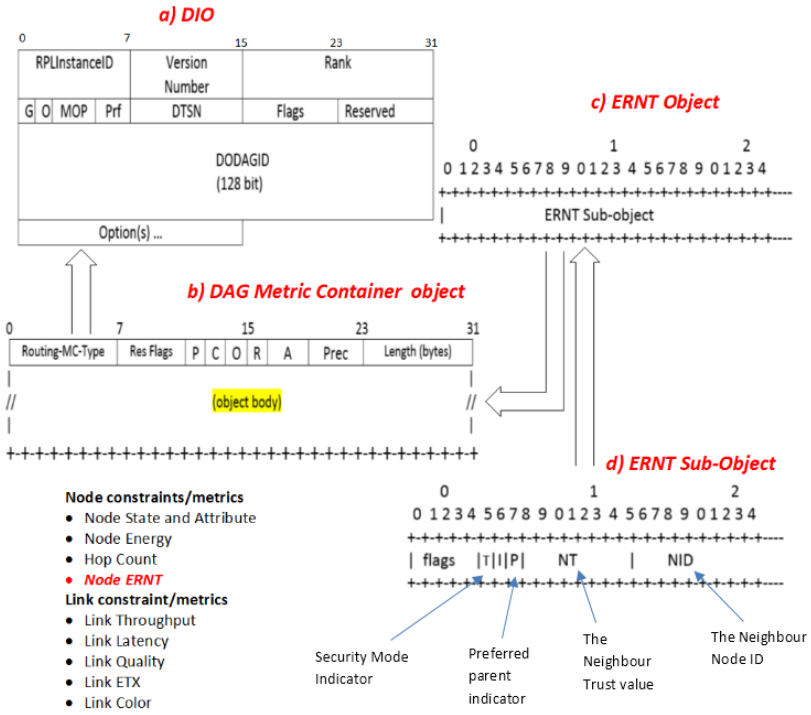


Figure 2: ERNT Object and ERNT sub-objects within the DIO DAG-Metric-Container Option

node i is the sum of the energy consumed in message transmission, and the energy consumed in message reception. Thus, the node i calculates its remaining energy as is given in equation 3.

$$ER(i) = ER(i) - (E_{mt}(i) + E_{mr}(i)) \quad (3)$$

Each node reports its residual energy to its neighbours periodically. The energy trust value $T_{ij}^{ER} \in [0,1]$ is equal to the ratio $ER_{ij(t)}$ and E_{max} as in equation 4, where $ER_{ij}(t) = \min(ER_{reported}(t), ER_{estimated}(t)$ and $ER_{estimated}(t) = ER(j)(t)$.

$$T_{ij}^{ER}(t) = \frac{ER_{ij}(t)}{E_{max}} \quad (4)$$

4.1.2. Selfishness

A selfish node is a node that intends to limit its resource expenditure while attempting to consume the resources of others. It can be calculated as a distributed and collaborative score. By using techniques such as overhearing and snooping [41], the node i evaluates the node j during a period P and decides if j is selfish or not. Assuming that an application requires minimum energy denoted by E_{min}. If ER(i)(t) is greater than E_{min}, the node i behaves correctly; if ER(i)(t) is less or equal to E_{min}, it does not take part in forwarding packets any longer and uses, for example, its energy for transmissions of its packets, which implies it is more likely to become selfish. Therefore, during the trust calculation phase, MRTS allows some degree of selfishness for the nodes to save their resources. Consequently, each node i increments the number N of time another node j does not cooperate $N = N+1$ in two cases: *i*) in the first case, the node j dropped data packets and $ER_{ij(t)} > E_{min}$. *ii*) in the second case, node j dropped control packet whatever its ER_{ij} value (this is because even if the remaining energy of the node j is less than the minimum energy, it is not tolerated to drop control packets since their critical importance for the network). When the number N exceeds a threshold $T_{selfish}$, the node i considers the node j selfish (Equation 5, where N is reset at the end of the period P.). This way, the nodes find a trade-off between energy and selfishness.

$$T_{ij}^{Selfish,new}(t) = \begin{cases} 0 & \text{if } N(t) \geq T_{selfish} \\ 1 - (\frac{N(t)}{T_{selfish}}) & \text{else.} \end{cases} \quad (5)$$

4.1.3. Honesty

The honesty parameter signals whether a node is malicious or not. Hence, the node i evaluates the node j behaviour to decide if j is compromised or not. To this end, some approaches use intrusion detection systems (IDS)

Table 1: Terminology

Notation	Description
MRTS	Metric-based RPL Trustworthiness Scheme
ERNT	Extended RPL Node Trustworthiness: Trust object that conveys trust values and related information, where T field for setting the security mode, P flag to indicate the parent status (path cost), NT flag to indicate the trust value, and NID to indicate the node identifier
TOF	Trust Objective Function
ETX	Expected Transmission Count
ER	Remaining Energy
$\mathbf{T}_{\mathbf{ij}}^{Direct}(t)$	Measures Direct Trust of node i towards node j at time t
$\mathbf{T}_{\mathbf{kj}}^{Recom}(t)$	Recommendation of node k towards node j at time t received in ERNT objects
$\mathbf{T}_{\mathbf{ij}}(t)$	Measures Trust of node i towards node j at time t using direct trust and recommendations
$\mathbf{T}_{\mathbf{ij}}^{X}(t)$	Measures Trust of node i towards node j at time t for the component $X \in \{honesty, selfish, energy, ETX\}$. These correspond to $\mathbf{T}_{\mathbf{ij}}^{Honesty}(t)$, $\mathbf{T}_{\mathbf{ij}}^{Selfish}(t)$, $\mathbf{T}_{\mathbf{ij}}^{ER}(t)$, and $\mathbf{T}_{\mathbf{ij}}^{ETX}(t)$
$\mathbf{w_1}$, $\mathbf{w_2}$, $\mathbf{w_3}$, $\mathbf{w_4}$	weights associated to honesty, selfishness, energy, and ETX parameters, respectively
$\mathbf{ER_{ij}}(t)$	Remaining Energy assessment of node i toward node j at time t
$\mathbf{T_{Selfish}}$	Selfishness threshold. The number of time a node is allowed to not forwarding other nodes packets
$\mathbf{T_{Trust}}$	Trust threshold
P	Monitoring period for selfishness assessment
$\mathbf{PC_i}$	Measures the path cost of the node i. This corresponds to the minimum of on-path nodes' trust values from the source node i to the destination BR
SOP	Set Of Parent
MRHOF-RPL	The Minimum Rank with Hysteresis Objective Function. The objective function that selects routes that minimize ETX

based on a set of anomaly detection rules [22][23]. In MRTS, each node i implements an IDS to monitor and detect malicious behaviours. If the IDS triggers an alert against a node j, the monitoring node i considers the node j dishonest and attributes to it an honesty-trust-value of 0 as in equation 6. The details of attacks detections by IDS are beyond the scope of this paper.

$$T_{ij}^{Honesty,new}(t) = \begin{cases} 0 & \text{if node j misbehaves} \\ 1 & \text{else.} \end{cases} \quad (6)$$

4.1.4. ETX

ETX is a QoS trust component. *"The ETX of a path is the expected total number of packet transmissions (including retransmissions) required to successfully deliver a packet along that path"* [42]. It is a reliability metric used to enable routing protocols to find high-throughput routes, and thus to reduce energy consumption. To calculate $T_{ij}^{ETX}(t)$, ETX(t) is firstly normalized to $[0, 1]$ using the Min-Max-Normalization method in equation 7, where $ETX_{min} = 0$ and $ETX_{max} = 255$ (as normalized in ContikiRPL implementation [39]). Then, the equation 8 is applied.

$$ETX(t) = \frac{ETX(t) - E_{min}}{E_{max} - E_{min}} = \frac{ETX(t)}{E_{max}} \quad (7)$$

$$T_{ij}^{ETX}(t) = 1 - ETX(t) \quad (8)$$

4.2. Trust Evaluation

The trust value of a node, in the MRTS mechanism, is a combination of both direct observation and indirect recommendations as follows.

4.2.1. Direct Trust

Each node evaluates the trust value, $T_{ij}(t)$ of its 1-hop neighbour at time t. There exist several methods to calculate the trust value of an entity (in this case, a node), such as belief theory, Bayesian systems, Fuzzy logic, and weighted sum. Because RPL's objects have limited storage and processing capacities, we chose the weighted sum method to evaluate nodes' trustworthiness. We rely on the work of Bao et al. [30] to calculate direct trust, as depicted in equation 9 [30]; where w_1, w_2, w_3 and w_4 are weights associated with honesty, selfishness, energy, and ETX parameters. We use

Algorithm 1 MRTS Decision Process

Require: $NodesList, NeighboursList, T_{Trust}, T_{Selfish}, w_1, w_2, w_3, w_4, P, \alpha$
Ensure: $PreferredParent, Rank$
  if $NeighboursList = \varnothing$ **then**
    Construct the topology according to MRHOF-RPL in ContikiRPL implementation [39]
  else

    while 1 **do**
      if $ERNT.T = 0$ (passive mode) **then**
        Construct the topology according to MRHOF-RPL
      else {$ERNT.T = 1$ (active mode)}

        for all $j \in NeighbourList$ **do**
          (Calculate Direct Trust)
          Activate Promiscuous mode, watchdog mechanism, and IDS
          $\mathrm{ER_{ij}(t)} \Leftarrow \min(\mathrm{ER_{reported}(t)}, \mathrm{ER_{estimated}(t)}$
          $\mathrm{T_{ij}^{ER}(t)} \Leftarrow \frac{\mathrm{ER_{ij}(t)}}{\mathrm{E_{max}}}$

          $\mathrm{ETX_j(t)} \Leftarrow \frac{\mathrm{ETX_j(t)}}{\mathrm{E_{max}}}$
          $\mathrm{T_{ij}^{ETX}(t)} \Leftarrow 1 - \mathrm{ETX_j(t)}$
          $\mathrm{T_{ij}^{Selfish,new}(t)} \Leftarrow 1 - (\frac{\mathrm{N}}{(}\mathrm{t})\mathrm{T_{selfish}})$
          $\mathrm{T_{ij}^{Honesty,new}(t)} \Leftarrow \{0, 1\}$
          Execute equation 9
          Update Trust Table ($T_{ij}^{Honesty}(t), T_{ij}^{Selfish}(t), T_{ij}^{Direct}(t)$)
        end for

        for all $j \in NeighbourList$ **do**
          (Calculate Indirect Trust using recommendations)
          Execute equation 11, where $T_{kj}^{recom}(t) = ERNT.NT$

          Update Trust Table ($T_{ij}(t)$)
          Update ParentList ($T_{ij}(t) \geq T_{Trust}$)
        end for
        From ParentList, Select $T_{ij}(t)$ with greater PC_i
        Update Rank
        Build DIO with calculated values and forward
      end if
    end while
  end if
  return $PreferredParent, Rank$

equation 10 [30] to evaluate each behavioural parameter $X \in \{Honesty; Selfish\}$, where Δt is the trust update interval, $\mathrm{T_{ij}^{X}}(t - \Delta t)$ is the old observation, and $\alpha \in [0, 1]$. If α tends to 1, then trust relies more on new observations. Otherwise, if α tends to 0, then trust relies more on old observations. Since the remaining energy reflects the ability of a node to achieve its functionalities and ETX reflects the status of the link, the trust calculation for both rely only on new observations, as presented in sections 4.1.1 and 4.1.4, respectively.

$$\begin{cases} T_{ij}^{Direct}(t) = w_1 T_{ij}^{Honesty}(t) + w_2 T_{ij}^{Selfish}(t) \\ \qquad\qquad +w_3 T_{ij}^{ER}(t) + w_4 T_{ij}^{ETX}(t) \\ \mathrm{w_1 + w_2 + w_3 + w_4 = 1} \end{cases} \tag{9}$$

$$\mathrm{T_{ij}^{X}(t) = \alpha T_{ij}^{X,new}(t) + (1 - \alpha) T_{ij}^{X}(t - \Delta t)} \tag{10}$$

4.2.2. Indirect Trust

Because MRTS is a cooperative mechanism aiming to select the most secure path toward the root, after calculating the direct trust for each neighbour j, the node i uses the trust values received within the DIO messages (i.e., in the ERNT objects) from its neighbours k (recommendations received from recommenders k) at time t to calculate the final trust value of the node j, as in equation 11; where the final trust value is the average of the direct trust value calculated according to equation 9 and all recommendations received for that neighbour j in ERNT objects.

$$\mathrm{T_{ij}(t) = Avg(T_{ij}^{Direct}(t) + T_{kj}^{Recom}(t))} \tag{11}$$

If the node i receives recommendations for nodes that are not 1-hop neighbours, it will ignore them.

4.3. Trust Propagation and Update

4.3.1. Trust Propagation

In MRTS, nodes exchange, share, and update trust information through the quantitative and dynamic trust the RPL Node Trustworthiness metric; ERNT. The ERNT metric is an object, which is carried and propagated through the DAG Metric Container [5][14] of the DIO message [9]. As depicted in Figure 2, a set of ERNT sub-objects form the ERNT object. MRTS uses the ERNT object both as a constraint and as a recorded metric. The BR uses an ERNT sub-object as a constraint to indicate the trust threshold ($\mathrm{T_{Trust}}$) that nodes must use to include or eliminate nodes that are not trustworthy. Besides, the BR and each node participating in the construction of RPL and following MRTS uses ERNT as a recorded metric, by inserting one ERNT sub-object (a record) for each calculated (final) trust value, in addition

to the path cost. Indeed, the path cost value represents the preferred parent's trust value.

4.3.2. Trust Update

MRTS updates trust values either periodically or reactively. The periodic trust update is time-driven, where MRTS uses the trickle timer for sending DIO messages as a regulator, while the reactive trust update is even-driven, where MRTS uses global repair and local repair events as triggers. In our solution, when the IDS rises an alarm (detects attacks) or if the $T_{Selfish}$ is reached, the local repair or global repair is triggered. Otherwise, the trickle timer regulates the update.

When a node n receives DIO messages from its neighbours, it uses the information conveyed in these DIO messages to update its routing table. It calculates the trust values of its neighbours using the direct assessments and recommendations received in DIO messages (according to section 4.2). It then selects a set of trusted parents allowing it to reach the BR. It calculates the path cost through each potential parents and selects as preferred parent the one with the highest path cost value (according to section 4.4.1), which ensures the most trusted and reliable traffic routing to the BR. Finally, it generates and broadcasts a new DIO message containing the calculated trust values for each of its neighbours. All the neighbouring nodes repeat the process until the DODAG is reconstructed.

Once the construction is completed, the maintenance begins respecting the Trickle timer. The timer regulates the transmission rate of the control messages. Thus, in the stable state, the trust update interval of the trickle timer increases, and the transmission rate will be slowed, which signifies fewer control messages, and thus less computation (i.e., the network consumes less energy, memory and CPU). Otherwise, if there are inconsistencies (e.g., attack detection, selfish behaviour detection, and a new node joining the DODAG), which involve changes in the topology, the Trickle timer will be reset to a lower value, and transmission rate will be fastened, which implies more control messages and thus more computation.

To reduce the computation cost in terms of energy consumption due to trust update overheads, MRTS smooths out a small path cost (trust) increase or decrease. In the proposed solution, we consider a hysteresis threshold of 0,15 to avoid frequent parent changes, which helps maintain stability and conserve energy.

4.4. Attackers Isolation and Parent Selection

4.4.1. Parent Selection

The Trust Objective Function (TOF) of MRTS implements both nodes isolation and parent selection procedures. TOF is composed of two steps: the topology initialisation step (neighbours discovery) and the context-aware adaptive security execution step. The nodes execute the first step at deployment because they do not know their neighbours and thus could not evaluate their trustworthiness regarding honesty and selfishness. Since at deployment, all nodes have the same initial energy, the only parameter to use to construct the RPL topology is ETX along the path. We used ContikiRPL built-in function to calculate ETX. The preferred parent is the one with minimum ETX value, where ETX is calculated as the sum of ETX along the path (from the BR to the parent node).

After the initialisation, each node knows its neighbours. If secure mode is not activated (T flag set to 0 in the ERNT sub-object), the only parameter to use is ETX as in the first step, and the nodes use TOF to find best paths by selecting parents with minimum ETX values. If secure mode is activated, each node evaluates the path cost, selects a set of parents having trust value greater or equal to the threshold T_{Trust}, and selects its preferred parent. There exist several ways to compute path cost using a trust metric, which is known as the trust inference problem. According to TOF, each node i calculates its path cost, PC_i, through each reachable potential parent j. PC_i is a scalar value representing node characteristics along the end-to-end path. To meet the MRTS routing requirements of consistency, optimality, and loop-freeness [43], TOF defines the path cost PC_i as the minimum trust value of on-path nodes' from the source node i to the destination BR. Thus, the node i selects its preferred parent as the parent who is in the path having the highest path cost, where the best path is the one with the highest minimum trust value [9]. To simplify, we calculate PC_i as the minimum value between the potential parent path cost PC_j and $T_{ij}(t)$ for that parent j, according to equation 12.

$$PC_i = \min_{j \in \{SOP\} \& T_{ij}(t) \geq T_{Trust}} (PC_j, T_{ij}(t)) \qquad (12)$$

The node i will change its current preferred parent with a new preferred parent only if the path cost through this new parent is higher than the currently selected parent by at least the hysteresis threshold of 0,15. Unlike in [9], if some candidate paths have the same path costs then, the node i will choose as the preferred parent, the one having the higher remaining energy.

4.4.2. Attackers Isolation

Several methods exist to isolate untrusted node from participating in network operations. In MRTS, each node maintains a blacklist with the collaboration of the IDS. Once a node is classified as untrusted, it is added to that blacklist. As a result, normal nodes ignore all data and control packets coming from the blacklisted nodes and do not consider them any more in routing decision.

5. MRTS Evaluation

5.1. Simulation Study

5.1.1. Simulation Settings

We studied the MRTS performances and compared them to MRHOF-RPL and SecTrust-RPL (SecTrust for short) performances. We calculated the average packet delivery ratio (%), the average energy consumption, and the average rank changes corresponding respectively to the ratio of the packets delivered to total packets sent, the average rate of energy consumption by all nodes in the network, and the average number of parent switches. For our simulations, we used the lightweight and open source Contiki 2.7/Cooja simulator [39]. We simulated a network of 30 nodes with one BR placed in the centre and 29 Sky mote (TelosB) nodes senders placed randomly around the BR. Each Sky mote is powered by an 8MHz, 16-bit Texas Instruments MSP430 microcontroller with 10kByte of RAM and 48kByte of flash memory. 3 of the 29 nodes are attackers planted randomly within the network, which trigger Rank or Blackhole attacks. Table 2 shows the simulation parameters. We executed the simulations ten times with three different topologies, and we averaged the outputs of the simulations.

We set the trust threshold T_{Trust} to 0.5, the selfishness threshold $T_{Selfish}$ to 5, and α to 0.75. Because we believe that all four factors are equally important to select secure routes that respect good QoS, initially, we set the weights w_1, w_2, w_3 and w_4 equally to 0.25. As in this study we focus on the security issues for RPL routing, during the simulation, if the IDS detects a node as malicious, the normal nodes will adjust the weights associated to the malicious node such as w_1 to 1, and w_2, w_3 and w_4 equally to 0. Likewise, if a node detects another node as selfish, the normal node will adjust the weights associated to the selfish node such as w_2 to 1, and w_1, w_3 and w_4 equally to 0.

In the simulation, we used both time-driven and event-driven update. The computation process is triggered according to the trickle timer (time-driven) and if the IDS sends an alert or if the $T_{Selfish}$ is reached (event-driven).

Table 2: Simulation Parameters

Parameter	Value
Simulator	Cooja-Contiki 2.7
Simulation time	1h
Number of nodes	30
Network area	100m*100m
Range of nodes	RX:50%, TX: 50m, interference: 60m
Radio medium model	UDGM: Distance Loss
Traffic rate	1 packet sent every 10 seconds
Number of attacker nodes	3
Attacks	Rank/Blackhole
w_1, w_2, w_3 and w_4	0.25
$T_{Selfish}$	5
T_{Trust}	0.5
α	0.75

5.1.2. Simulation Results

Rank Changes. Figure 3 shows the average rank changes rate for MRHOF-RPL, SecTrust, and MRTS under Rank and Blackhole attacks. As the simulation progresses the average frequency of rank changes for MRHOF-RPL under both attacks is very high (Blackhole attack: 300 times the first 30mn to 450 times, and Rank attack: 120 changes the first 30mn to 380 changes). Indeed, the high rate of rank changes is due to a high rate of parent changes to handle topology instability caused by both attacks. Even though SecTrust shows significant improvement regarding the network stability over MRHOF-RPL, MRTS shows better results inducing more stability (Blackhole attack: 60 first 30mn to 80, and Rank attack: 50 first 30mn to 40).MRTS performs better than SecTrust because nodes collaborate to detect and isolate attackers quickly, which helps to maintain the stability of the network.

Packet Delivery Ratio. In addition to network congestion and packet collision, it can be observed from Figure 4 that the effects of Blackhole and Rank attacks on packet delivery ratio for MRHOF-RPL are disastrous (25-40%). Several causes can explain the observations. For instance, the fact normal nodes choose malicious nodes as a preferred parent to forward their packets, and the deletion of control packets making the topology unstable and unavailable. In contrary, MRTS maintained the packet delivery ratio quite high (up to 90%) since it uses IDS to detect attacks and provides a new routing scheme to isolate malicious nodes and maintain a secure topology. As a result, attacks on MRHOF-RPL

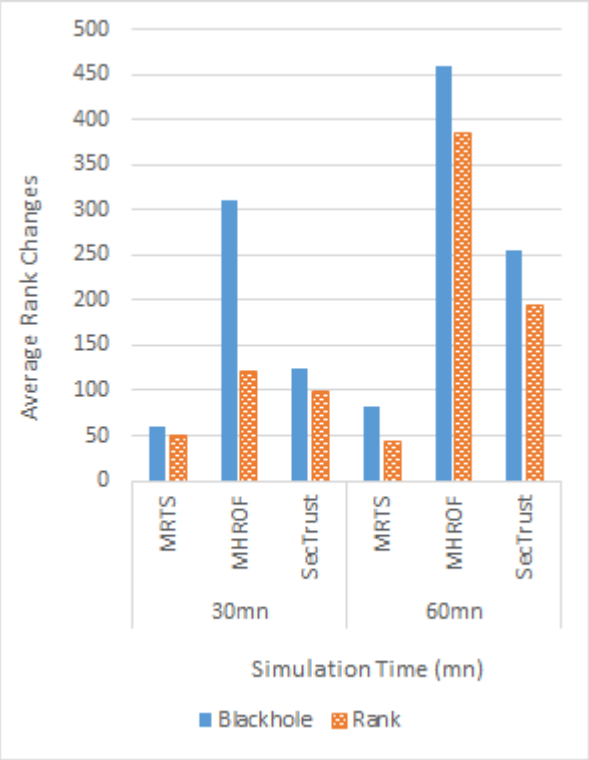


Figure 3: Average Node Rank Changes under Blackhole and Rank attacks for MRTS MRHOF-RPL, and SecTrust.

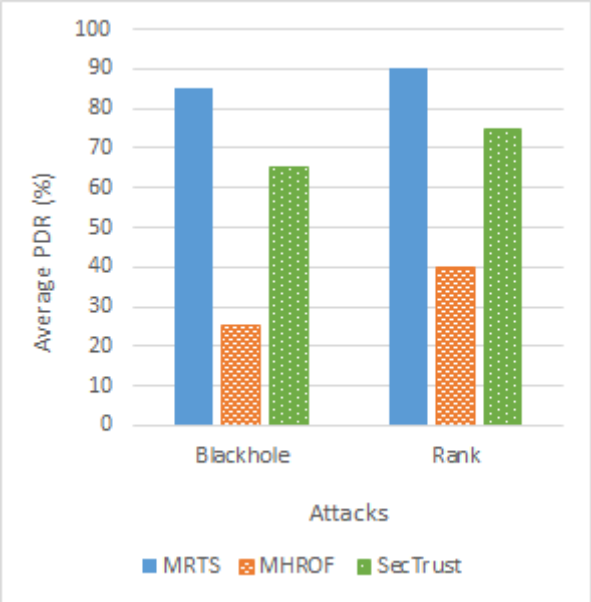


Figure 4: Average Packet Delivery Ratio under Blackhole and Rank attacks for MRTS, MRHOF-RPL, and SecTrust.

cause significant damages compared to MRTS. We can see that MRTS shows better packet delivery ratio compared to SecTrust. These results are related to the rank changes rate where MRTS provides more stable network over SecTrust, thus reducing packet loss.

Energy Consumption. In MRHOF-RPL network, some nodes consume more energy than others do because they tend more often to be chosen as preferred parent relying on their ETX; this is an issue since the higher energy cost to the chosen parents affects the entire network's lifetime. As depicted in 3 and Figure5, when MRHOF-RPL network is under attacks, nodes consume more energy due to topology instability and rank changes rate (i.e., due to parent changes). We explain the instability of the network by the fact that MRHOF-RPL does not have any mechanism to handle attacks. From Figure 5, we notice that the first 20-30mn, MRHOF-RPL and SecTrust consumed lower energy than MRTS. After some time, MRTS performed better because energy consumption is much more balanced among different nodes. The performance of MRTS is due to the fact, in addition to security (selfishness and honesty) and link quality (ETX) parameters, our solution takes into account the remaining energy for each node in routing decision. Indeed, under attacks, MRTS consumes the most energy in calculation and DIO transmissions, but once the malicious nodes detected and isolated, the topology becomes more stable, and thus the energy consumption rate decreases. Furthermore, as already stated, if two candidate parents have the same trust values, the node selects the one having the highest remaining energy.

5.2. MRTS Requirement

To forward packets to the border router, RPL can use either hop-by-hop forwarding scheme or source routing. Furthermore, since RPL is a distance-vector routing protocol, it uses the Bellman-Ford algorithm to calculate path cost [44]. In a weighted directed graph, this algorithm computes the shortest paths from a source node to a destination node. In this section, we validate MRTS relying on the study of Yang et al. [43]. According to the authors, a routing protocol consists of two components: a path calculation algorithm and a packet-forwarding scheme. Besides, a routing protocol needs three requirements to operate properly: consistency, optimality and loop-freeness. In this section, we demonstrate that MRTS combining ERNT, MRTS's Bellman-Ford algorithm [9], and either hop-by-hop forwarding scheme or source-routing meets these different requirements.

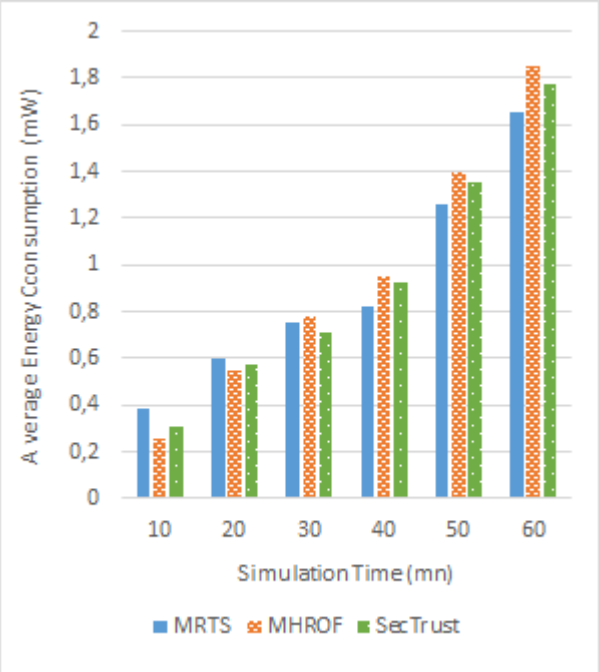


Figure 5: Average Energy Consumption over Time for MRTS, MRHOF-RPL, and SecTrust.

5.2.1. Network Model

MRTS based network is defined as a directed weighted graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges representing links between neighbouring nodes. Each edge, $e = (i, j)$ is associated to a positive weight corresponding to T_{ij}, where $e \in E$, $i, j \in V$, and T_{ij} is the trust evaluation of node i for its neighbour node j. Traffic within the network is multipoint-to-point, where all source nodes send data to a single destination node (BR). Thus, traffic is transmitted upward from source nodes u_n to a destination u_1, where u_1 is the BR. A path p from u_n to u_1 is denoted as $p < u_n, u_1 >= (u_n, u_{n-1}, ..., u_1)$. $next(p, u_k)$ denotes the next-hop of u_k on $p < u_n, u_1 >$, where $k = n, n-1, ..., 1$. In MRTS, $next(p, u_k)$ represents the preferred parent (1-hop: PP_k) that u_k stores in its routing table and uses to forward traffic (i.e., $next(p, u_k) = PP_k$). The path calculation algorithm of MRTS routing protocol (MRTS's Bellman-Ford algorithm [9]) is represented by a function MRTS(G,f,s,BR) that returns a path $p < s, BR >$ from the source node s to the BR. In addition, we denote $SP(p < u_n, u_1 >, v, u_1)$ the sub-path of path $p < u_n, u_1 >$ between on-path nodes v and u_1. In our network model, we consider two kinds of nodes: trusted and untrusted nodes. The first category represents legitimate nodes, with trust values $T_{ij} \geq T_{Trust}$. The second one represents nodes with trust values $T_{ij} < T_{Trust}$.

5.2.2. Consistency

According to [43], a routing protocol is consistent if whatever a node along a given path, the packet forwarding decision is consistent with each node in that path. In other words, if a node u_n decides to forward packets through the path $p < u_n, u_1 >= (u_n, u_{n-1}, ..., u_1)$, other on-path-$p < u_n, u_1 >$ nodes should forward packets through p-subpaths. It seems obvious that MRTS is consistent. For source routing, on-path nodes forward packets relying on packets headers, and thus, routing consistency is systematically satisfied. For hop-by-hop routing, because in MRTS the routing tree is constructed from the BR to leaf nodes, each intermediate node is, in fact, a preferred parent of each sending node along a given path. Hence, if a node selects a preferred parent as its next-hop ($next(p, u_n) = u_{n-1}$), it selects automatically the sub-path $SP(p < u_n, u_1 >, u_{n-1}, u_1) = p < u_{n-1}, u_1 >= (u_{n-1}, ..., u_1)$ through that parent u_{n-1} to forward packets. The process is recursive for each node on the sub-path.

5.2.3. Optimality

A routing protocol is optimal if it routes the traffic along the best path for every pair of nodes within the network [43]. MRTS uses *MRTS(G,f,s,BR)* path calculation algorithm allowing a source node s to calculate path cost (denoted PC^x) for each path $p^x < s, BR >$ to the BR, where x is the number of paths. For instance, if we consider three possible paths p^1, p^2, and p^3, with $PC^1 = 0.5$, $PC^2 = 0.8$, and $PC^3 = 0.7$ then, according to section 4.4, s will select p^2 as the best path to route the traffic.

5.2.4. Loop-freeness

A routing protocol is said to be loop-free if it does not create any packet forwarding loop. Besides, if the routing protocol uses hop-by-hop routing and Bellman-Ford algorithm to calculate valid paths, consistency is sufficient to ensure loop-freeness [43]. According to our analysis, MRTS is consistent, and consequently, it is loop-free. Otherwise, after the selection of the best path, the source node calculates its rank using the rank of its selected parent. If a node receives a DIO message from another node having a rank value greater than its rank value, the receiving node will discard the DIO message, and thus, will never select a child as a parent; therefore, loops will be avoided. In the case of source-routing, a source node can eliminate loops.

5.3. ERNT Mathematical Model

To ensure the proper operation of a given routing protocol (i.e. consistency, optimality and loop-freeness), authors in [43] identify the properties that a routing metric should have: isotonicity and monotonicity. The monotonic property could ensure the convergence of the routing algorithm, while the isotonicity property essentially affects the order of the paths weights and could ensure their convergence is optimal for distance vector protocols like RPL. As explained in [43][45][46], we represent a routing metric as an algebra on top of a quadruplet $(S, \oplus, f, \leq)$, where S is the set of all paths, f is a function that maps a path to a path cost, $\leq$ represents a total order of costs, and $\oplus$ is the path concatenation operation. According to the proved Lemma 2 in [43] *"if for every source $s \in V$, destination $d \in V$ the path weight structure $(S, \oplus, f, \leq)$ is left-isotonic and left-monotonic, there exists a lightest path from s to d such that all its sub-paths with source s are also lightest paths. Such a lightest path is called a D-lightest path"*. For ERNT metric, the algebraic path weight structure is $(S, \oplus, f(max), \leq)$ where $f(max) = max(p)$ is the maximum trust value $(1 - T_{ij}(t))$ along the path p. The order relation for ERNT is $\leq$, to minimise the trust of different paths p^x. Based upon the proved Theorems 7, 8 and 9 in [43], and given that MRTS uses MRTS's Bellman-Ford algorithm for path calculation [9], it is enough to demonstrate that ERNT is left isotonic and left monotonic for calculating lightest paths. In the case of source-routing, it is enough to demonstrate that ERNT is left isotonic

5.3.1. Isotonicity

A routing metric is left isotonic if the order relation between two paths is preserved if a common third path prefixes them. Formally, the algebraic structure $(S, \oplus, f, \leq)$ is left isotonic if $\forall a, b, c \in S, f(a) \leq f(b) \Rightarrow f(c \oplus a) \leq f(c \oplus b)$. This means, $(ERNT, \oplus, f(max), \leq)$ is left isotonic if $\forall a, b, c \in S, max(a) \leq max(b) \Rightarrow max(c \oplus a) \leq max(c \oplus b)$.

Let consider two given paths a and b so that $max(a) \leq max(b)$. For a given prefixed path c, three cases could occur:

1. In the case $max(c) \leq max(a) \leq max(b)$ we have:
   $\left.\begin{array}{l} max(c \oplus a) = max(a) \\ max(c \oplus b) = max(b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$
2. In the case $max(a) \leq max(c) \leq max(b)$ we have:
   $\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(c \oplus b) = max(b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$
3. In the case $max(a) \leq max(b) \leq max(c)$ we have:
   $\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(c \oplus b) = max(c) \\ max(c \oplus a) = max(c \oplus b) \end{array}\right\} \Longrightarrow max(c \oplus a) \leq max(c \oplus b)$

5.3.2. Monotonicity

A routing metric is left monotonic if the path cost will not decrease when prefixed by another path. Formally, $(S, \oplus, f, \leq)$ is left monotonic if $\forall a, c \in S, f(a) \leq f(c \oplus a)$. This means, $(ERNT, \oplus, f(max), \leq)$ is left monotonic if $\forall a, c \in S, max(a) \leq max(c \oplus a)$.

For a given prefixed path c, two cases could occur:

1. In the case $max(a) \leq max(c)$ we have:
   $\left.\begin{array}{l} max(c \oplus a) = max(c) \\ max(a) \leq max(c) \end{array}\right\} \Longrightarrow max(a) \leq max(c \oplus a)$
2. In the case $max(c) \leq max(a)$ we have:
   $\left.\begin{array}{l} max(c \oplus a) = max(a) \end{array}\right\} \Longrightarrow max(a) \leq max(c \oplus a)$

5.4. Discussion

Our solution is flexible and depends on the context according to weights associated with trust calculation, to the ERNT metric object flags, and to trust-related thresholds. From one side, nodes executing MRTS can dynamically modify the weights, w_i ($i \in \{1, 2, 3, 4\}$), according to rules specified by the context. Hence, our solution allows a trade-off between security effectiveness (honesty and selfishness) and QoS requirements (energy efficiency and ETX) depending on the weights dynamic changes defined by each IoT application. For instance, in our study, we set the weights initially equally, and then we adjusted them according to security breaches detection (honesty and selfishness). In another case, w_1 and w_2 could be greater than w_3 and w_4. If the energy saving is the most important, w_3 could have the greater value. In another case, an application could switch to active (secure) mode and sets w_1 and w_2 to greater values if the energy state is greater than a threshold (E_{min}), and switch to passive (non-secure) mode if the energy state is less than a threshold.

Even though some degree of selfishness is allowed for nodes to save their resources, in the present study we do not consider reintegration of isolated nodes once classified as untrusted. In some cases the reintegration and backup of untrusted nodes can be required (e.g. to support fault tolerance). Nevertheless, considering a reintegration mechanism raises new problems regarding the effectiveness of the solution, such as preventing nodes from abusing the mechanism. In this context, if the BR sets the I flag (in the ERNT object) to 1, the parent selection process permits the selection of untrusted nodes

in the set of parent, whereas, if I is set to 0, the parent selection process does not allow the insertion of untrusted nodes in the set of parents. This flexibility can be seen as a reintegration mechanism, in which the border router switches the I flag between 1 and 0 according to the necessity of the application context (fault tolerance). However, rules need to be designed to handle this reintegration carefully.

In the simulation study, we noticed good results regarding energy consumption. However, our solution could not always get the right balance between the issues of energy efficiency and security. The setting of the threshold parameters $T_{selfish}$ and T_{Trust} is a trade-off issue, and no value can fit all scenarios and criteria. For instance, in a scenario in which security is of great concern, T_{Trust} value needs to be as high as possible while $T_{selfish}$ value needs to be as small as possible. This way, malicious and selfish nodes would be isolated quickly. Nevertheless, the topology could be more frequently unstable, leading to more energy consumption. Besides, if T_{Trust} is too big lots of nodes might be isolated, and thus the proper functioning of the network can be affected.

Since MRTS is built upon the RPL protocol, it inherits both its advantages and disadvantages. Indeed, the developers of Contiki-ng[4] confirmed that because of supporting new functionalities from standards and Internet drafts, the implementation of RPL becomes more complex and thus gets a large ROM footprint. In MRTS, each node maintains the list of all its neighbours with the necessary information to calculate trust values. Hence, if the network scales up, the neighbours' list will increase, and obviously, nodes will need more storage capacity. Indeed, the storage limitation of LNN objects is still a big challenge, especially for large scale routing. As presented by Xiyuan et al. [47], the challenge is to find a balanced solution that reduces the memory overhead risk while improving the utilisation of the node capacity. In our future work, we are going to test the performance of MRTS in the case of large scale networks and highlight the issues and the solutions.

The collaborative (cooperation) isolation gives MRTS many advantages. From one hand, it permits to reduce false positives. From the other hand, since all neighbours cooperate in the evaluation of a given node, even if two successive nodes misbehave, MRTS can detect and isolate them.

Table 3 gives a comparative of MRTS with other solutions used to secure RPL from routing attacks.

[4]The contiki OS for Next Generation IoT Devices that implements new functionalities to enhance RPL. Readers can reach the documentation online at https://github.com/contiki-ng/contiki-ng/wiki

6. MRTS : a Strategy For Cooperation Enforcement

Following MRTS, the participation in the network operations is conditioned by the trust value of each node; this means, if TOF classifies a node as untrusted, it will be discarded from the network. As a result, there is no advantage for a smart, rational intruder to misbehave because it will be discarded from the network. As consequence, nodes in the network could achieve the effective cooperation, and MRTS can be seen as a stable strategy of the interactive nodes within a repeated game. The network will then obtain service of higher security and trust between the cooperating nodes. In this section, we introduce our system model and explain how we mapped MRTS into a strategy for the iterated Prisoner's Dilemma (PD)[5]. Then, we analyse the MRTS strategy formally and compare it to other strategies using simulation software, with regard to cooperation promotion and evolution.

6.1. System Model

We used the non-zero-sum non-cooperative iterated PD game as the conceptual foundation for modelling interaction between the nodes of an MRTS-based network, as well as the trust decision making process for each node, which finally results in cooperating (trusted) or defecting (untrusted). From the security point of view, cooperating and defecting nodes correspond to the fact nodes execute the network's operation correctly or misbehave, respectively. In the context of this work, a misbehaving node is either a selfish node, an intruder that trigger attacks against the network, a node with not enough energy, or/and a node with not a good ETX. These pieces of information are abstract for mathematical modelling. Every two players engaged in the game (decision process to cooperate or defect -trust or untrust-) play simultaneous moves PD in every stage of the game. After every stage, the players reveal all information -trust values- about the previous stage. In the first stage, all players cooperate (trust), and then intruder nodes will defect (untrust) while normal nodes will choose either to cooperate or defect according to other players' moves in previous stages. We define MRTS as:

1. Cooperate on the first move;

[5]The PD is a non-cooperative game with imperfect information that can be applicable in many domains. The PD can be extended to a multi-player or a repeated game and it is the basis for many models used to analyse the performance of networks' routing protocols. For more details please refer to [48][49][50].

Table 3: Synthesis of security solutions for RPL

Works	Technique	Collaboration	Attacks	Disadvantages	Validation
[16]	IDS	No	Rank, sinkhole, DIS, and neighbour attacks	More overhead because of the amount of information needed by the cluster head. Less accuracy detection after 10 minutes of execution. Cannot deal with mobile nodes	Simulation (Cooja)
[22]	IDS	No	Sinkhole, Rank, selective forwarding	High false detection rate and the lack of DIO synchronisation. Cannot deal with mobile nodes	Simulation (Cooja)
[23]	IDS	Yes	Wormhole attack	Consumes energy. Cannot deal with mobile nodes	Simulation (Cooja)
[31]	Trust	No	Selective-forwarding attack	The solution does not deal with bad-mouthing and good-mouthing attacks because each node takes a decision based only on its own knowledge. If this node misbehaves, it will choose a failing path rather than a trusted one. Uses only one parameter (packet forwarding) to calculate trust	Simulation (J-Sim)
[32]	Trust	No	Network-level-attacks	Uses only direct trust. Vulnerable to bad-mouthing and good-mouthing attacks. Uses only one parameter (packet forwarding) to calculate trust. No specific attacks addressed	Simulation (MATLAB)
[33][34]	Trust	No	Rank, Sybil, Blackhole attacks	Uses only one parameter (packet forwarding) to calculate trust. Vulnerable to bad-mouthing and good-mouthing attacks	Simulation (Cooja)
[9]	Trust	Yes	Routing attacks	No simulation analysis to verify the effectiveness of the model against routing attacks. No specific attacks addressed	Simulation (Linux-C-Based)
[35]	Trust and IDS	Yes	Blackhole, Sybil and Rank attacks	The programmed code size does not fit into the objects' (Tmote Sky) available memory. Massive and complex computation with too much information to use and store. No details of the integration method of the model to RPL. No details of the use of the IDS with the model	Simulation (Cooja)
New MRTS	IDS and Trust	Yes	Blackhole and Rank attacks	Although the solution presents good performances regarding packet delivery, energy consumption and rank change, further investigation is needed to prove its effectiveness in big networks	Simulation (Cooja) and Mathematical analysis

2. In each period observe the past opponent's actions and count the number of defection (which corresponds to its trust evaluation): nbD;
3. If $nbD < Threshold$ Cooperate else Defect for the remainder of the game.

In section 6.2, we will give the equilibrium analysis of the proposed MRTS strategy and compare it with other known strategies. The equilibrium tells us about the most rational choice for each player in the game in a particular situation, and the network follows that by either isolating misbehaving nodes or not.

6.2. MRTS Strategy Analysis

Defection is the equilibrium in the one-shot PD game. Likewise, in a finite repeated PD, the only equilibrium is to defect, and it represents the Sub-game Perfect Equilibrium (SPE) . However, it is not the only equilibrium in an Infinitely repeated PD (IPD). Indeed, it is possible to have cooperation as an equilibrium because players can anticipate future rewards and punishments. It can be different equilibria in repeated games [49].

According to [48][49][50], each player i has a repeated game strategy $s_i = (s_i^0, s_i^1, ..., s_i^T)$, where each s_i^t is history-dependent, the game is repeated T periods (stages), and T can be infinite ($T = \infty$). Formally, we represent each MRTS strategy of a player i as a sequence of history-dependent stage-game strategies such that in equation 13; where, C: Cooperate, D: Defect, $(C,C)^t$: means (C, C) repeated t times, and $(D,C)^{nbD}$: means (D, C) repeated nbD times.

$$s_i^t(h^t) = \begin{cases} C & \text{if } t = 0 \text{ or } h^t = ((C,C)^t) \\ C & \text{if } h^{nbD} = ((D,C)^{nbD}) \text{ and } nbD < Threshold \\ D & \text{if } h^{nbD} = ((D,C)^{nbD}) \text{ and } nbD \geq Threshold \end{cases} \quad (13)$$

So, is it an equilibrium for two players to play MRTS for this iterated PD game? We will look to the game as two phases' game: The cooperation phase and the defection phase. In the cooperation phase, no one has defected previously, so both players are cooperating. In MRTS, the defection phase itself is divided into two sub-phases: the defection-cooperation phase and the defection-defection phase. In the defection-cooperation phase opponent defects either alternatively or continuously, while the player cooperates until the number of defection is equal or greater than a defined threshold. In the defection-defection phase, the number of opponent's defections exceeds the threshold and thus defection-defection is played forever. We will check if in any of these phases of the game a player will need to deviate from MRTS strategy, with the assumption that the other player also is adopting MRTS strategy. It is assumed that the environment for the repeated game is stationary [49], and thus the payoff matrix is the same in every period. In this analysis, to calculate repeated game payoffs, we use PD payoff matrix[6] from Table 4, and formulas from [49][7].

Table 4: Prisoner's Dilemma payoff matrix

Player 1 \ Player 2	Cooperate (C)	Defect (D)
Cooperate (C)	(R,R)	(S,T)
Defect (D)	(T,S)	(P,P)

6.2.1. Cooperation Equilibrium

For IPD, there are infinitely many equilibria, and it is possible to have an equilibrium in which both players always cooperate; this is what we will see in following. If both players cooperate on the first-period $t = 0$. Therefore, at period $t = 1$, the history is $h^1 = (C, C)$; so they both play cooperatively again. As consequence, at period $t = 2$, the history is $h^2 = ((C, C), (C, C))$; and so on which generates an infinite path of (C, C). Thus, assuming that cooperation is an equilibrium, we calculate the repeated game Equilibrium Payoff (EP) to each player as in equation 14, where δ[8] is the discount factor that takes values in [0,1].

$$EP = \sum_{t=0}^{n} \delta^t R = R + R\delta + R\delta^2 + R\delta^3 + ... + R\delta^n \quad (14)$$

Thus, the average equilibrium payoff is : $\overline{EP} = R$

The question is: can any player gain from deviating from cooperation given that other players are accurately following it? Several cases can occur: defecting k times from the first period, defecting k times from the x^{th} period, and defecting k times extended on several periods, where $k > 0$. It must be remembered that by following the MRTS strategy, if player 1 defects, player 2 will cooperate while the number of defection is less than the threshold. In other words, (MRTS, MRTS) strategy at period-(t+1) depends not only on what is played at period-t but also on previous plays.

Case 1)-1: If player 1 defects in this first period and continues defecting k times ($Threshold = k$), he/she will have a payoff of T for the first k defections and then a payoff of P for the remainder

[6] If both players cooperate they both get a cooperation reward (R). However, if only one cooperates then the cooperating player gets a sucker score (S) whereas the defecting player receives a selfish temptation salary (T). Finally, if they both defect they both get a selfish punishment (P).

[7] http://virtualperfection.com/gametheory/5.2.InfinitelyRepeatedGames.1.0.pdf

[8] The discount factor allows to bound the stage-game payoffs and thus allows the infinite sum of the weighted payoffs to be finite.

of the game. So the payoff for the defecting player 1 (Defection Payoff: DP) will be as in equation 15.

$$\begin{aligned} \mathrm{DP} &= \textstyle\sum_{t=0}^{k-1} \delta^t \mathrm{T} + \sum_{t=k}^{n} \delta^t \mathrm{P} \\ &= \mathrm{T} + \mathrm{T}\delta + ... + \mathrm{T}\delta^{k-1} + \mathrm{P}\delta^k + ... + \mathrm{P}\delta^n \end{aligned} \tag{15}$$

Thus, the average defection payoff is :
$\overline{DP} = T(1-\delta^k) + P\delta^k$.

A player will continue cooperating according to the MRTS strategy if the following condition holds: $\overline{EP} \geq \overline{DP}$ (i.e., $R \geq T(1-\delta^k) + P\delta^k$), and thus if inequality 16 holds.

$$\delta \geq \left(\frac{\mathrm{T}-\mathrm{R}}{\mathrm{T}-\mathrm{P}}\right)^{\frac{1}{k}} \tag{16}$$

Case 1)-2: If player 1 cooperates, and then defects in the x^{th} period, and continues defecting k times, he/she will have a payoff of R for the x first periods, a payoff of T for the k defection times, and then a payoff of P for the remainder of the game. So the payoff will be as in equation 17.

$$\begin{aligned} \mathrm{DP} &= \textstyle\sum_{t=0}^{x-1} \delta^t \mathrm{R} + \sum_{t=x}^{x+k-1} \delta^t \mathrm{T} + \sum_{t=x+k}^{n} \delta^t \mathrm{P} \\ &= \mathrm{R} + \mathrm{R}\delta + ... + \mathrm{R}\delta^{x-1} \\ &\quad + \mathrm{T}\delta^x + ... + \mathrm{T}\delta^{x+k-1} \\ &\quad + \mathrm{P}\delta^{x+k} + ... + \mathrm{P}\delta^n \end{aligned} \tag{17}$$

Thus, the average defection payoff is :
$\overline{DP} = R(1-\delta^x) + T(\delta^x - \delta^{x+k}) + P(\delta^{x+k} - \delta^n)$

Such as in case 1)-1, a player will continue cooperating according to MRTS strategy if $\overline{EP} \geq \overline{DP}$ (i.e., $R \geq \overline{DP} = R(1-\delta^x) + T(\delta^x - \delta^{x+k}) + P(\delta^{x+k} - \delta^n)$), and thus if inequality 16 holds.

We conclude that cooperation is an equilibrium (i.e., every player will be willing to cooperate forever) as long as the condition 16 holds. In other words, if the value of the discount factor δ is as in inequality 16, the deviation is not profitable (i.e. we mean by deviation k deviation times). Indeed, in the case of defection and for sufficiently patient players, there is a trade-off of getting a good payoff for k-defection-stages and then suffering for the rest of the time. As already stated in the literature, higher δ means more patience from a player, more care for the future, a higher chance for surviving into the next stage, and consequently enabling greater cooperation.

In this analysis, we do not present the case where the player defects k times extended on several periods. Nevertheless, we believe that the biggest gain a player can have by defecting is to play defection at the k first periods since the discount factor decreases more and more with time. In other words, the defection equilibrium payoff (DP) for any choice of defection periods is weakly less than the cooperation equilibrium payoff (CP) for $\delta \geq \left(\frac{T-R}{T-P}\right)^{\frac{1}{k}}$. As a result, the player willingness is to cooperate rather than defect. The simulation results in section 6.3 demonstrate that the cooperation is maintained regardless of defection positions (periods). What matters is the number of defections throughout the game.

6.2.2. Defection Equilibrium

Can any player gain from deviating from defection strategy, given that other players are accurately following it? Two cases could occur:

Case 2)-1: If *nbD* through h^t is greater than or equal to the threshold, then play (D, D). If both players arrive at a sub-game of mutual defection forever (D, D) (after "$k = threshold$" defection times), this sub-game consists of the IPD. Playing the stage-game Nash equilibrium (D, D) of a game that is being infinitely repeated (in this case, PD) is an equilibrium itself [49]. Thus, if the two players are defecting forever, the best response for both of them is to continue defecting forever, and no one will need to deviate.

Case 2)-2: If *nbD* through h^t is less than the threshold then play (D, C). According to MRTS, the punishment phase is reached when *nbD* equals the threshold and it corresponds to Case 2)-1. However, if player 2 deviates from playing C before reaching the threshold, he/she plays (D, D) and the Case 2)-1 applies, where both players play the equilibrium path (D, D) forever. So, it is the best response to him/her to deviate from this phase of the strategy. From another hand, if player 1 deviates from playing D, he/she plays (C, C), and thus both players play the equilibrium path (C, C) forever. Consequently, the best response to him/her is also to deviate from this phase of the strategy.

MRTS is a complex strategy which can be an SPE[9] or not. Under the condition of threshold equal to 1, it is equivalent to the Spiteful strategy, which makes it SPE. From the MRTS strategy point of view, the whole period where a player defects while the other player cooperates and the number of defection is less than a threshold is equivalent to the cooperation period (i.e., (C, C)).

[9] Sub-game Perfect Equilibrium [49].

Thus, the limit when *nbD* reaches the threshold corresponds to the one-shot deviation. In other words, the overall defection period can be reduced to the one-shot deviation, making it like Spiteful. Nevertheless, when playing (D, C) or (C, D), if there was a deviation and we enter the punishment phase (D, D) forever, no player will want to deviate again since this guarantees a minimum gain of P for both players. Likewise, if there was a deviation and we enter the cooperation phase (C, C) forever, no player will want to deviate again since this guarantees a gain of R for both players. Like the tit-for-tat strategy, these two beneficial deviations imply that MRTS is not SPE.

6.3. Simulation Results with Perfect Vs. Imperfect Monitoring

Axelrod [51][52][50] was the first to organise computer tournaments to numerically detect strategies that would favour cooperation among players in the iterated PD. To this end, authors used ecological evolution algorithm for finding the optimal and robust strategies. At the beginning of the execution, there exists the same number of population for each strategy. A round-robin tournament is executed inducing the population of bad strategies to decrease whereas good strategies obtain new players. Because game theory is developed based on the understanding that all involved players are rational, when a defecting player discovers that benefit of cooperating players is higher than defecting ones, it will change its strategy in order to get higher benefit. Therefore, the proportion of players with different strategy is changing with time. Thus, the process is iterated until the population does not change anymore. *"At the end, the good strategy is the one which stays alive in the population for the longest possible time, and in the biggest possible proportion"*.

In this section, we present a numerical analysis of the performance of the MRTS strategy in term of cooperation and cooperation evolution among nodes. We compare MRTS to other known strategies; always cooperate (all_c), always defect (all_d), tit-for-tat, Spiteful, and soft-major. In the tit-for-tat strategy, each player starts the game by cooperating, and for all the future stages, each player copies the opponent's move from the previous stage. In the Spiteful strategy, each player starts by cooperating and continues to cooperate as long as everyone has cooperated previously. In the soft-major strategy, each player cooperates, then plays the opponent's majority move, and if equal cooperate. Both all_c and all_d strategies are history-not-dependent. We consider two cases: perfect and imperfect monitoring and use for the simulation a software introduced in [53]. We implemented the MRTS strategy and added it to the list of strategies in [53]. As inputs, each strategy begins the simulation with a population of 100 players and competes in a round-robin tournament. Besides, we use the payoff matrix in Table 4, where $P = 1$, $T = 5$, $S = 0$, and $R = 3$.

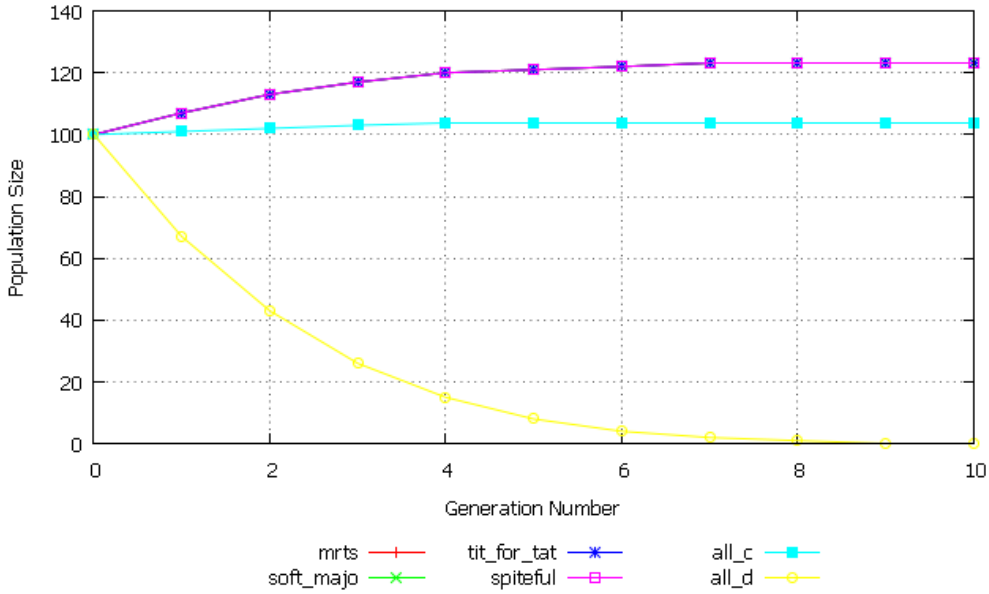
6.3.1. Perfect Monitoring

As depicted in Figure 6a, the MRTS strategy won the tournament equally with tit-for-tat, Spiteful, and soft-major. This achievement implies that MRTS is an Evolutional Stable Strategy (ESS) of the IPD game, and is equivalent to the three other strategies as an evolutionary strategy to favour and enforce cooperation among players. Figure 6b shows that MRTS is as good as other strategies to promote cooperation and cooperation evolution since it was ranked eighth out of 38 strategies involved in the simulation, with a size of 228 players. These results give the conclusion that the MRTS punishment strategy enforces the cooperation of participating nodes and prompts smart adversary nodes to become honest and cooperative.

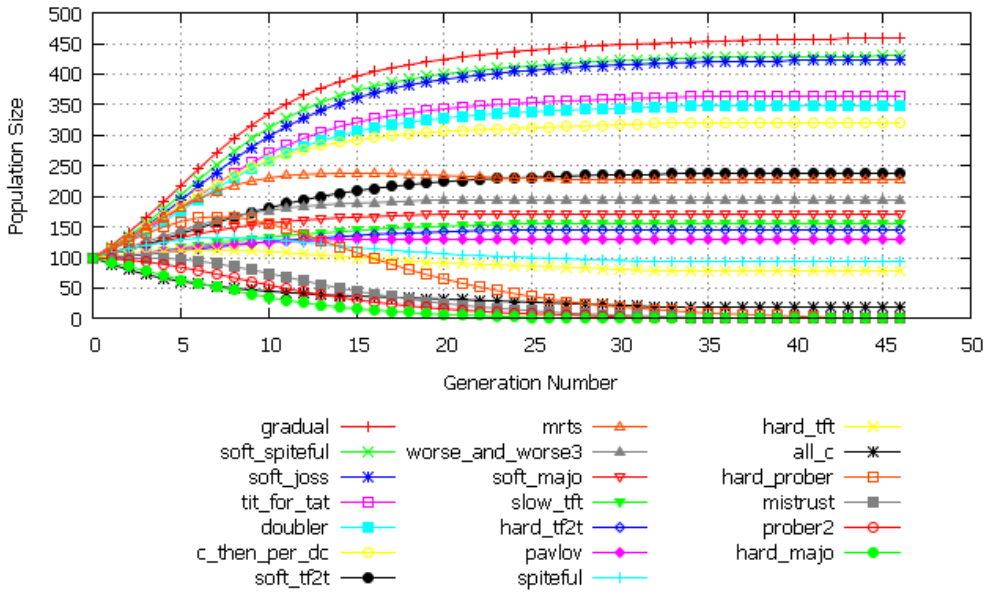
6.3.2. Imperfect Monitoring

It is unrealistic to model real-world scenarios with the assumption of a noise-free environment, for instance, in the case of errors due to IDS monitoring tools, link quality, and promiscuous mode. The software introduces noises when playing the game to simulate imperfect monitoring. When we make the imperfect monitoring assumption, different results appear. Indeed, for a defection threshold of 10, the MRTS strategy performs better than the other strategies when the noise percentage is in-between 4% and 10%. It is the most evolutional stable strategy (ESS) of the IPD game (See Figure 7a and Figure 7b). As depicted in Figure 8a, we notice that when the noise is significant (25%), the strategy soft-majo performs better than other strategies. Nevertheless, MRTS still performs better than Spiteful and tit-for-tat. As depicted in Figure 8b, when the noise is about 45% to 50% even if MRTS loses players, it stabilises after 90 generations with a population of 55 players, and thus performs better than other strategies such as tit-for-tat, which dies after 33 generations and Spiteful after 65 generations. However, when the noise exceeds 50%, the MRTS strategy disappears after 149 generations but still better than tit-for-tat, which disappears after 34 generations and Spiteful after 67 generations.

We can explain the results above as follow. By adopting the MRTS strategy, a node bases its decision of whether to trust (cooperate) or not (defect) relying



(a) 6 Strategies



(b) 38 Strategies

Figure 6: MRTS Compared to Different Strategies Under Perfect Monitoring.

both on observations that it made on its opponents' past moves and a predefined defection threshold. Thus, the trust measure evaluated by the node takes into account more than one observation. Thus, the behaviour of the MRTS strategy does not depend on the noise percentage. Instead, it depends on the misperception noise itself and the threshold. Two cases could occur: i) in the first case, the misperception noise is cooperation instead of defection, and the threshold is not reached so the cooperation phase is extended and thus MRTS performs better. ii) in the second case, the misperception noise is defection instead of cooperation, and the threshold is reached quickly, so the cooperation phase is shortened, and thus MRTS disappears faster. In other words, the MRTS strategy will maintain the equilibrium state of cooperation when there exist small defection deviations, and the threshold is not reached, or when case i) occurs.

7. Conclusion

In this paper, we presented MRTS: a cooperation-trust-based routing mechanism for RPL. According to MRTS, at each hop of an RPL routing path, the child node selects the node that has higher trust value, more energy and better link quality as its preferred parent. We proved with a simulation study that MRTS using multi-criteria based trust as routing metric (ERNT) is an efficient mechanism to reduce the network security risks and maintain its performance and stability. Indeed, results show that MRTS has less energy consumption and more packet delivery ratio due to both: its capacity to detect and isolate attacks and to its energy balanced topology mechanism. Furthermore, we demonstrated that ERNT fulfils the isotonic and monotonic properties, hence allowing MRTS-based routing protocol to satisfy the requirements of consistency, optimality and loop-freeness.

In this paper, we also translated MRTS into a strategy using game theory concepts. The MRTS strategy makes the malicious non-cooperative nodes be punished (by decreasing their trust values) and isolated, and thus enforces the security of the network by enforcing nodes to cooperate rather than cheating. We analysed the cooperation evolution of the MRTS strategy and demonstrated mathematically and with a simulation that nodes in the network could achieve effective cooperation and the MRTS strategy will become the stable strategy of the interactive nodes. Furthermore, the simulation study showed that the MRTS strategy is an evolutional stable strategy, and under perfect monitoring, it is equivalent to the tit-for-tat and the Spiteful strategies to favour and enforce cooperation among nodes.

As future work, we plan to experiment and analyse MRTS performances in a real testbed and for large scale networks. Furthermore, we will extend MRTS with more criteria such as mobility, and test its functionalities against different trust thresholds and other routing attacks.

8. Acknowledgements

This work is financially supported by the Research Center on Scientific and Technical Information (CERIST).

References

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645–1660.

[2] J. W. Hui, D. E. Culler, Extending ip to low-power, wireless personal area networks, IEEE Internet Computing (4) (2008) 37–45.

[3] M. Ammar, G. Russello, B. Crispo, Internet of things: A survey on the security of iot frameworks, Journal of Information Security and Applications 38 (2018) 8–27.

[4] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, P. Levis, Connecting low-power and lossy networks to the internet, IEEE Communications Magazine 49 (4) (2011).

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Rpl: Ipv6 routing protocol for low-power and lossy networks, RFC 6550, Internet Engineering Task Force (2012).

[6] S. H. Shah, I. Yaqoob, A survey: Internet of things (iot) technologies, applications and challenges, in: 2016 IEEE Smart Energy Grid Engineering (SEGE), IEEE, 2016, pp. 381–385.

[7] F. Medjek, D. Tandjaoui, I. Romdhani, N. Djedjig, Security threats in the internet of things: Rpl's attacks and countermeasures, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 147–178.

[8] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, M. Richardson, A security threat analysis for the routing protocol for low-power and lossy networks (rpls), Tech. rep. (2015).

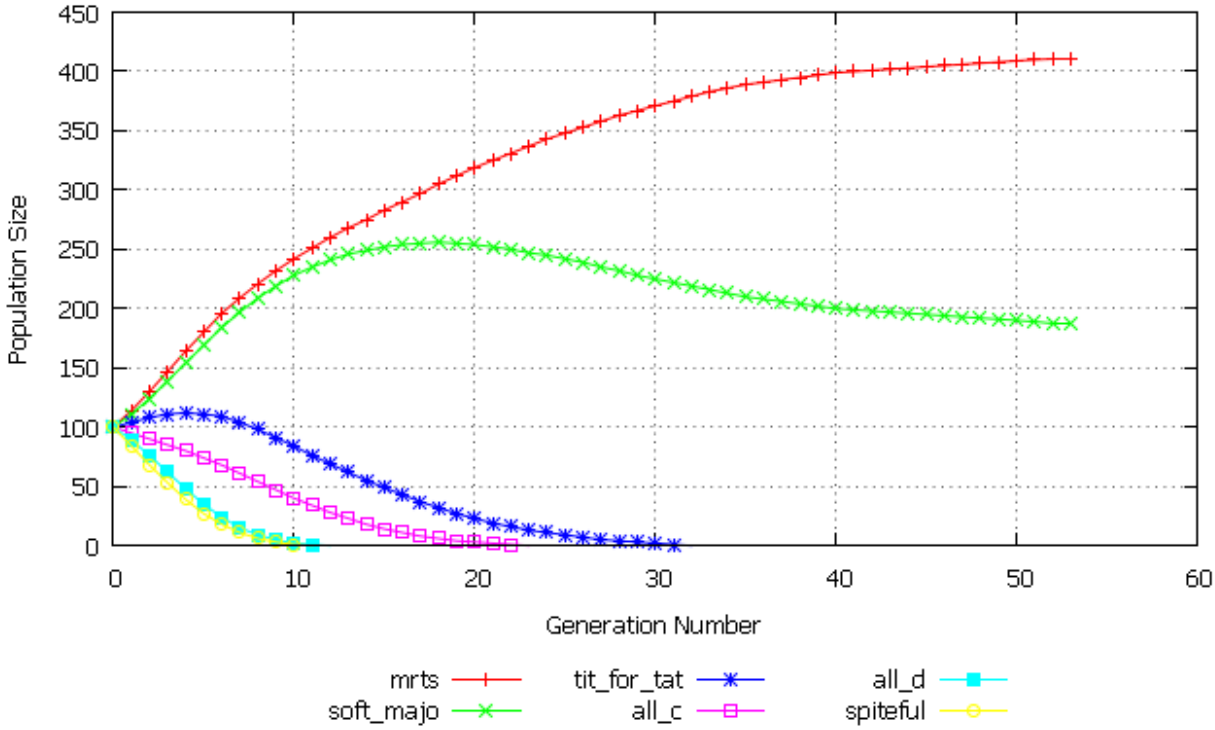
[9] N. Djedjig, D. Tandjaoui, F. Medjek, I. Romdhani, New trust metric for the rpl routing protocol, in: Information and Communication Systems (ICICS), 2017 8th International Conference on, IEEE, 2017, pp. 328–335.

[10] L. Buttyan, J.-P. Hubaux, Security and cooperation in wireless networks: thwarting malicious and selfish behavior in the age of ubiquitous computing, Cambridge University Press, 2007.

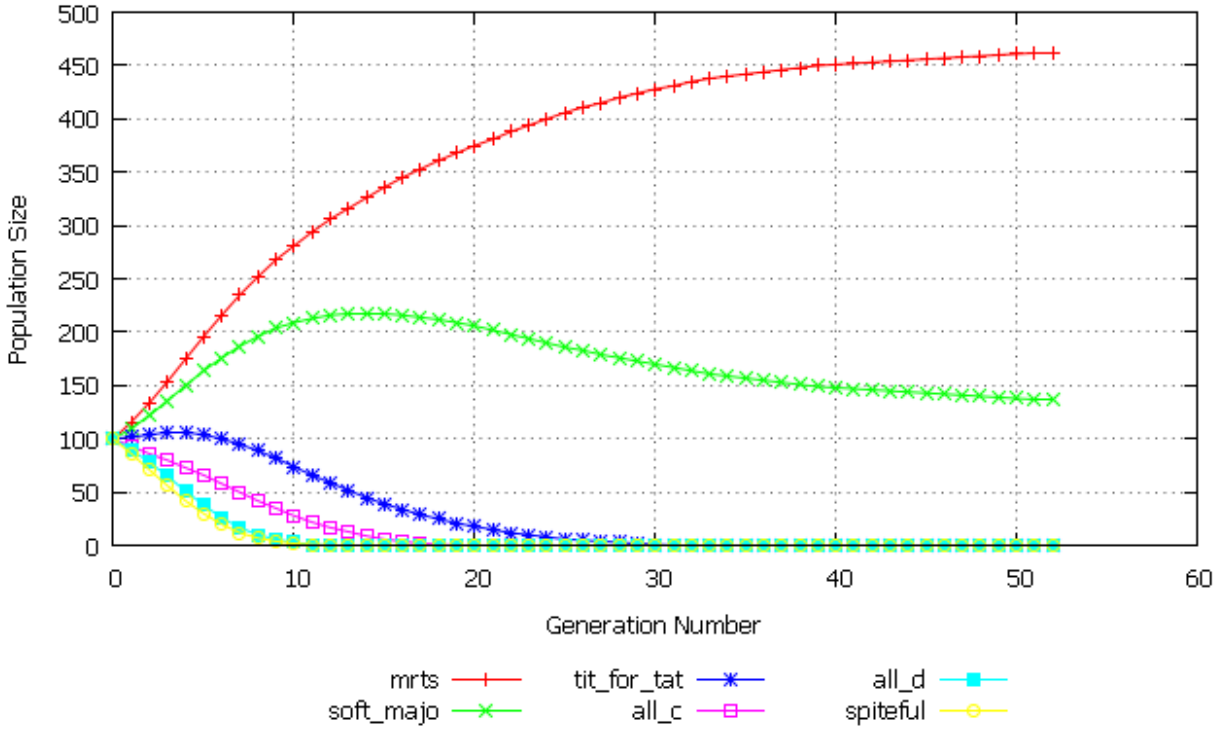
[11] N. Djedjig, D. Tandjaoui, F. Medjek, Trust-based rpl for the internet of things, in: 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2015, pp. 962–967.

[12] N. Djedjig, D. Tandjaoui, I. Romdhani, F. Medjek, Trust management in the internet of things, in: Security and Privacy in Smart Sensor Networks, IGI Global, 2018, pp. 122–146.

[13] P. Thubert, Objective function zero for the routing protocol for low-power and lossy networks (rpl), RFC 6552, Internet Engineering Task Force (2012).

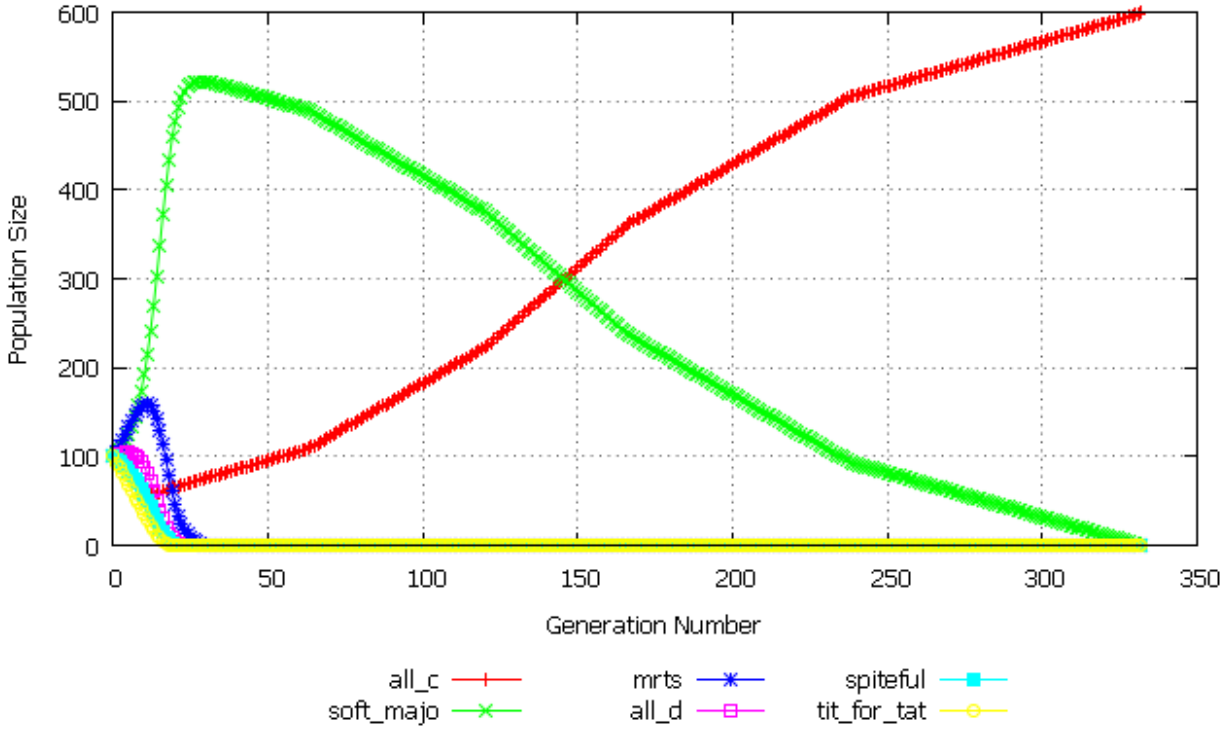


(a) 5% Noise

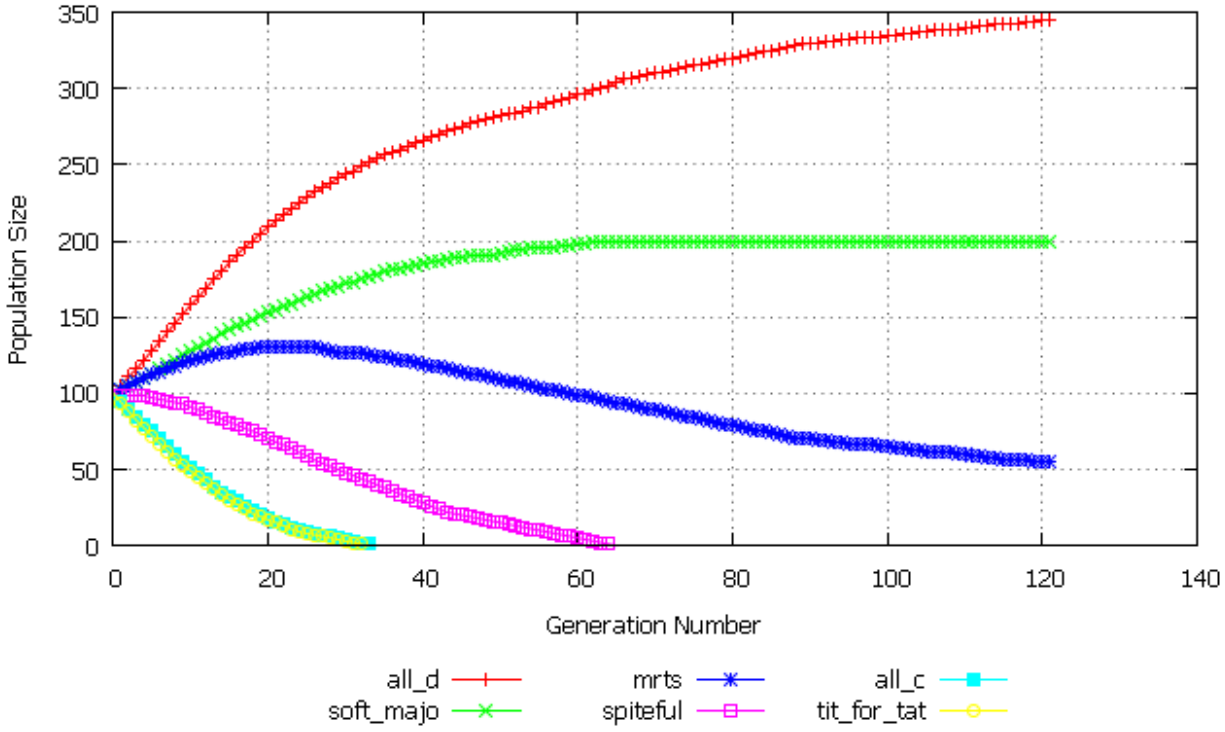


(b) 10% Noise

Figure 7: MRTS Under Imperfect Monitoring for 5% and 10% Noise.



(a) 25% Noise



(b) 50% Noise

Figure 8: MRTS Under Imperfect Monitoring for 25% and 50% Noise.

[14] J. Vasseur, M. Kim, K. Pister, N. Dejean, D. Barthel, Routing metrics used for path calculation in low power and lossy networks, RFC 6551, Internet Engineering Task Force (2012).

[15] A. Dvir, T. Holczer, L. Buttyan, Vera-version number and rank authentication in rpl, in: Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, IEEE, 2011, pp. 709–714.

[16] A. Le, J. Loo, K. Chai, M. Aiash, A specification-based ids for detecting attacks on rpl-based network topology, Information 7 (2) (2016) 25.

[17] A. Le, J. Loo, Y. Luo, A. Lasebae, The impacts of internal threats towards routing protocol for low power and lossy network performance, in: Computers and Communications (ISCC), 2013 IEEE Symposium on, IEEE, 2013, pp. 000789–000794.

[18] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in rpl-based internet of things, International Journal of Network Security 18 (3) (2016) 459–473.

[19] K. Chugh, L. Aboubaker, J. Loo, Case study of a black hole attack on lowpan-rpl, in: Proc. of the Sixth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Rome, Italy (August 2012), 2012, pp. 157–162.

[20] K. Weekly, K. Pister, Evaluating sinkhole defense techniques in rpl networks, in: 2012 20th IEEE International Conference on Network Protocols (ICNP), IEEE, 2012, pp. 1–6.

[21] Z. J. Haas, L. Yang, M.-L. Liu, Q. Li, F. Li, Current challenges and approaches in securing communications for sensors and actuators, in: The Art of Wireless Sensor Networks, Springer, 2014, pp. 569–608.

[22] S. Raza, L. Wallgren, T. Voigt, Svelte: Real-time intrusion detection in the internet of things, Ad hoc networks 11 (8) (2013) 2661–2674.

[23] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in internet of things, International Journal of Computer Applications 121 (9) (2015).

[24] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, I. Ahmedy, Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol, IEEE Access 6 (2018) 16623–16638.

[25] F. Y. Yavuz, D. Ünal, E. Gül, Deep learning for detection of routing attacks in the internet of things, International Journal of Computational Intelligence Systems 12 (1) (2018) 39–58.

[26] F. Bao, I.-R. Chen, Trust management for the internet of things and its application to service composition, in: IEEE WoWMoM 2012 Workshop on the Internet of Things: Smart Objects and Services, IEEE, 2012, pp. 1–6.

[27] R. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, IEEE transactions on dependable and secure computing 13 (6) (2015) 684–696.

[28] R. Chen, J. Guo, Hierarchical trust management of community of interest groups in mobile ad hoc networks, Ad Hoc Networks 33 (2015) 154–167.

[29] R. Chen, J. Guo, D.-C. Wang, J. J. Tsai, H. Al-Hamadi, I. You, Trust-based service management for mobile cloud iot systems, IEEE Transactions on Network and Service Management 16 (1) (2018) 246–263.

[30] F. Bao, R. Chen, M. Chang, J.-H. Cho, Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection, Network and Service Management, IEEE Transactions on 9 (2) (2012) 169–183.

[31] P. Karkazis, I. Papaefstathiou, L. Sarakis, T. Zahariadis, T.-H. Velivassaki, D. Bargiotas, Evaluation of rpl with a transmission count-efficient and trust-aware routing metric, in: Communications (ICC), 2014 IEEE International Conference on, IEEE, 2014, pp. 550–556.

[32] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, P. Herrmann, A trust-based resilient routing mechanism for the internet of things, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, 2017, p. 27.

[33] D. Airehrour, J. A. Gutierrez, S. K. Ray, Sectrust-rpl: A secure trust-aware rpl routing protocol for internet of things, Future Generation Computer Systems (2018).

[34] D. Airehrour, J. Gutierrez, S. K. Ray, Securing rpl routing protocol from blackhole attacks using a trust-based mechanism, in: 2016 26th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, 2016, pp. 115–120.

[35] S. Y. Hashemi, F. S. Aliee, Dynamic and comprehensive trust model for iot and its integration into rpl, The Journal of Supercomputing (2018) 1–30.

[36] A. Lahbib, K. Toumi, S. Elleuch, A. Laouiti, S. Martin, Link reliable and trust aware rpl routing protocol for internet of things, in: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), IEEE, 2017, pp. 1–5.

[37] V. Kiran, S. Rani, P. Singh, Trust based defence system for ddos attack detection in rpl over internet of things, INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY 18 (12) (2018) 239–245.

[38] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, ACM Transactions on Sensor Networks (TOSN) 8 (4) (2012) 34.

[39] N. Tsiftes, J. Eriksson, A. Dunkels, Low-power wireless ipv6 routing with contikirpl, in: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, 2010, pp. 406–407.

[40] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on wireless communications 1 (4) (2002) 660–670.

[41] S. Marti, T. J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, 2000, pp. 255–265.

[42] D. S. J. De Couto, High-throughput routing for multi-hop wireless networks, Ph.D. thesis, Massachusetts Institute of Technology (2004).

[43] Y. Yang, J. Wang, Design guidelines for routing metrics in multihop wireless networks, in: INFOCOM 2008. The 27th conference on computer communications. IEEE, IEEE, 2008, pp. 1615–1623.

[44] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, Journal of network and computer applications 42 (2014) 120–134.

[45] G. Theodorakopoulos, J. S. Baras, On trust models and trust evaluation metrics for ad-hoc networks, IEEE Journal on selected areas in Communications 24 (LCA-ARTICLE-2007-016) (2006) 318–328.

[46] C. Zhang, X. Zhu, Y. Song, Y. Fang, A formal study of trust-based routing in wireless ad hoc networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.

[47] X. Liu, Z. Sheng, C. Yin, F. Ali, D. Roggen, Performance analysis of routing protocol for low power and lossy networks (rpl) in large scale networks, IEEE Internet of Things Journal 4 (6) (2017) 2172–2185.

[48] G. J. Mailath, L. Samuelson, Repeated games and reputations: long-run relationships, Oxford university press, 2006.

[49] J. Ratliff, Game theory.
URL `http://virtualperfection.com/gametheory`

[50] R. Axelrod, et al., The evolution of strategies in the iterated prisoners dilemma, The dynamics of norms (1987) 1–16.

[51] R. Axelrod, W. D. Hamilton, The evolution of cooperation, science 211 (4489) (1981) 1390–1396.
[52] R. Axelrod, The evolution of cooperation (1985).
[53] Iterated prisoners dilemma simulation software.
URL http://www.lifl.fr/IPD