

Article

Enhancing Security and Trust in Internet of Things through Meshtastic Protocol Utilising Low-Range Technology

Fabrizio Messina [†], Corrado Santoro [†] and Federico Fausto Santoro ^{*,†}

Department of Mathematics and Informatics, University of Catania, 95124 Catania, Italy; fabrizio.messina@unict.it (F.M.); corrado.santoro@unict.it (C.S.)

* Correspondence: federico.santoro@unict.it

[†] These authors contributed equally to this work.

Abstract: The rapid proliferation of Internet of Things (IoT) devices has raised significant concerns regarding the trustworthiness of IoT devices, which is becoming a crucial aspect of our daily lives. In this paper, we deal with this important aspect by taking into account Meshtastic, a dynamic mesh networking protocol that offers robustness and adaptability, important characteristics for the dynamic and heterogeneous IoT environment. LoRaWAN (Low-Range Wide Area Network), a low-power, long-range wireless communication standard, introduces energy efficiency and extends the reach of IoT networks, enabling secure communication over extended distances. To improve the trustworthiness of IoT devices, we present an integrated approach that leverages the strengths of Meshtastic's dynamic mesh networking capabilities and LoRa's low-power, long-range communication, along with the integration of a reputation model specifically designed for IoT. We evaluated the performance of the proposed solution through several simulations and real-world experiments. The results show that the devices' measured values of trust reflect the real behaviour of the devices. These findings underscore the viability and applicability of the Meshtastic protocol utilising LoRa technology as a pivotal step towards establishing resilient and trustworthy IoT infrastructures in the face of evolving security challenges.

Keywords: Internet of Things; security; reputation; LoRa; Meshtastic; LoWPAN; simulation



Citation: Messina, F.; Santoro, C.; Santoro, F.F. Enhancing Security and Trust in Internet of Things through Meshtastic Protocol Utilising Low-Range Technology. *Electronics* **2024**, *13*, 1055. <https://doi.org/10.3390/electronics13061055>

Academic Editor: Hung-Yu Chien

Received: 14 January 2024

Revised: 3 March 2024

Accepted: 6 March 2024

Published: 12 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The advent of the Internet of Things (IoT) [1,2] has ushered in an era of unprecedented connectivity and inter-device communication, revolutionising industries ranging from healthcare to transportation. As IoT ecosystems continue to proliferate along with new communication protocols [3], the need for robust security and trust mechanisms becomes increasingly imperative. The vulnerabilities associated with the vast network of interconnected devices raise concerns about data integrity, privacy, and unauthorised access [4–7]. In particular, the exponential growth of IoT deployments has brought forth a multitude of security concerns, including the risk of eavesdropping, data tampering, and unauthorised access. Traditional security measures often fall short in dynamically evolving IoT environments, necessitating innovative solutions to fortify these networks.

Given the premises above, our research aims to establish a resilient framework for secure communication over extended distances within IoT deployments. In particular, we investigated the efficacy of the Meshtastic protocol in conjunction with the LoRaWAN communication technology (<https://loro-alliance.org/>) in securing and enhancing trust in IoT networks.

Meshtastic [8] is a dynamic mesh networking solution designed to provide adaptability to the dynamic nature of IoT deployments. The LoRaWAN [9,10] networking protocol allows IoT devices to operate in regional, national or global networks by satisfying basic Internet of Things requirements: bi-directional communication, end-to-end security, mobility, and localisation services.

By leveraging the strengths of Meshstastic's dynamic mesh networking capabilities and LoRa's low-power, long-range communication, we designed an integrated approach to mitigate common security threats prevalent in IoT environments. Indeed, Meshtastic's adaptability is complemented by LoRa's efficiency, offering a solution that not only enhances security but also addresses the power constraints often associated with IoT devices. In particular, we designed the integration of a specific reputation model within the Meshtastic protocol. The reputation model was originally developed by some of the authors of this manuscript for a generic set of IoT devices. We developed a particular, customised version to integrate the model itself into the Meshtastic protocol. The integration has been tested in a few experiments to obtain a set of experimental results that validate our approach. The integrated approach presented in this paper contributes to the ongoing discourse on securing IoT networks, providing a pathway for researchers, industry practitioners, and policymakers to fortify the foundation of trust in expanding the IoT landscape. The integrated approach of the Meshtastic protocol with LoRa technology in IoT environments offers a wide range of potential real-world applications due to its enhanced security, energy efficiency, and adaptability, such as Industrial IoT (IIoT), environmental monitoring, supply chain management, and disaster management. Overall, the integration of the Meshtastic protocol with our reputation model opens up a plethora of opportunities for enhancing security, reliability, and efficiency in various IoT applications across different sectors.

This manuscript is organised as follows: Section 2 discusses the main related works. Section 3 introduces details of the Meshtastic protocol, which was useful in enabling the integration of the reputation model. Section 4 explains the design of the original reputation model previously developed by some of the authors of this manuscript, while Section 5 explains the design of the specific reputation model integrated into the protocol. Section 6 provides details of the simulations performed to validate our approach and, finally, in Section 7, we draw our conclusions.

2. Related Works

In this section, we discuss a few related works whose focus is the design of a reputation system for IoT devices with particular reference to network protocols. First of all, we will deal with a few related works about trust and reputation for IoT. In the second part of this section, we explore some related studies focusing on the integration of trust and reputation models into network protocols. Since our research work is focused on the integration of a reputation protocol into the Meshtastic protocol, in the following part of this section, we conduct a brief comparison of our work with those research works regarding the integration of trust and reputation models into network protocols.

2.1. Trust and Reputation for IoT

Providing models and tools to measure trustworthiness in IoT is a well-known problem for which several authors have proposed several solutions at various levels [11,12]. For example, in [13], the authors take into account the various security challenges that come with IoT and CPS (Cyber Physical systems), such as the detection of malicious attacks in the context of direct communication using, for instance, the 6LoWPAN (IPv6 over low-power wireless personal area networks) protocol. The authors designed a trust and reputation TRM-based model for IoT to facilitate the cooperation between IoT devices in a network of IoT/CPS based on the behaviours of IoT devices. They measured the accuracy, robustness, and lightness of the proposed model using a wide set of simulations.

In a more recent work [14], the authors studied the problem of the heterogeneity of IoT vs the security requirement in the same environment. They designed a novel IoT trust and reputation model built over distributed PNNs (probabilistic neural networks) to classify trustworthy nodes from malicious ones. The model particularly addresses the cold start problem in IoT environments: the characteristics of newly joined devices are analysed to give a starting trust value of reputation, and then, their behaviour is analysed over time. The processing is completely distributed along the nodes of the network to provide better

availability. The authors also illustrate that the various capabilities and types of IoT devices are supported by the model itself by providing different levels of security depending on the sensitivity of the data being transmitted.

In [12], the authors discuss the typical problems related to the integration of the existing trust and reputation models into the IoT environment: the large number of physical entities, the limited computation ability of physical entities, and the highly dynamic nature of the network. To address the limitations above, they present the design of IoTrust—a trust architecture that integrates SDN (Soft Defined Network) into IoT—and a cross-layer authorisation protocol based on IoTrust. They also provide—to address the problem of trust establishment by a particular model—a Behaviour-based Reputation Evaluation Scheme (BES) for the node and an Organisation Reputation Evaluation Scheme (ORES). Finally, they present a theoretical analysis and a few simulation results to validate the efficiency of BES and ORES.

In [15], the authors discuss the various security and privacy issues related to machine-to-machine interactions (M2M) and H2M (human-to-machine) interactions. They consider factors like the trustworthiness of a user or the detection of a malicious user in the particular context of fog/edge computing. They propose a context-aware multi-user trust evaluation model to evaluate the trustworthiness of a user in a Fog-based IoT (FIoT). In particular, they propose context-aware feedback and feedback crawler systems to make trust evaluation unbiased, effective, and reliable. They also introduce monitor mode for malicious/untrustworthy users to effectively monitor the behaviour and trustworthiness of a user. In the design of their approach, they provide several tunable factors, which can be tuned based on the system's requirements, and discuss a few simulation results to validate their approach. A few aspects of their work are interesting, such as the contextualisation of the problem of trust in edge/fog computing to support IoT devices. Nevertheless, in our work, we consider trust in the context of M2M interactions.

2.2. Integration of Trust and Reputation Models into Network Protocols

Mesh networks represent a recent development for IoT devices, and there is not much work on the integration of reputation models inside the mesh protocol. Some authors worked on the problem of self-sovereign identity management on Internet of Things mesh networks [16]. They used LoRaWAN (Low-Range Wide Area Network) as an example and applied Sovrin's decentralised identifiers and verifiable credentials in combination with Schnorr signatures for secure communications. They focused on simplex and broadcast connections. In detail, they also discussed an ESP32-based implementation using SX1276/SX1278 LoRa chips and adaptations made to the lmick- and MbedTLS-based software stack, and practically evaluated performance aspects (overhead, time-on-air impact, and power consumption). The research work discussed in this paragraph is slightly different from our work because the authors used LoRaWAN as an example and applied Sovrin's decentralised identifiers and verifiable credentials in combination with Schnorr signatures for secure communications. On the other hand, the authors have tested their solution for self-sovereign identity management in a real test-bed, which represents our objective for future research.

In [17], the authors study the problem of intrusion prevention techniques in the context of WMNs (wireless mesh networks); indeed, in this case, it is especially challenging and requires particular design considerations. To address this problem, they propose a particular anomaly detection scheme (RADAR) to detect anomalous mesh nodes in WMNs. First of all, they provide a first definition of reputation to quantify the mesh node's behaviour/status and, therefore, to define fine-grained performance metrics. In this way, they define a robust baseline for leveraging and measuring the derivation between normal and anomalous behaviour of mesh nodes. They also propose a cooperative anomaly detection scheme (based on the measurement of reputation) based on the full exploration of the spatio-temporal properties of mesh nodes' behaviour. They provide an implementation based on a reactive routing protocol, aimed at detecting malicious mesh nodes which intentionally violate

normal routing mechanisms. Finally, they provide a few simulation results to validate their approach in terms of detection accuracy, false positive rate, computational overhead, and scalability. The interesting part of the protocol developed by the authors is represented, in our opinion, by the reactive routing protocol, which represents an effective countermeasure to isolate anomalous nodes of the mesh network. On the other hand, our research work is different because we introduce a very simple reputation protocol as a payload of the Meshtastic protocol.

In [18], the authors take into account the model of the hybrid wireless mesh network, which has the advantage of a flexible infrastructure but is vulnerable to attacks. They state that traditional reputation schemes are effective in addressing security, but they are not suitable for direct application to hybrid wireless mesh networks, because node cooperation and hierarchical construction are not taken into account. Therefore, they propose a dynamic hierarchical reputation evaluation scheme (DHRES) to provide security for hybrid wireless mesh networks: the scheme includes a virtual cluster structure to introduce the reputation relation, including the related nodes' roles and functions. The reputation evaluation mechanism is based on the correlation between nodes, and each node's reputation is updated according to the different roles of the mesh nodes. They provide a few simulation results to compare their approach with traditional reputation models: the results highlight that the proposed scheme can accurately reflect the security status of nodes, particularly of malicious nodes. Our approach is different as we don't introduce any hierarchical protocol for reputation, with obvious advantages related to flat P2P management of the information about reputation.

The work described in [19] is based on the context of a wireless mesh network and a multi-path routing protocol. The authors state that, although the multi-path routing protocol provides many benefits such as fault tolerance and improved security, high bandwidth, and better load balance, the issue of cooperation among nodes in a wireless mesh network that uses a multi-path routing protocol, at the time of writing, has not been well addressed. Then, they propose a reputation-based system for the wireless mesh network using a multi-path routing protocol that promotes cooperation such that each node in different paths is "stimulated" to forward packets from others. The reputation system can detect misbehaving nodes with a reputation value under a certain threshold. The experiments highlighted that, in this way, the reputation system encourages the misbehaving nodes to behave honestly. These experiments show that the reputation system designed by the authors is effective in detecting misbehaving nodes.

Our research work is different because we integrate a reputation protocol in the Meshtastic protocol by using LoRa technology. The main advantage is the energy efficiency and the secure communications over extended distances.

3. The Meshtastic Protocol

The Meshtastic protocol [8] is a robust and efficient routing protocol specifically designed for mesh networks, with a particular focus on LoRa-based wireless communication. This protocol provides a simple yet effective mechanism for facilitating communications within a mesh network. It is, therefore, well-suited for applications requiring low-power, long-range wireless connectivity.

The Meshtastic protocol is heavily influenced by the mesh routing algorithm used in RadioHead [20], a popular open-source Arduino library for various wireless modules. RadioHead operates on the LoRa physical layer, which enables long-range communication while consuming minimal power. The protocol's design is simple and aimed at minimising power consumption, which is crucial for battery-operated devices in remote or resource-constrained environments. The protocol enables the creation of self-organising mesh networks, where each device can receive and transmit messages from other network participants. Additionally, each device can act as a repeater, extending the network's diameter. This inherent capability for message relaying is a key feature of the protocol,

enabling the establishment of robust and expansive mesh networks without the need for a centralised infrastructure.

The Meshtastic protocol is structured into four conceptual layers, each serving a specific function in the routing process (Figure 1):

- **Unreliable Zero-Hop Messaging**

This layer is represented by a conventional non-reliable LoRa packet transmission. It provides the most basic form of message delivery within the network. In addition to the converted packet bytes, there is also a preamble sent at the start of any data packet. This preamble allows receiving radios to synchronise clocks and start framing. The protocol uses a preamble length of 16, which is longer than the minimum preamble length of 8. After the preamble comes the LoRa Physical Header, which contains information about the packet length as well as a sync word to distinguish networks (in this case, it is set to *0x2B*).

- **Reliable Single-Hop Messaging**

Building upon the first layer, this layer ensures reliable LoRa packet transmission within a single hop, by extending the messaging provided by layer1 with the *WantAck* flag in the MeshPacket protobuf. This flag holds special significance in the context of broadcast messages: in this special case, the standard ACK behaviour is intentionally restrained to avoid the saturation of the channel with ACK messages. Instead, the initial sender monitors the channel to ascertain whether at least one node is rebroadcasting the packet, adhering to a naive flooding algorithm.

- **Reliable Multi-Hop Messaging**

This layer enables the transmission of LoRa packets across multiple network nodes by extending the reliability to multi-hop scenarios. The protocol is mainly designed around flooding to meet our specific use case. Nevertheless, the current implementation does not incorporate optimisation strategies for flooding, except for discontinuing re-transmission once a node observes a nearby receiver acknowledging the flooded packet. Consequently, in an N-node mesh, there is potential for up to N re-transmissions of a packet.

If any node within the mesh perceives a packet with a HopLimit other than zero, it will decrement that HopLimit counter and endeavour to re-transmit the packet on behalf of the original sender. To encourage distant nodes to flood the message, thereby ensuring its broader dissemination, the contention window (refer to Layer 1) for a flooding message is contingent on the signal-to-noise ratio (SNR) of the received packet. To obtain a low SNR (signal-to-noise ratio), the contention window size is small, enhancing the likelihood that nodes farther away will flood first, while closer nodes receiving this information will refrain from flooding.

- **Mesh Broadcast Algorithm**

This layer is responsible for propagating messages through the mesh network; it plays a pivotal role in ensuring that messages reach their intended destinations, leveraging the collective capabilities of the network's nodes.

The Meshtastic protocol handles network partitioning through its inherent mesh networking capabilities and the utilisation of a mesh broadcast algorithm. When a network partition occurs, the protocol's design allows for the continued operation of the network on both sides of the partition, enabling the independent evolution of each partition. This approach is known as *split-brain*, where each side of the partition can operate autonomously, with both sides potentially perceiving the other as having failed. The mesh broadcast algorithm, a fundamental component of the Meshtastic protocol, facilitates the propagation of messages through the mesh network. In the event of a network partition, this algorithm allows for the continued dissemination of messages within each partition, enabling the independent evolution of the network on both sides of the partition. This capability is essential for maintaining communication and data exchange within the network, even in the presence of network partitioning.

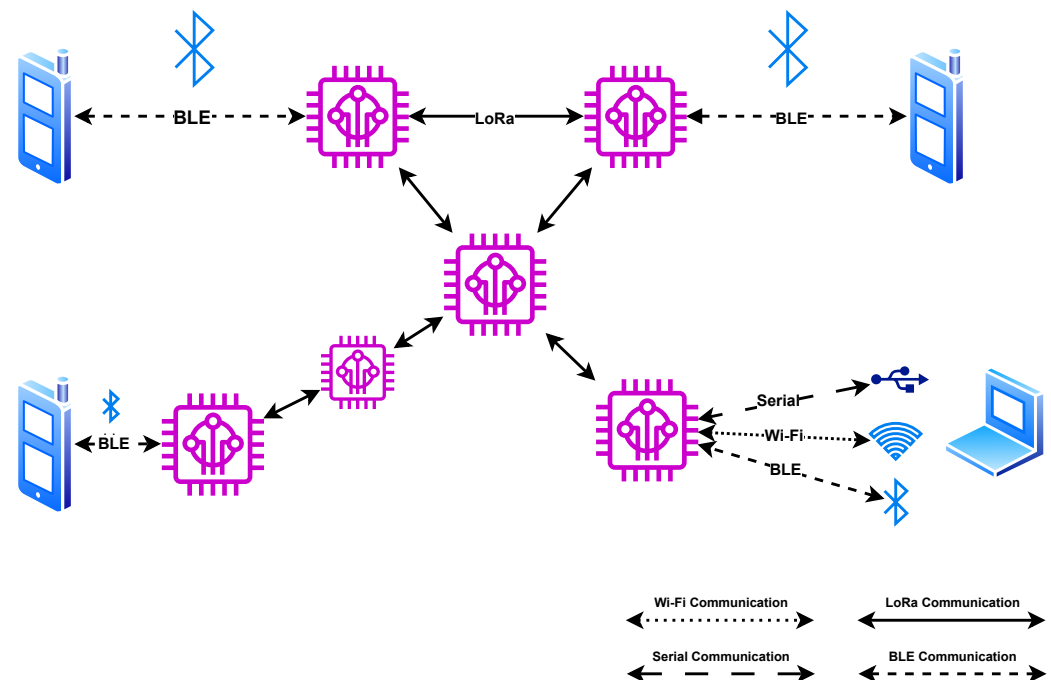


Figure 1. Meshtastic protocol scheme.

The Meshtastic protocol incorporates Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) for channel activity detection, a mechanism commonly used in WiFi. This approach requires all transmitters to perform Channel Activity Detection (CAD) before attempting to transmit, ensuring efficient channel utilisation and minimising the risk of data collisions. Furthermore, the protocol's use of Protocol Buffers ensures cross-vendor and cross-implementation compatibility, allowing devices from various manufacturers to seamlessly interoperate within the network. Additionally, the protocol integrates encryption mechanisms to secure the transmitted data, safeguarding the privacy and integrity of the communication within the mesh network. The Meshtastic protocol is designed to be adaptable to various applications and use cases, particularly in the context of IoT and mesh networking. Its efficient and reliable routing mechanisms make it well-suited for scenarios requiring long-range, low-power wireless communication, such as environmental monitoring, asset tracking, and remote sensor networks.

Although Meshtastic includes robust mesh networking capabilities, it still faces challenges related to trustworthiness. Indeed, in the context of wireless mesh networks, the establishment of trust between network participants is crucial for ensuring secure and reliable communication. Several research works and discussions shed light on the significance of trust and reputation models in mitigating the effects of attacks and evaluating the trustworthiness of network entities. For instance, in [21], the authors present a study related to protocols in wireless mesh networks. They emphasise the vulnerability of IoT settings to various types of attacks, which can compromise the authenticity, integrity, and real-time delivery of data. To address these challenges, the implementation of trust and reputation models is highlighted as a means to evaluate the trustworthiness of different players in IoT settings. Additionally, discussions within the Meshtastic community highlight the potential risks and challenges related to protocol validation, circular routing, and unintended denial-of-service attacks. These discussions underscore the importance of trustworthiness and, as a consequence, the proper processing of network packets to avoid potential disruptions and security vulnerabilities. For this reason, in the next section, we present a reputation model based on reliability and reputation to address these challenges and ensure the trustworthiness and security of communication within the network with the Meshtastic protocol.

4. Reputation Model

In this work, we take into account the reputation model discussed in [22], which introduces a comprehensive approach to evaluating the trustworthiness of Smart Objects (SOs) in a Social Internet of Things (SIoT) environment. The scenario described by the authors includes various edge/fog domains [23,24], where SOs will engage in activities involving the exchange of resources, such as offering or purchasing, with other SOs within the same edge domain.

The reputation model is designed to calculate a reputation score for all the SOs. The reputation score represents the history of the SO's behaviour concerning the honesty and quality of its interactions. Such a score is based on the previous behaviours of the counterparts within their interactions. Trust computation is based on feedback, a real number ranging from 0 to 1, which is used to update the reputation scores of the SOs, where 0 indicates the minimum appreciation and 1 indicates the maximum appreciation of the interaction. After each transaction, feedback is provided by each SO to express their level of satisfaction with the resource that was exchanged and the conduct of the other party involved. The trust score calculation takes into account not only the economic importance of the resource but also factors such as the frequency of interactions and the reputation of the party placing trust.

In the following section, we provide a few details related to the reputation model introduced in the previous paragraphs; as we explain later in this work, we integrated the reputation model within the Meshtastic protocol.

4.1. Smart Objects' Activities

These interactions are mainly determined by their group membership, aiming to enhance the likelihood of satisfactory exchanges by being part of a well-performing group. It is assumed that every Smart Object is assisted by a proper software agent.

An (i) **activation** task is carried out the first time that an SO joins the framework. To this end, let SO be an IoT device (and an associated trust agent t) and let E and e be an edge/fog domain active in the framework and its associated software agent, respectively. To start the activation process, an SO affiliation request is sent to e , after which the SO's agent t will become active and the domain agent e will provide it with the following:

- A **reputation score**, set to **0.5 by default**, is utilised to initially associate the SO with a group that is active within that specific domain;
- A **unique set** of cryptographic keys, with one key being public and the other private;
- The public key of e ;
- A **digital certificate**, endorsed by entity e , confirming the SO's identity and group affiliation, and including a timestamp indicating the certificate's expiry date.

An SO interested in a resource r can perform a simple (ii) **resource search** task using a simple message exchange. In more detail, the trust agent t , hosted from its associated SO, that is interested in r (i.e., the consumer) sends to agents belonging to the same edge domain a message consisting of

- The **membership** of the consumer group;
- A description of the resource being searched.

All SOs capable of fulfilling this request can respond by sending a signed message containing the provider group membership and a description of the offered resource, including its price.

(iii) **Resource provisioning** is performed whenever a potential consumer has received one or more messages from providers in response to their request. Let us denote the potential consumer as SO_i , and the potential provider of the service as SO_j . Let us denote as i and j their respective trust agents. When SO_i selects the resource offer from SO_j , then t_j and t_i will proceed as follows:

- Each of them signs their certificate and mutually exchanges these certificates;

- A negotiation stage is possibly carried out;
- Finally, they perform the resource delivery and payment steps.

The (iv) **trust management** task is the last step, after the resource provisioning. The SOs involved in the resource provisioning send a message to the agent **e**. The payload of the message is represented by the feedback indicating the “personal” satisfaction regarding the counterparts, and the signed certificate of the counterpart. The sender agent’s private key is used to sign this message. The main operation performed by the agent **e** after the message is received is represented by an update the reputation scores of the involved SOs (the provider and the consumer).

4.2. The Reputation Model

The basic information of the reputation model is represented by the feedback: once two SOs (SO_i and SO_j) have interacted for a resource **r**, then each SO releases feedback in the range of $[0, 1] \subset \mathbb{R}$ to the edge agent **e**. Then, let **r** be a resource, let $F_{(i,j)}$ be the feedback computed by SO_i about SO_j , and let $F_{(j,i)}$ be the feedback computed by SO_j about SO_i . Then, the trust agents t_i and t_j will send the respective $F_{(i,j)}$ and $F_{(j,i)}$ to the edge agent **e** to update the reputation scores of the two agents.

The feedback is combined in a parameter ϑ called “Interaction”, which also takes into account the economic relevance (δ) of the resource **r**, the frequency (ϕ) of the interactions that occurred between two SOs, and a parameter (ξ) related to the trustor’s reputation. Therefore, the effect of an interaction that occurred between SO_i and SO_j for a given resource **r** is represented by the following formula (1):

$$\vartheta_{i,j} = f(\delta_{i,j}, \phi_{i,j}) \cdot \xi_j \cdot F_{j,i} \quad (1)$$

where the economic relevance (δ) is defined by

$$\delta = \begin{cases} 0, & \text{if } p = 0 \\ \frac{p}{P_{\text{Max}}}, & \text{if } 0 < p \leq P_{\text{max}} \\ 1, & \text{if } p > P_{\text{max}}. \end{cases} \quad (2)$$

where p is the price of the resource and P_{max} is the price threshold. The frequency of interactions (ϕ) was designed to limit collusive activities carried out by two or more SOs to increase their reputations by mutually providing positive feedback.

$$\phi = \begin{cases} 1, & F < 0.5 \\ \mu^{-1}, & F \geq 0.5 \end{cases} \quad (3)$$

The value of ϕ will be set to 1 in the presence of negative feedback ($F < 0.5$); conversely, it will be computed as the inverse of the parameter μ , which, in turn, is defined by the time interval with which the SOs mutually provide feedback about each other (preliminarily set to 1 and $\mu \in [0, 1] \subset \mathbb{R}$).

$$\mu = \begin{cases} \max(1, \mu^{\text{old}}), & \Delta t < T \\ \max(1, \mu^{\text{old}} - \frac{\Delta t}{T}), & \Delta t \geq T. \end{cases} \quad (4)$$

In the above equation, **T** is a time threshold defined at the application level.

The function $f(\delta, \phi)$ returns a value in the domain $[0, 1] \subset \mathbb{R}$ and is designed to limit potential collusive and alternate activities between two SOs by penalising all iterations

characterised by a low or null relevance of the resource and a high value of the frequency parameter.

$$f(\delta, \phi) = \begin{cases} \sqrt{\delta^z + \phi^z}, & \text{if } f(\delta, \phi) \leq 1 \\ 1, & \text{otherwise} \end{cases} \quad (5)$$

The parameter z defines the behaviour of $f(\delta, \phi)$, and its value for best performance was obtained for $z = 1.35$.

The value of the parameter ξ , that is, the trustor's relevance, is computed based on the trustor's reputation using a step function ruled by the system threshold $\sigma \in [0, 1] \subset \mathbb{R}$ and is a parameter of the reputation model.

$$\xi = \begin{cases} 1, & \text{if } R \geq \sigma \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The feedback F is released by the trustor about the trustee in the domain $[0, 1] \subset \mathbb{R}$ based on its satisfaction degree concerning both the SO's honesty and the quality of the interaction that occurred, which, in turn, also depends on the resource r .

Once that parameter $\vartheta_{i,j}$ for the resource r has been computed, the reputation of SO_i and SO_j is updated, but only when the condition $\vartheta > 0 \vee R^{new} \geq 0.5$ is satisfied.

$$R_i^{new} = \begin{cases} \alpha \cdot R_i^{new} + (1 - \alpha) \cdot \vartheta_{i,j}, & \vartheta_{i,j} > 0 \vee R_i^{new} \geq 0.5 \\ R_i^{new}, & \text{otherwise} \end{cases} \quad (7)$$

In the above equation, $\alpha \in [0, 1] \subset \mathbb{R}$ is a parameter weighting the current reputation score R_i^{new} against the contribution $\vartheta_{i,j}$.

4.3. Dynamicity of IoT Environments

To handle the dynamicity of the IoT environment, a few measures can be adopted. The information about the reputation of the several IoT agents can be easily maintained by a *domain agent*. Each domain agent can, in turn, send that information to a *Cloud agent*; in this way, any domain agent can obtain updated information about the reputation of a single IoT agent by asking for the information itself from the Cloud agent. The simple measure above allows for the management of the dynamicity of IoT environments (migration of IoT devices), as the environments can be divided into domains managed by the domain agents and with the collaboration of a Cloud agent.

5. Integration of Reputation on Meshtastic

In this section, we explain how we designed the integration of the reputation model described in Section 4 within the Meshtastic protocol. Indeed, as explained in the introductory section, Meshtastic does not define a real approach to node trustworthiness as it only defines the mesh network routing protocol. Moreover, we found a few characteristics in the original reputation model that allowed us to effectively integrate trust mechanisms into the Meshtastic protocol.

In the Meshtastic context, we will use the term *nodes* when referring to Smart Objects (SOs), and the term *Mesh Packets* when referring to messages. In addition, we have made a few changes to the reputation model; such changes were necessary to integrate the model itself into the protocol.

First of all, the original model was designed to calculate a reputation score based on the interactions between SOs. In a Meshtastic network, such interactions are equivalent to message exchanges between two or more nodes. Moreover, the calculation of the reputation score is based on a few parameters, such as the frequency of interactions, the reputation of the two agents, and the economic relevance of a given resource, which, in our case, changes direction.

Moreover, each node participating in the network must hold the reputation scores of the other nodes. Still, the amount of information held by each node is, in principle, limited to the amount of memory available in the single device. This will be updated according to feedback from the other nodes of the network, as well as with the user's own opinion. The other nodes will send this feedback after a fixed time interval after an interaction with a specific node and it will be used to update the reputation of the same node. As we explain later in this section, these reputations will be sent to all nodes in the network during the flooding period of the Meshtastic protocol.

In the following section, we explain the details of the integration of the reputation protocol within the Meshtastic protocol.

5.1. Nodes' Activities

The **activation task** is performed as soon as the node successfully connects to the Meshtastic network. Here, the Edge Domain is considered the same Meshtastic protocol. A new packet, *Mesh Packet Affiliation*, is sent by the node administrator of the network (at that time) to the node joining the network. Once it has received the package, the new node becomes active with the following characteristics:

- The **reputation score**, set to **0.5 by default**;
- The network **symmetric** AES256 key (the Meshtastic Channel key);
- The network public and private keys (network-wide encryption keys);
- The ID of the node in the network.

A node that wants to perform a specific **resource search**—for instance, obtaining data from a specific sensor—sends a message to its neighbours, indicating the following:

- The type of resource required;
- The time interval required to send the resource.

Any node can respond to the request, using the same Meshtastic keys to sign the response message. The price of the resource is calculated based on the number of **HOPs** measured from the potential provider to the requesting node. Indeed, in a mesh network, the path length needed by a message to reach the destination represents a critical aspect in terms of energy consumption.

A **resource provisioning** task is then performed after the node has selected its provider from all nodes that have sent a response. Let us suppose node **A** (the consumer) has accepted to receive a service from node **B** (the provider). Then, the trust agents of the two nodes will perform the following activities:

- The two nodes, through their trust agent, will verify the response message with their keys;
- The consumer will start the negotiation for the requested resource;
- The provider will start the transmission of the requested resource, considering the time interval proposed to the consumer and the number of hops towards the consumer (the price of the resource).

Trust Management. Depending on the time interval set by the mesh network, each node of the network will send a Mesh Packet via flooding including its feedback on the resource requested to a given node and its behaviour, a message encrypted using the private network key. When a node receives this message type, it achieves the following:

- It certifies that the message is lawful to its network;
- It updates the reference node's reputation score.

5.2. The Reputation Model

Let us assume that we have two nodes, *A* and *B*, in the Meshtastic network, that have interacted for a certain resource *r*; then, each node will give feedback to all the other nodes of the network. Then, let $F_{(a,b)}$ and $F_{(b,a)}$ be the feedback generated by *A* against *B* and vice

versa. First of all, the two feedbacks will be flooded to all the nodes of the network, so they will use this information to update the scores of nodes A and B .

The parameter ϑ (interaction), in our implementation, takes into account the feedback F , the economic relevance δ of the resource, the frequency ϕ of interaction between the two nodes, and a parameter ξ describing the trustor's reputation, as in the original definition described in Section 4:

$$\vartheta_{a,b} = f(\delta_{a,b}, \phi_{a,b}) \cdot \xi_b \cdot F_{b,a} \quad (8)$$

and the economic relevance (δ), in this specific context of Meshtastic, is defined by

$$\delta_{a,b} = \begin{cases} 0, & \text{if } Hops_{(a,b)} = 0 \\ \frac{Hops_{(a,b)}}{Hops_{Max}}, & \text{if } 0 < Hops_{(b,a)} \leq Hops_{Max} \end{cases} \quad (9)$$

where $Hops_{Max}$ is represented by the maximum number of jumps allowed in the Meshtastic network in which the two nodes A and B participate. The protocol provides three-layer modes to exchange packages, with the first two including zero-hop messaging and the third one flooding multi-hop messaging. The latter does not aim to optimise flooding (in fact, in a network of N nodes, N could be the re-transmission of the same message in the network) and a header is inserted in each package called $Hops_{Limit}$ that is decremented with each arrival in an intermediate node. When the number reaches the value 0, then the package is discarded. As a consequence, our choice was to define the price as dependent on the Meshtastic layer mode, as defined by Equation (9).

We left the frequency of interactions $\phi_{a,b}$ the same as defined in the original model and as defined by Equation(3), unlike the μ parameter, where the Threshold T parameter is not defined at the application level but by the node itself during the *resource search* activity.

Finally, we left the function $f(\delta, \phi)$ as described in Equation (5); the parameter ξ , that is, the trustor's relevance as described in Equation (6); and the final computed reputation R^{new} as in the original model described in Section 7.

The parameters $(\alpha, \sigma) \in [0, 1] \subset \mathbb{R}$ are generally defined during tests and experiments. Typically, the α is defined as the difference between the resource value received and the thresholds expected.

6. Simulations and Experiments

A few simulations were conducted to evaluate the Meshtastic protocol's performance along with the modified reputation model in the context of IoT trust. Our experiments simulated an IoT network scenario where devices equipped with temperature sensors communicate over a Meshtastic-enabled network. To this end, we used the Meshtastic simulator known as Meshtasticator [25], which allowed us to emulate the Mesh networks, enabling the assessment of Meshtastic's dynamic adaptability and the influence of the reputation model on trust relationships.

Despite the existence of many IoT environments, due to the heterogeneity of the IoT devices, in our manuscript, we focus on combining a specific communication protocol and a trust/reputation scheme. In this sense, our solution can be adopted in a wide range of IoT application scenarios without taking into account the specific requirements of the single IoT environment.

In particular, we simulated a scenario involving ten IoT devices, each with a temperature sensor. In our simulated scenario, nine of these devices function as normal nodes, while the tenth device acts as a potential threat, reporting anomalous temperature values. The abnormal temperature data coming from this malicious or damaged device triggers dynamic adaptations within the Meshtastic protocol based on the reputation model.

We depict, in Figure 2, a simulated topology where a few nodes are mutually connected with high degree, and the node marked *malicious/damaged* is connected with a single node as a bridge to reach all the remaining nodes. To implement our modified version of the reputation model described in Section 4 within the Meshtasticator simulator's MeshNode

class, we had to integrate the reputation score computation and to update mechanisms based on the interactions between nodes.

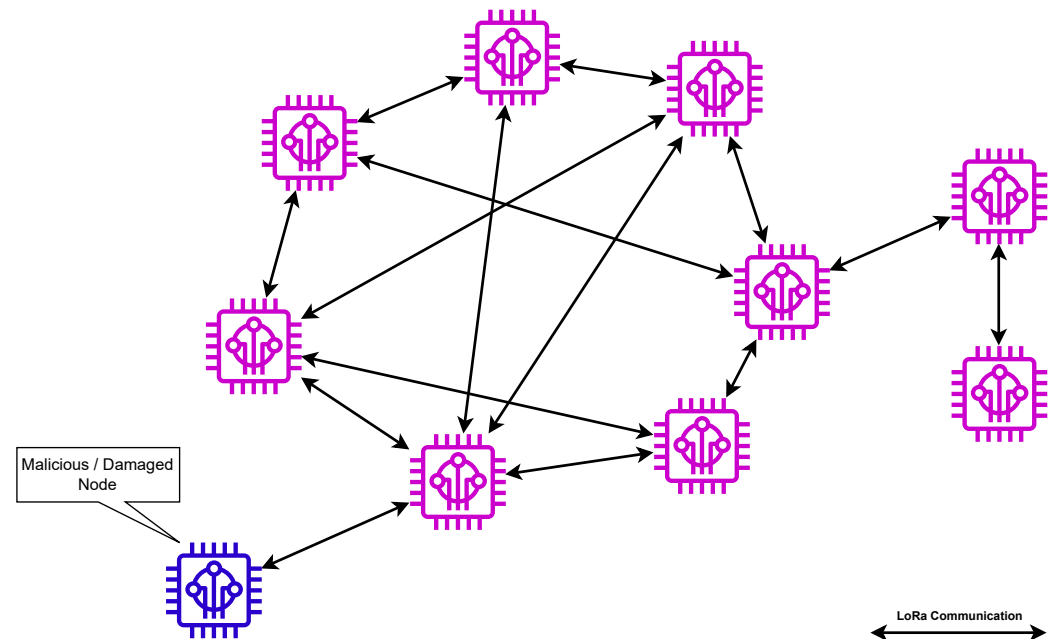


Figure 2. Architecture simulated for the experiment. The topology of the simulated network.

While in the following part of this section, we explain the main behaviour of the node, the reader can examine Appendix A, which contains a Python code snippet that outlines the behaviour of the `MeshNode` class. The entire behaviour of the `MeshNode` has been adapted to include the features defined in the reputation model described in this section: each node holds a list of interactions with nodes and the ability to calculate the new reputation and send their feedback to the network. In our study, the parameter α for the computation of the reputation assumes the value 0.5. The feedback is computed simply as follows: the value of the feedback will depend on the difference between the temperature locally detected and the temperature sent by a neighbour node. The Python code in Appendix A proves that we performed a very simple and linear integration of the reputation model into the Meshtastic protocol. Such a prototype can be considered a very effective blueprint for the future integration of similar and further features.

6.1. Results

We ran our simulations on a machine equipped with an Intel Core i7-13700H CPU and 32 GB of RAM, using Docker with Wale [26], using a pre-built simulator image. The simulations were performed based on the parameters described in Table 1. In particular, we used the European Frequency for LoRa communication (868 Mhz) because soon, we will perform a few experiments on the reputation model using real devices. In our laboratory, we hold LoRa modules running at a frequency of 868 Mhz. Moreover, the value of the Hop Limit shown in Table 1 was calculated after several attempts to tune the simulation, as well as the maximum retransmission of messages within the network.

Figure 3 shows how the nodes were spatially located in the topology (the plot was generated by the simulator), while Figure 4 depicts the message exchange between nodes.

Figure 5 shows the results after the first 8 h of simulations (then, eight temperature submissions by the nodes). The reputation score of the malevolent node, which is labelled as node 0, approaches values around 0.1, after a short transitory period while the reputation of the other nodes increases slowly after 11 h. These fluctuations are generally due to the cost of sending the resource and the number of jumps of each message. We remark that

the key point highlighted by this experiment is represented by the fact that the measured reputation quickly assumes values that reflect the real (simulated) behaviour of the nodes.

Placement of 10 nodes

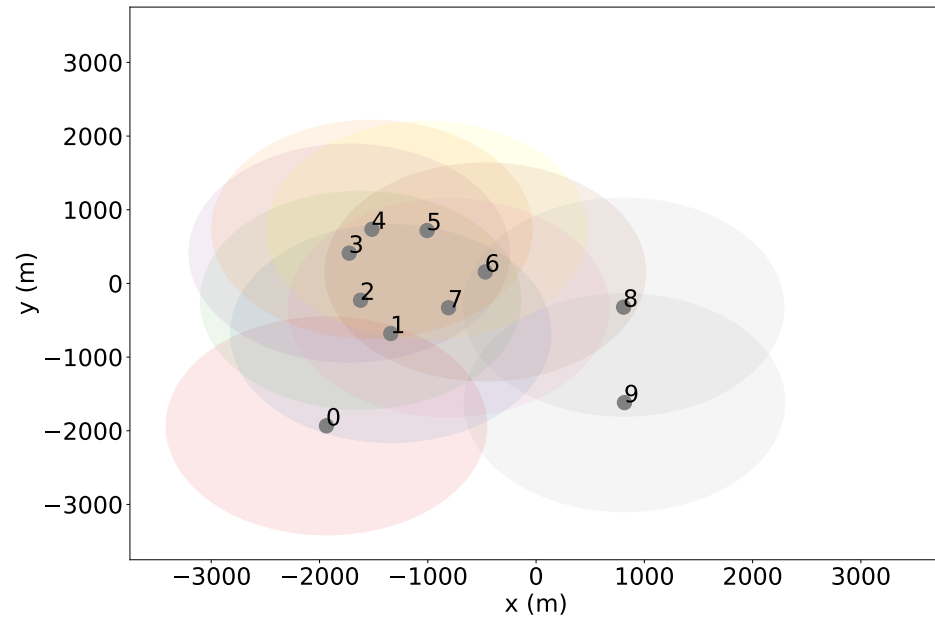


Figure 3. Architecture simulated for the experiment in the Meshtasticator simulator — spatial distribution of the simulated topology.

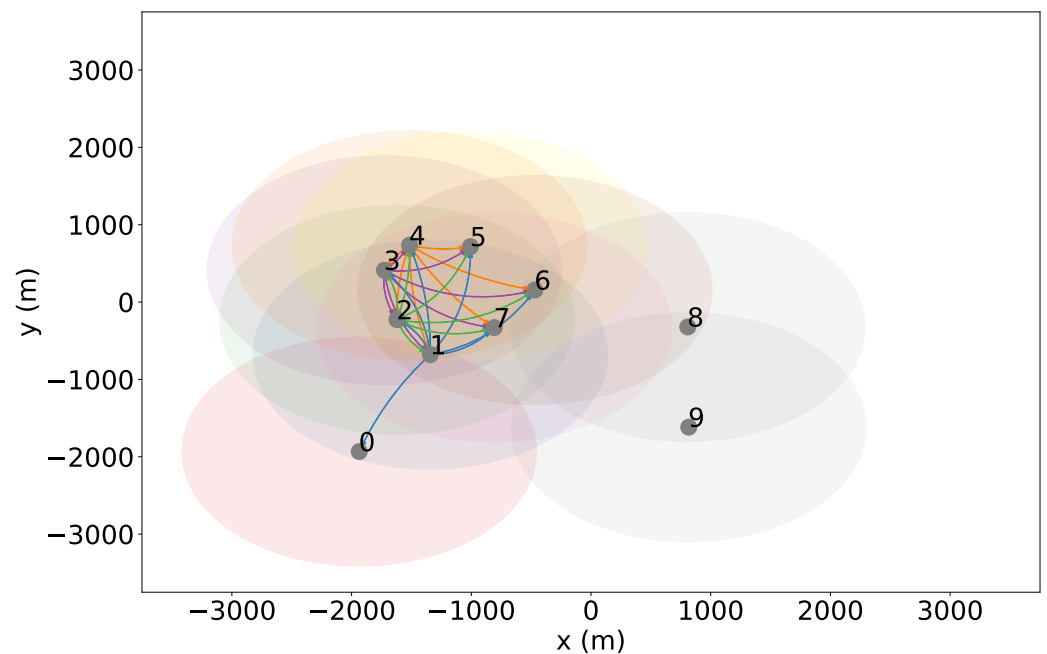


Figure 4. Messages of resource from Node 0 (malicious one) to the consumers in the Meshtasticator simulator.

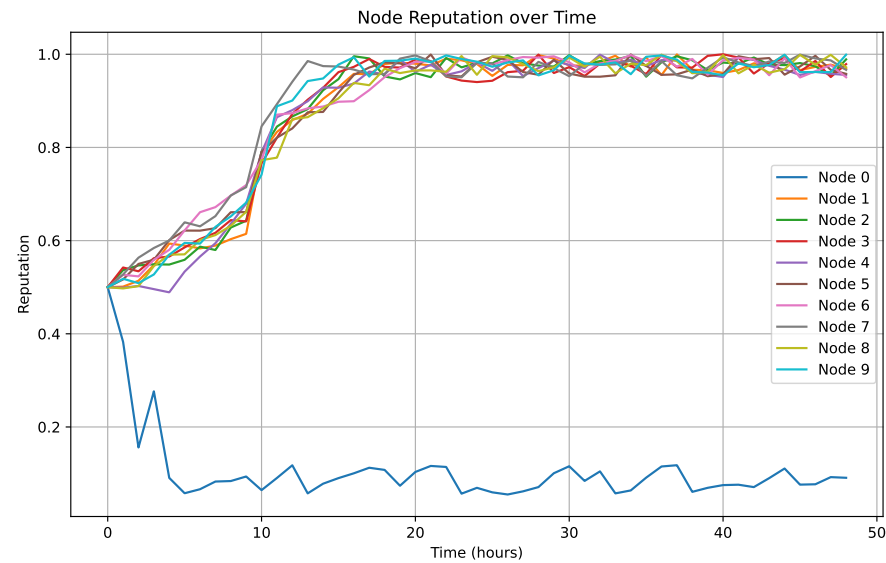


Figure 5. Results of malicious node detection based on reputation score in the Meshtastic simulator after 48 h (simulated).

The result clearly shows that the reputation model is effective in identifying a malicious/malfunctioning node. The simulations with Meshtastic were useful to show that integrating the reputation model is possible, thus providing trustworthiness in a Meshtastic-based IoT network. Nevertheless, results were obtained through simulations in a discrete simulation environment; therefore, we are planning to perform tests in a real-world IoT network.

Table 1. Simulation configuration.

Setting	Value
Parameters of the reputation model	
Number of malicious nodes	1
Hop limit	5
α	0.5
Other parameters	
Number of nodes	10
Pathloss model	Suburban macro-cell ¹
Max retransmission	3
Frequency (Mhz)	EU 868
Bandwidth (kHz)	240
Coding rate	4/8
Spreading factor	7
Data rate (kbps)	10.94
Simulation time (h)	48
Interval time (h)	1

¹ Specified by the 3rd Generation Partnership Project (3GPP) [27].

6.2. Experimental Results in a Real Environment

To perform such experiments in the real world, the Meshtastic firmware of the devices must be slightly modified, and proper devices (compatible with the modified firmware) must be selected. To implement the model, you need to create a module for Meshtastic, or a subset of *SinglePortModule*. This way, the created module will automatically reload incoming packets from the network. These are forwarded to a certain port for listening

(it is not true that they have relevance to the classic network ports); in our case, our port was defined as “REPUTATION_APP”, for which the port number was 256. The firmware was also forced to receive messages to simulate the architecture shown in Figure 2 because during the tests, the nodes were too close to not see each other, so all the nodes in the network (except for the border ones) ignored direct messages from nodes outside the network.

As Table 2 shows, to test our model in real devices, we used boards based on ESP32 microcontrollers, manufactured by Espressif Systems, a Chinese company based in Shanghai. The same has been used in related research for privacy preservation in IoT networks [28,29]. The used boards were the TTGO (or Lilygo) LoRa V2.1_1.6.1 equipped with ESP32 dual-core 32-bit microprocessors, 4 MB of flash memory, Wi-Fi and Bluetooth Low Energy, a 0.96 OLED display, an SX1276 868 MHz LoRa module and SMA LoRa antenna with 2 dBi of antenna gain; a Heltec Wireless Stick, also equipped with an ESP32 dual-core 32-bit microprocessor, 8MB of flash memory, Wi-Fi and Bluetooth Low Energy, an SX1276 868 MHz LoRa module and a LoRa antenna with 1 dBi of antenna gain; and finally, a new Arduino Nano ESP32 equipped with an ESP32-S3 dual-core 32-bit microprocessor, 16 MB of flash memory, Wi-Fi and Bluetooth Low Energy, and an external SX1276 868 MHz LoRa module with zero antenna gain (a wire was used as the antenna). As a temperature sensor, a classic DHT11 was used during the tests. Figures 6 and 7 show the arrangement of the experiments in the laboratory (nodes 1–9 and the malicious node). The nodes were arranged relatively close together, so communication was forced so that the architecture shown in Figure 2 was respected. The devices were power bank-powered and programmed to go into deep sleep about every 47 min to save energy.

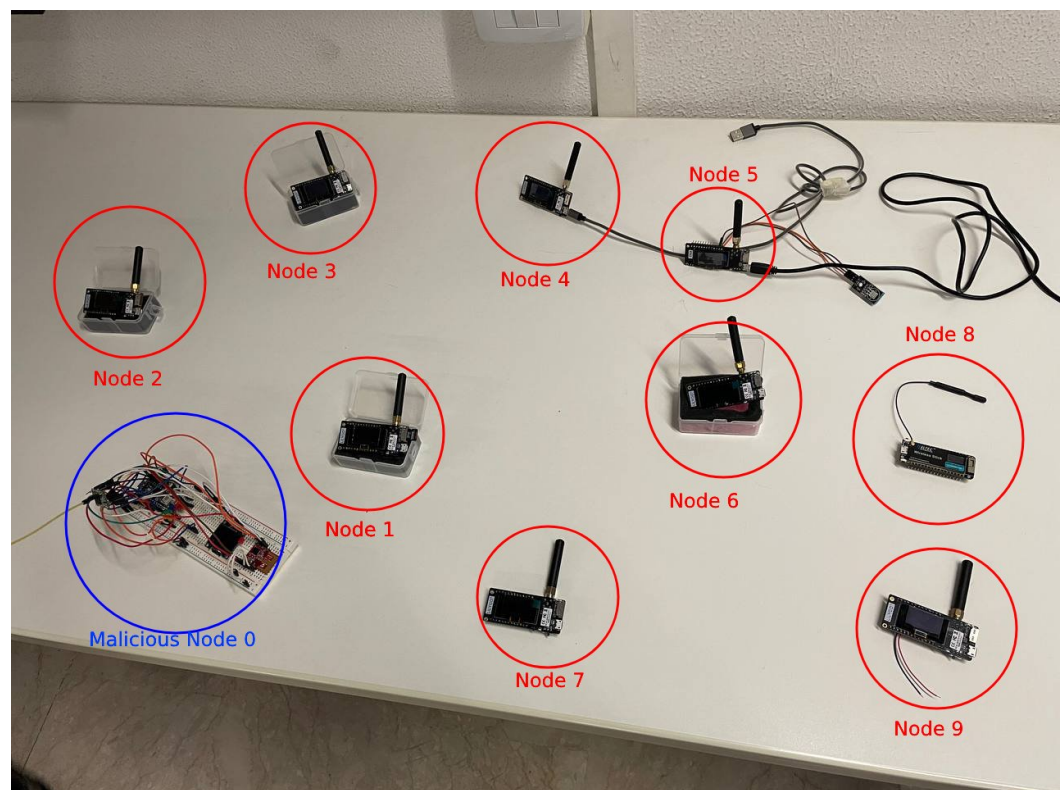
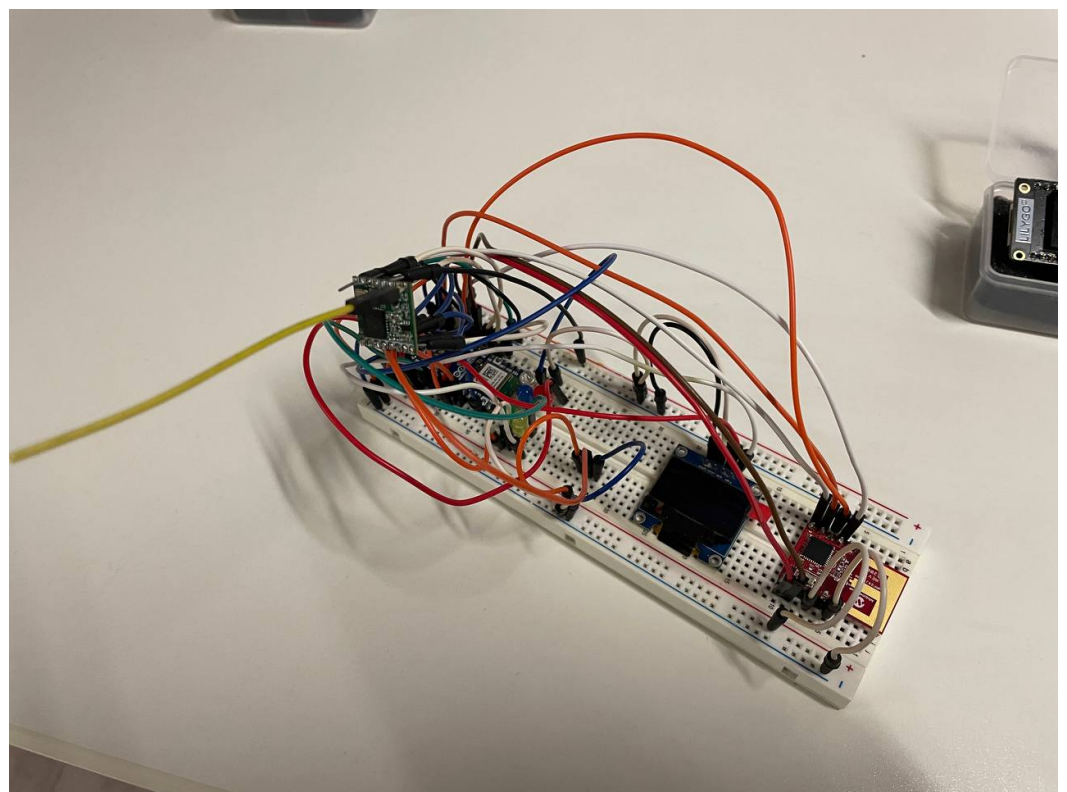


Figure 6. Simulated architecture with real nodes.

Table 2. Used devices.

	TTGO LoRa32 V2.1_1.6	Heltec Wireless Stick	Arduino Nano ESP32
Microcontroller	ESP32 dual-core 32-bit		ESP32-S3 dual-core 32-bit
Frequency	240 MHz		
ROM Memory	4 MB	8 MB	16 MB
RAM Memory	520 KB		
Built-in Modules	Wi-Fi, BLE, and LoRa		Wi-Fi and BLE
LoRa Module	SX1276 868 MHz Internal		SX1276 868 MHz External
LoRa Antenna Gain	2 dbi	1 dbi	0 dbi
Number of Nodes	8 nodes	1 node	1 node (malicious)

**Figure 7.** The malicious node (Arduino Nano ESP32) equipped with an external SX1276 868 MHz LoRa module.

Finally, Figure 8 contains the results of the experiments with our IoT devices after about 48 h: the behaviour of the devices is the same as in the simulations; they communicate the temperatures and exchange the feedback within the network. We can observe that, compared to the simulated environment, the measured reputation shows more oscillations. This is probably due to the delays in the communication of the LoRa packets. This study integrates trust thresholds to assess interactions among Smart Objects (SOs) based on feedback scores ranging from 0 to 1. These thresholds are pivotal in updating reputation scores and determining SO trustworthiness within the network. The simulations conducted to evaluate the integrated Meshtastic protocol with the reputation model yielded promising results, demonstrating significant improvements in the security and trustworthiness of IoT networks.

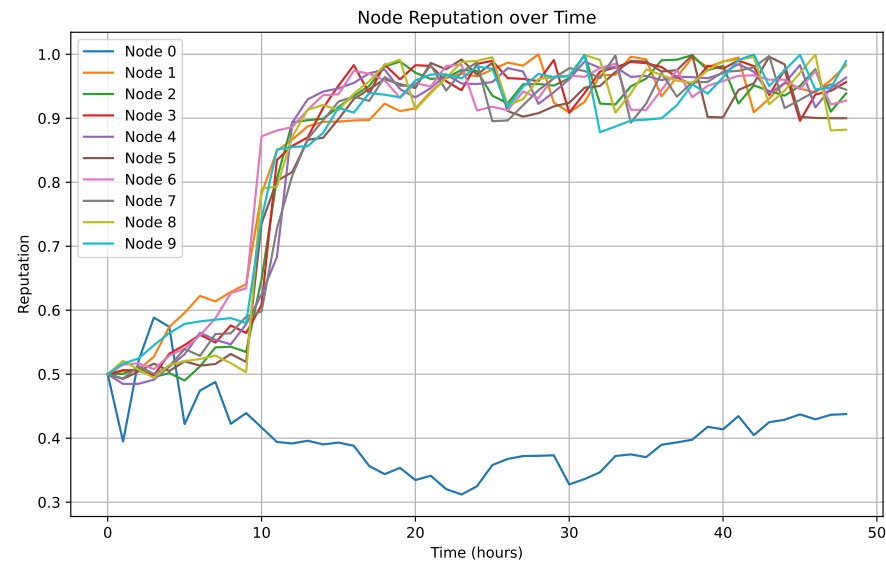


Figure 8. Results of malicious node detection based on reputation score in the real environment with real nodes after 48 h.

The analysis of network performance metrics revealed notable enhancements in latency, packet delivery rate, and energy efficiency compared to traditional IoT networks. Meshtastic's adaptive routing mechanisms, informed by reputation scores, enabled the establishment of robust communication paths, even in dynamic and challenging environments. Moreover, the security measures implemented within Meshtastic in response to potential threats, such as malicious or damaged nodes, proved effective in mitigating risks and preserving data integrity. By dynamically adjusting communication paths and isolating suspicious nodes based on reputation scores, the protocol demonstrated resilience against various security vulnerabilities. Overall, the results indicate that the integrated Meshtastic protocol with the reputation model holds significant promise for enhancing the security and trustworthiness of IoT networks.

The scalability of the Meshtastic protocol was evaluated by increasing the number of nodes in the simulated network. The results indicated that the protocol maintained robust communication paths and efficient routing even as the network grew. Additionally, the efficiency of the integrated protocol was analysed in terms of resource utilisation, including energy consumption and bandwidth usage. The adaptive routing mechanisms enabled by Meshtastic, informed by reputation scores, optimised resource allocation and minimised overheads, leading to the more efficient utilisation of network resources. Moreover, the security measures implemented within Meshtastic in response to potential threats, such as malicious or compromised nodes, were found to be highly efficient. By dynamically adjusting communication paths based on reputation scores, the protocol effectively isolated suspicious nodes while maintaining uninterrupted data transmission for legitimate devices. Furthermore, the simulations provided valuable insights into the behaviour of the reputation model under varying network conditions and loads. Observations regarding the model's adaptability and performance scalability informed the refinement of reputation management strategies for enhanced efficiency and reliability in large-scale deployments.

Overall, these results indicate that the Meshtastic protocol integrated with the reputation model offers a scalable, efficient, and secure solution for IoT networking. Future research directions may focus on further optimising resource utilisation, refining reputation management algorithms, and validating the approach through real-world deployments.

7. Conclusions

The integration of the Meshtastic protocol with LoRa technology can represent a promising solution to the security and trust challenges in IoT networks; the dynamic mesh networking capabilities of Meshtastic, combined with the low-power, long-range communication offered by LoRa, provide a robust and energy-efficient solution for secure IoT communication. The integration of a specific reputation model within the Meshtastic protocol further enhances the security of the system. In particular, we performed (i) a few simulations and (ii) a set of experiments with real devices. These experiments have validated the effectiveness of the studied approach in enhancing security and trust in IoT environments. For future research, we have planned to test the solution in a real test-bed, composed of a large number of IoT devices, to further validate its effectiveness and applicability. This work contributes to the ongoing discourse on securing IoT networks and provides a pathway for researchers, industry practitioners, and policymakers to fortify the foundation of trust in the expanding IoT landscape.

Author Contributions: All authors collaborated equally on this article and have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Italian Ministry of University and Research (MUR) Project “T-LADIES” under Grant PRIN 2020TL3X8X and in part by Pia.ce.ri. 2020–2022 funded by the University of Catania. The contribution of Federico Fausto Santoro was supported by MUR under Mission 4, Component 2, Investment 1.4 under the project HPC.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Listing A1. Python code.

```

1 class MeshNode:
2     def __init__(self, nodes, env, bc_pipe, nodeid, period, messages,
3         packetsAtN, packets, delays, nodeConfig):
4         # Existing code
5         self.reputation_score = 0.5 # Default reputation score
6         self.interactions = {} # Stores feedback and interaction details
7         self.temperature = random.uniform(20, 30) # Simulating normal
8         temperature values
9     def update_temperature(self):
10         self.temperature = random.uniform(20, 30) # Simulating normal
11         temperature values
12     def update_reputation(self, other_node_id, feedback, resource_relevance,
13         interaction_frequency, trustor_reputation):
14         """
15         Update the reputation score based on the feedback from an interaction
16         .
17         :param other_node_id: The ID of the node interacted with
18         :param feedback: The feedback score from the interaction
19         :param resource_relevance: The economic relevance of the resource
20         :param interaction_frequency: The frequency of interactions
21         :param trustor_reputation: The reputation of the trustor
22         """
23         # Compute the Interaction parameter
24         interaction = resource_relevance * interaction_frequency *
25         trustor_reputation * feedback
26
27         # Update the reputation score based on the interaction
28         # This is a simplified version and can be adjusted based on the model
29         's specifics
30         self.reputation_score = (self.reputation_score + interaction) / 2

```



```

27
28     # Store the interaction details
29     self.interactions[other_node_id] = {
30         'feedback': feedback,
31         'resource_relevance': resource_relevance,
32         'interaction_frequency': interaction_frequency,
33         'trustor_reputation': trustor_reputation
34     }
35
36     def get_reputation(self):
37         """
38         Get the current reputation score of the node.
39         """
40         return self.reputation_score
41
42     def get_interaction(self, other_node_id):
43         return self.interactions[other_node_id]
44
45     def receive(self, in_pipe):
46         # Existing code
47         # ....
48         self.isReceiving[self.isReceiving.index(True)] = False
49         except:
50             pass
51         self.airUtilization += p.timeOnAir
52         if p.collidedAtN[self.nodeid]:
53             continue
54         p.receivedAtN[self.nodeid] = True
55         delays.append(env.now-p.genTime)
56
57         # update hopLimit for this message
58         if p.seq not in self.leastReceivedHopLimit: # did not yet receive
59             packet with this seq nr.
60             self.usefulPackets += 1
61             self.leastReceivedHopLimit[p.seq] = p.hopLimit
62             if p.hopLimit < self.leastReceivedHopLimit[p.seq]:
63                 self.leastReceivedHopLimit[p.seq] = p.hopLimit
64
65             temperature = p.payload # The received temperature
66             feedback = max(0, 1 - (self.temperature - 20) / 10) # Lower
67             reputation for nodes reporting higher temperatures
68             new_interaction = self.get_interaction(p.txNodeId)
69
70             self.update_reputation(self, p.txNodeId, feedback, (p.hopLimit/self.
71             hopLimit), new_interaction.interaction_frequency+1, new_interaction.
72             trustor_reputation)
73             continue
74         # ....
75         # Existing code

```

References

1. Rose, K.; Eldridge, S.; Chapin, L. The internet of things: An overview. *Internet Soc. (ISOC)* **2015**, *80*, 1–50.
2. Li, S.; Xu, L.D.; Zhao, S. The internet of things: a survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [\[CrossRef\]](#)
3. Buffa, M.; Messina, F.; Santoro, C.; Santoro, F.F. Design of self-organizing protocol for LoWPAN networks. In Proceedings of the Internet and Distributed Computing Systems: 12th International Conference, IDCs 2019, Naples, Italy, 10–12 October 2019; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2019; pp. 424–433.
4. Djedjig, N.; Tandjaoui, D.; Medjek, F.; Romdhani, I. Trust-aware and cooperative routing protocol for IoT security. *J. Inf. Secur. Appl.* **2020**, *52*, 102467. [\[CrossRef\]](#)
5. Gowrishankar, J.; Kumar, P.S.; Narmadha, T.; Yuvaraj, N. A trust based protocol for Manets in Iot environment. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 2770–2775.
6. Yugha, R.; Chithra, S. A survey on technologies and security protocols: Reference for future generation IoT. *J. Netw. Comput. Appl.* **2020**, *169*, 102763. [\[CrossRef\]](#)
7. Tournier, J.; Lesueur, F.; Le Mouël, F.; Guyon, L.; Ben-Hassine, H. A survey of IoT protocols and their security issues through the lens of a generic IoT stack. *Internet Things* **2021**, *16*, 100264. [\[CrossRef\]](#)

8. Meshtastic. An Open Source, Off-Grid, Decentralized, Mesh Network Built to Run on Affordable, Low-Power Devices. Available online: <https://meshtastic.org/docs/introduction> (accessed on 3 March 2024).
9. Almuhaaya, M.A.; Jabbar, W.A.; Sulaiman, N.; Abdulmalek, S. A survey on LoRawan technology: Recent trends, opportunities, simulation tools and future directions. *Electronics* **2022**, *11*, 164. [CrossRef]
10. What Is LoRawan Specification. Available online: <https://lora-alliance.org/about-lorawan/> (accessed on 3 March 2024).
11. Ahmed, A.I.A.; Ab Hamid, S.H.; Gani, A.; Khan, M.K.; Khan, M.K. Trust and reputation for Internet of Things: Fundamentals, taxonomy, and open research challenges. *J. Netw. Comput. Appl.* **2019**, *145*, 102409. [CrossRef]
12. Chen, J.; Tian, Z.; Cui, X.; Yin, L.; Wang, X. Trust architecture and reputation evaluation for internet of things. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3099–3107. [CrossRef]
13. Chen, D.; Chang, G.; Sun, D.; Li, J.; Jia, J.; Wang, X. TRM-IoT: A trust management model based on fuzzy reputation for internet of things. *Comput. Sci. Inf. Syst.* **2011**, *8*, 1207–1228. [CrossRef]
14. Asiri, S.; Miri, A. An IoT trust and reputation model based on recommender systems. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; pp. 561–568.
15. Hussain, Y.; Zhiqiu, H.; Akbar, M.A.; Alsanad, A.; Alsanad, A.A.A.; Nawaz, A.; Khan, I.A.; Khan, Z.U. Context-aware trust and reputation model for fog-based IoT. *IEEE Access* **2020**, *8*, 31622–31632. [CrossRef]
16. Grabatin, M.; Hommel, W. Self-sovereign Identity Management in Wireless Ad Hoc Mesh Networks. In Proceedings of the 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), Bordeaux, France, 17–21 May 2021; pp. 480–486.
17. Oliviero, F.; Romano, S.P. A reputation-based metric for secure routing in wireless mesh networks. In Proceedings of the IEEE GLOBECOM 2008–2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 8 December 2008; pp. 1–5.
18. Yu, Y.; Peng, Y.; Yu, Y.; Rao, T. A new dynamic hierarchical reputation evaluation scheme for hybrid wireless mesh networks. *Comput. Electr. Eng.* **2014**, *40*, 663–672. [CrossRef]
19. Li, Y. A reputation system for wireless mesh network using multi-path routing protocol. In Proceedings of the 30th IEEE International Performance Computing and Communications Conference, Orlando, FL, USA, 17–19 November 2011; pp. 1–6. [CrossRef]
20. Radiohead: Packet Radio Library for Embedded Microprocessors. Available online: <https://www.arduino.cc/reference/en/libraries/radiohead/> (accessed on 3 March 2024).
21. Al-Shamaileh, M.; Anthony, P.; Charters, S. Evaluating Trust and Reputation Models for IoT Environment. In Proceedings of the Agents and Multi-Agent Systems: Technologies and Applications 2022; Jezic, G., Chen-Burger, Y.H.J., Kusek, M., Šperka, R., Howlett, R.J., Jain, L.C., Eds.; Springer: Singapore, 2022; pp. 49–60.
22. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarnè, G.M.L. A Social Edge-Based IoT Framework Using Reputation-Based Clustering for Enhancing Competitiveness. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 2051–2060. [CrossRef]
23. Laroui, M.; Nour, B.; Mounghla, H.; Cherif, M.A.; Afifi, H.; Guizani, M. Edge and fog computing for IoT: A survey on current research activities & future directions. *Comput. Commun.* **2021**, *180*, 210–231.
24. De Donno, M.; Tange, K.; Dragoni, N. Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. *IEEE Access* **2019**, *7*, 150936–150948. [CrossRef]
25. Meshtasticator: Discrete-Event and Interactive Simulator for Meshtastic. Available online: <https://github.com/GUVWAF/Meshtasticator> (accessed on 3 March 2024).
26. Santoro, C.; Messina, F.; D’Urso, F.; Santoro, F.F. Wale: A Dockerfile-Based Approach to Deduplicate Shared Libraries in Docker Containers. 2018; pp. 776–784. Available online: <https://ieeexplore.ieee.org/document/8511978> (accessed on 3 March 2024).
27. TR 125 996-V13.0.0-Universal Mobile Telecommunications System (UMTS); Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations (3GPP TR 25.996 Version 13.0.0 Release 13). Available online: https://www.etsi.org/deliver/etsi_tr/125900_125999/125996/13.00.00_60/tr_125996v130000p.pdf (accessed on 3 March 2024).
28. Meli, D.; Milotta, F.L.M.; Santoro, C.; Santoro, F.F.; Riccobene, S. Privacy Preserving on Delay-Tolerant Networks. In Proceedings of the International Conference on Innovative Computing and Communications, Delhi, India, 16–17 February 2023; Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A., Eds.; Springer: Singapore, 2023; pp. 163–171.
29. Meli, D.; Milotta, F.L.M.; Santoro, C.; Santoro, F.F.; Riccobene, S., An Adaptive Blurring Routing Protocol for Delay-Tolerant Networks in IoT Environments. In *Security, Trust and Privacy Models, and Architectures in IoT Environments*; Fotia, L., Messina, F., Rosaci, D., Sarnè, G.M., Eds.; Springer International Publishing: Berlin, Germany, 2023; pp. 63–76. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.