July 15, 2019

Contents

List of Figures

List of Tables

General introduction

We are witnessing the emergence of the Internet of Things, a new paradigm that is revolutionizing the field of telecommunication networks. The Internet of Things, which is an integral part of the Internet of the future, consists of a wide interconnection of all kinds of objects (other than computers and mobile phones) in our environment, for example vehicles, roads, home, television, etc. for an ambient and intelligent world. Linking these objects with the Internet, is usually supported by sensors connected to the Internet, along with many other non-less important technologies, such as RFID (Radio Frequency Identification), Drones, and so on.

Indeed, IoT cannot achieve its goals unless many issues could be effectively resolved. Security is making an important part of those issues. In this work, we are particularly interested in trust as security-specific issue. The establishment of trustful connections among the IoT-connected devices is very important as those devices exchange sensitive and private information. We have addressed trust issue in the context of Drone-WSN communications. The sensor nodes assign dynamic trust degree to the drone they communicate with and the communication takes place only if that trust degree is acceptable. We have conducted simulations on cooja simulator to assess our solution.

This document is organized in three chapters. The first chapter presents generalities on IoT. The second chapter studies the trust issue in the IoT. The final third chapter provides an overview on the realized solution as well as the evaluation results.

Chapter 1

Internet of things

1.1 Introduction

The Internet of Things (IoT) provides connectivity for anyone at any time and place to anything at any time and place. With the advancement in technology, we are moving towards a society, where everything and everyone will be connected [1]. The IoT is considered as the future evaluation of the Internet that realizes machine-to-machine (M2M) learning [2]. The basic idea of IoT is to allow autonomous and secure connection and exchange of data between real world devices and applications. The IoT links real life and physical activities with the virtual world. This chapter presents the generalities about the internet of Things.

1.2 Definition

The IoT consists of objects, sensor devices, communication infrastructure, computational and processing unit that may be placed on cloud, decision making and action invoking system [3]. The objects have certain unique features and are uniquely identifiable and accessible to the Internet. These physical objects are equipped with Radio-Frequency IDentification (RFID) tags or other identification bar-codes that can be sensed by the smart sensor devices [4]. The sensors communicate object specific information over the Internet to the computational and processing unit. A combination of different sensors can be used for the design of smart services.

The result of processing is then passed to the decision making and action invoking system that determines an automated action to be invoked.

1.3 The internet of things ecosystem

The Internet has tremendously evolved in the last few years connecting billions of things globally. These things have different sizes, capabilities, processing and computational power and support different kind of applications [4]. Thus, the traditional Internet merges into smart future Internet, called IoT. The generic scenario of IoT is shown in figure 1.1. The IoT connects real world objects and embeds the intelligence in the system to smartly process the object specific information and take useful autonomous decisions. Thus, IoT can give birth to enormous useful applications and services that we never imagined before With the advancement in technology, the devices processing power and storage capabilities significantly increased while their sizes reduced. These smart devices are usually equipped with different type of sensors and actuators. are able to connect and communicate over the Internet that can enable a new range of opportunities [5]. Moreover, the physical objects are increasingly equipped with RFID tags or other electronic bar codes that can be scanned by the smart devices, e.g., smart phones or small embedded RFID scanner. The objects have unique identity and their specific information are embedded in the RFID tags. In 2005, the International Telecommunications Union (ITU) proposed that "Internet of Things" will connect the real world objects in both a sensory and intelligent manner [6] figure1.2 shows basic IoT system implementing different type of applications or services.

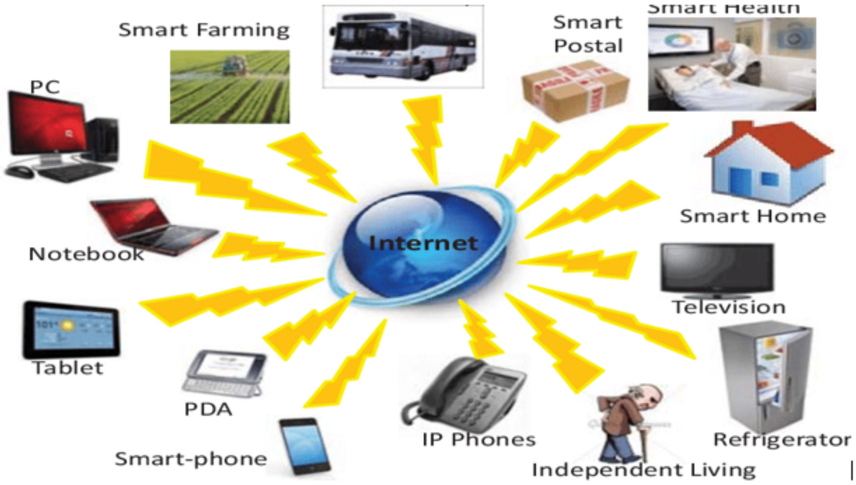


Figure 1.1: The IoT generic scenario

The things connect and communicate with other things that implement the same service type. The basic simplified workflow of IoT can be described as follows

- Object sensing, identification and communication of object specific information. The information is the sensed data about temperature, orientation, motion, vibration, acceleration, humidity, chemical changes in the air etc depending on the type of sensors. A combination of different sensors can be used for the design of smart services.

- Trigger an action. The received object information is processed by a smart device/system that then determines an automated action to be invoked.

- The smart device/system provide rich services and includes a mechanism to provide feedback to the administrator about the current system status and the results of actions invoked.

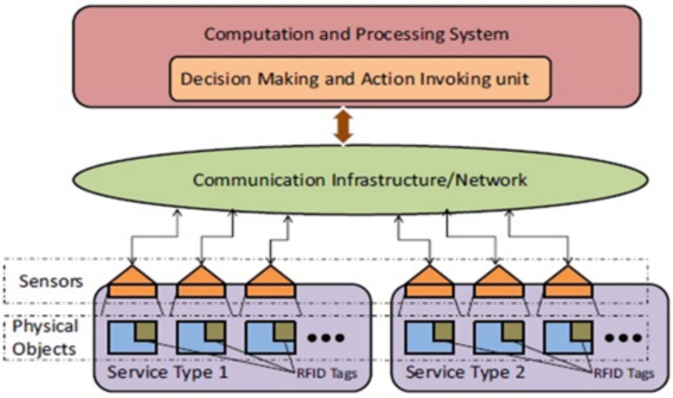

Computation and Processing System
Decision Making and Action Invoking unit
Communication Infrastructure/Network
Sensors
Physical Objects
Service Type 1
RFID Tags
Service Type 2
RFID Tags

Figure 1.2: Basic IoT system

1.4 Generic IoT architecture

The IoT's standard architecture is composed of four layers figure 1.3 [7]:

1.4.1 Perception Layer

The Perception layer is also known as 'Device Layer'. It consists of the physical objects and sensor devices. The sensors can be RFID, 2D-barcode, or Infrared sensor depending upon objects identification method. This layer basically deals with the identification and collection of objects specific information by the sensor devices.

1.4.2 Network Layer:

The Network layer can also be called ‘Transmission Layer’. This layer securely transfers the information from sensor devices to the information processing system. The transmission medium can be wired or wireless and technology can be 3G, UMTS, Wifi, Bluetooth, infrared, ZigBee, etc depending upon the sensor devices. Thus, the Network layer transfers the information from Perception layer to Middleware layer.

1.4.3 Middleware Layer:

The devices over the IoT implement different type of services. Each device connects and communicates with only those other devices which implement the same service type. This layer is responsible for the service management and has link to the database. It receives the information from Network layer and store in the database. It performs information processing and ubiquitous computation and takes automatic decision based on the results

1.4.4 Application Layer:

This layer provides global management of the application based on the objects information processed in the Middleware layer. The applications implemented by IoT can be smart health, smart farming, smart home, smart city, intelligent transportation, etc.

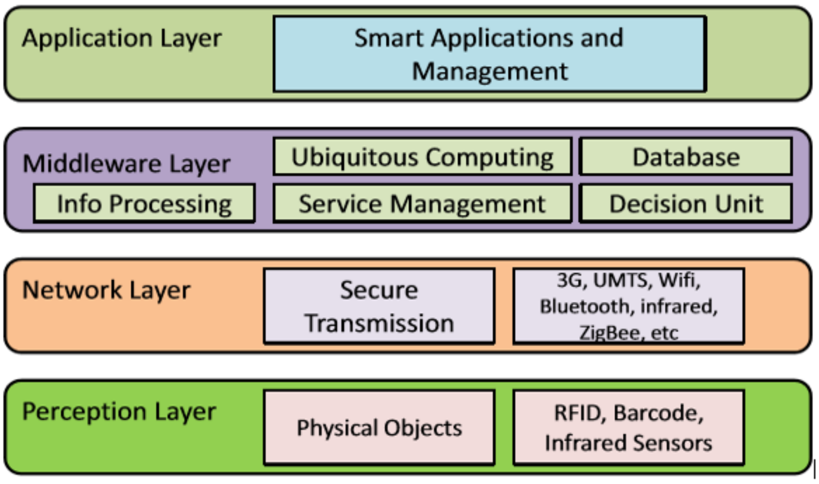


Figure 1.3: The IoT Architecture

1.5 Potential applications

The IoT can find its applications in almost every aspect of our daily life. Below are some of the examples [8].

1.5.1 Prediction of natural disasters:

The combination of sensors and their autonomous coordination and simulation will help to predict the occurrence of land-slides or other natural disasters and to take appropriate actions in advance.

1.5.2 Industry applications:

The IoT can find applications in industry e.g., managing a fleet of cars for an organization. The IoT helps to monitor their environmental performance and process the data to determine and pick the one that need maintenance.

1.5.3 Water Scarcity monitoring:

The IoT can help to detect the water scarcity at different places. The networks of sensors, tied together with the relevant simulation activities might not only monitor long term water interventions such as catchment area management, but may even be used to alert users of a stream, for instance, if an upstream event, such as the accidental release of sewage into the stream, might have dangerous implications.

1.5.4 Design of smart homes:

The IoT can help in the design of smart homes e.g., energy consumption management, interaction with appliances, detecting emergencies, home safety and finding things easily, home security etc. Design of smart homes: The IoT can help in the design of smart homes e.g., energy consumption management, interaction with appliances, detecting emergencies, home safety and finding things easily, home security etc.

1.5.5 Medical applications:

The IoT can also find applications in medical sector for saving lives or improving the quality of life e.g., monitoring health parameters, monitoring activities, support for independent living, monitoring medicines intake etc.

1.5.6 Agriculture application:

A network of different sensors can sense data, perform data processing and inform the farmer through communication infrastructure e.g., mobile phone text message about the portion of land that need particular attention. This may include smart packaging of seeds, fertilizer and pest control mechanisms that respond to specific local conditions and indicate actions.

1.5.7 Intelligent transport system design:

The Intelligent transportation system will provide efficient transportation control and management using advanced technology of sensors, information and network. The intelligent transportation can have many interesting features such as non-stop electronic highway toll, mobile emergency command and scheduling, transportation law enforcement, vehicle rules violation monitoring, reducing environmental pollution, anti-theft system, avoiding traffic jams, reporting traffic incidents, smart beaconing, minimizing arrival delays etc.

1.5.8 Design of smart cities

The IoT can help to design smart cities e.g., monitoring air quality, discovering emergency routes, efficient lighting up of the city, watering gardens etc.

1.5.9 Smart metering and monitoring

The IoT design for smart metering and monitoring will help to get accurate automated meter reading and issuance of invoice to the customers. The IoT can also be used to design such scheme for wind turbine maintenance and remote monitoring, gas, water as well as environmental metering and monitoring.

1.5.10 Smart Security:

The IoT can also find applications in the field of security and surveillance e.g., surveillance of spaces, tracking of people and assets, infrastructure and equipment maintenance, alarming etc.

1.6 IoT Elements

We present a taxonomy that will aid in defining the components required for Internet of Things from a high level perspective [9].

1.6.1 Radio Frequency Identification (RFID)

RFID technology is a major breakthrough in the embedded communication paradigm which enables design of microchips for wireless data communication. They help in automatic identification of anything they are attached to acting as an electronic barcode [10-11]. The passive RFID tags are not battery powered and they use the power of the reader‘s interrogation signal to communicate the ID to the RFID reader. This has resulted in many applications particularly in retail and supply chain management. The applications can be found in transportation (replacement of tickets, registration stickers) and access control applications as well. The passive tags are currently being used in many bank cards and road toll tags which is among the first global deployments. Active RFID readers have their own battery supply and can instantiate the communication. Of the several applications, the main application of active RFID tags is in port containers for monitoring cargo

1.6.2 Wireless Sensor Networks (WSN):

Recent technological advances in low power integrated circuits and wireless communications have made available efficient, low cost, low power miniature devices for use in remote sensing applications. The combination of these factors has improved the viability of utilizing a sensor network consisting of a large number of intelligent sensors, enabling the collection, processing, analysis and dissemination of valuable information, gathered in a variety of environments [12]. Sensor data are shared among sensor nodes and sent to a distributed or centralized system for analytics. It is worth mentioning at this level that WSNs are the most important integral part in the IoT. Other technologies of non-least importance may coexist with the already mentioned technologies, such as Drones or UAV (unmanned aerial vehicle) devices in general.

1.6.3 Data storage and analytics:

One of the most important outcomes of this emerging field is the creation of an unprecedented amount of data. Storage, ownership and expiry of the data become critical issues. The internet consumes up to 5% of the total energy generated today and with these types of demands, it is sure to go up even further. Hence data centers which run on harvested energy and which are centralized will ensure energy efficiency as well as reliability. The data have to be stored and used intelligently for smart monitoring and actuation. It is important to develop artificial intelligence algorithms which could be centralized or distributed based on the need. Novel fusion algorithms need to be developed to make sense of the data collected. State-of-the-art nonlinear, temporal machine learning methods based on evolutionary algorithms, genetic algorithms, neural networks, and other artificial intelligence techniques are necessary . to achieve automated decision making. These systems show characteristics such as interoperability, integration and adaptive communications. They also have a modular architecture both in terms of hardware system design as well as software development and are usually very well-suited for IoT applications [13].

1.7 Key challenges:

The IoT can offer enormous economic benets but it also faces many key challenges [12-14]. Some of them are brieу described here.

1.7.1 Naming and Identity Management:

The IoT will connect billions of objects to provide innovative services. Each object/sensor needs to have a unique identity over the Internet. Thus, an efcient naming and identity management system is required that can dynamically assign and manage unique identity for such a large number of objects.

1.7.2 Interoperability and Standardization:

Many manufacturers provide devices using their own technologies and services that may not be accessible by others. The standardization of IoT is very important to provide better interoperability for all objects and sensor devices.

1.7.3 Information Privacy:

The IoT uses different kind of object identification technologies e.g., RFID, 2D-barcodes etc. Since, every kind of daily use objects will carry these identification tags and embed the object specific information, it is necessary to take proper privacy measures and prevent unauthorized access.

1.7.4 Objects safety and security:

The IoT consists of a very large number of perception objects that spread over some geographic area, it is necessary to prevent the intruder's access to the objects that may cause physical damage to them or may change their operation.

1.7.5 Data confidentiality and encryption:

It is necessary that the sensor devices should have proper encryption mechanism to guarantee the data integrity at the information processing unit. The IoT service determines who can see the data, thus, it is necessary to guard the data from externals .

1.8 Conclusion

This chapter presented the Internet of Things, their applications, the technological elements making it possible. Then we presented the main challenges that should be effectively considered if we want the IoT project achieves all of its pre-established goa

Chapter 2

Trust in IoT

2.1 Introduction

As already seen in the previous chapter, the IoT data are often critical and need for security and safe communication among the devices handling them. In this context, the establishment of trustful links between the networked IoT objects, as well as between these objects and their users is of high importance. Trust in IoT is strongly related to security. It is even considered as one important aspect of it. Another important concept related to trust is user privacy that is the ability of an entity to determine whether, when, and to whom information about itself is to be used [15]. A trust-based security should take special care of users' privacy, which is one of the ways to gain user trust. The current chapter is devoted to a study of the trust issue in the Internet of things context.

2.2 Network model for Trust management in loT

The network architecture of IoT systems consists of three layers: Sensor Layer, Network layer and Application layer.

- Sensor layer includes many wireless sensor networks deployed everywhere. The main tasks of sensor network are gathering and processing information from physical world.

- Network layer mainly provides inter-connecting and routing services to transmit sensor information overall internet.

- Application layer provides context-aware intelligent services to IoT users.

A good trust ensuring system should ideally consider every layer of the IoT framework.[17]

2.3 Trust establishment

In this section, we highlight the trust establishment context in the IoT, as well as the trust management.

2.3.1 Trust parts

The trust establishment functionality is the assessment of reputation for a given entity, regarding its earlier behaviors. Trust establishment involves three main parts the trustee, the trustor and the context of trust establishment [16-18].

- The trustee: determines the entity that is concerned by trust estimation procedure. In IoT context, the trustee could be any connected object in the IoT or classical internet hosts.

- The trustor: is the part that makes trust decision either autonomously or cooperatively. In other words, the trustor is the responsible of reputation evaluation.

- Context of trust relationship, such as the purpose of trust, the environment of trust (e.g., time, location, activity type of used devices, their operational mode, etc.). Context is a very important factor inuencing trust. It species the situation where trust exists [17].

2.3.2 Goals of trust management

The following purposes need to be addressed while providing trust for IoT with the consideration of the already discussed network model [18]. ‘

2.3.2.1 Trust relationship and decision (TRD)

Trust management provides an effective way to evaluate trust relationships of the IoT entities and assist them to make a wise decision to communicate and collaborate with each other. Trust relationship evaluation (in short trust evaluation) concerns all IoT system entities in all layers and plays a fundamental role for intelligent and autonomic trust management.

2.3.2.2 Data perception trust (DPT)

Data sensing and collection should be reliable in IoT. In this aspect, we pay attention to the trust properties like sensor sensibility, preciseness, security, reliability, and persistence, as well as data collection efficiency, i.e., the trustee's objective properties in the IoT physical perception layer.

2.3.2.3 Privacy preservation (PP)

User privacy including user data and personal information should be flexibly preserved according to the policy and expectation of IoT users.

2.3.2.4 Data transmission and communication trust (DTCT)

Data should be transmitted and communicated securely in the IoT system. Unauthorized system entities cannot access private data of others in data communications and transmission. This goal is related to the security and privacy properties of IoT system wherein light security/trust/privacy solution is needed. Trusted routing and key management in IoT networks are two important issues required to be solved for achieving this objective.

2.3.2.5 Quality of IoT services (QIoTS)

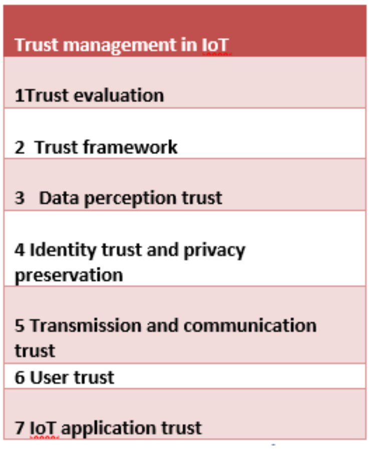
Quality of IoT services should be ensured. "Only here, only me and only now" services are expected , which implies that the IoT services should be personalized and precisely offered at exactly right place and time to a right person. This objective is mainly about the trust management in the IoT application layer, but required to be supported by other layers. The QIoTS TM objective concerns not only the objective properties of IoT services (the trustee), but also the objective and subjective properties of users (the trustor), as well as context.

2.3.2.6 Identity trust (IT)

The identifiers of IoT system entities are well managed for the purpose of trustworthy IoT. Scalable and efficient identity management in IoT is expected.

2.4 Trust management in IoT

In this section, we study the existing works based on many taxonomies among: Trust Evaluation, Trust Framework, Data Perception Trust, Identity Trust and Privacy Preservation, Transmission and Communication Trust, User Trust, IoT Application Trust [18].table 2.1



Trust management in IoT
1Trust evaluation
2 Trust framework
3 Data perception trust
4 Identity trust and privacy preservation
5 Transmission and communication trust
6 User trust
7 IoT application trust

Table 2.1: Trust management in IoT

2.4.1 Trust evaluation

Trust evaluation is a technical approach of representing trust relationships for digital processing, in which the properties influencing trust will be evaluated. A solution proposed in [19] presents a trust management protocol considering both social trust and QoS trust metrics and using both direct observations and indirect recommendations to update trust. Concretely, three trust properties: honesty, cooperativeness, and community interest are considered in the trust evaluation of IoT things. The honesty trust property represents whether or not a node is honest. The cooperativeness trust property represents whether or not the trustee is socially cooperative with the trustor The community-interest trust represents whether or not the trustor and trustee are in the same social groups (e.g., in a co-location or co-work relationship) or have the similar capabilities. In this work, trust was defined and quantified using social network theory [20] and evaluated based on both direct observations and indirect recommendations. The effectiveness of the trust management protocol was demonstrated in a service composition application. It is one of the first to consider social relationships in trust management for IoT. Another solution have further studied the scalability, adaptability and survivability of the trust management protocol in a dynamically changed IoT system [21]. The solution in [22] proposed a trust management model based on fuzzy reputation for IoT that considers a specific IoT environment consisting of only wireless sensors with QoS trust metrics containing such elements as packet forwarding/delivery ratio and energy consumption. Based on a social Internet of Things (SIoT) paradigm, according to which the objects are capable of establishing social relationships in an autonomous way with respect to their owners, Another solution proposed in [23] studied how the information provided by other members of the Social IoT has to be processed to set a reliable system on the basis of the behavior of the objects. It defined a model for trust management where each node computes the trust of its friends on the basis of its own experience and the opinion of common friends with potential service providers. A feedback system is employed and the credibility and centrality of the IoT nodes are applied to evaluate the trust level. Existing work in this taxonomy considered some objective and subjective properties of trustee for trust evaluation and decision in the context of IoT. But context-aware based on social computing has not yet been seriously investigated.

2.4.2 Trust framework

Trust framework is the system architecture designed to achieve trust management of a whole. The solution proposed in [24] provides an IoT system architecture that offers a solution to the broad array of challenges in terms of general system security, network security, and application security with respect to the basic information security requirements of data confidentiality, integrity, and availability, authority, non-repudiation, and privacy preservation.

The trust architecture proposed in [25] contains a perception logical layer, a mark recognition logical layer, a decision control logical layer and a trusted interface logical layer. The trusted interface logical layer is consisted of cross-layer trusted protocols by which the network architecture interacts with the perception logical layer, the mark-recognition logical layer and the decision-control logical layer. This architecture can afford trusted and reliable data transmission in wireless sensor networks.

The work presented in [26-27] briefly reviewed the research progress of IoT, paying special attention to security. By means of deeply analyzing the security architecture and features, security requirements were given in each layer of IoT, such as lightweight cryptographic algorithm and protocol, integrity and authenticity of sensor data, key agreement in the physical perception layer; identity authentication, anti-DDoS (Distributed Denial of Service), encryption mechanism and communication security in the network layer; secure multi-party computation, secure cloud computing and anti-virus for data processing; authentication and key management, security education and management, and privacy preservation in the application layer.

A general architecture of trusted security system for IoT was proposed in [28], which mainly includes trusted user module, trusted perception module, trusted terminal module, trusted network module and module agent module. [29] proposed a comprehensive set of trust-enhancing security functional components for the resolution infrastructure as a crucial part of IoT. These components cover not only basic IoT resource access control, but also essential functions such as identity management, key exchange and management and trust and reputation management. This component composition with its interdependencies provides compulsory mechanisms for securing communications between subjects to guarantee an inviolable interaction and therefore ensure data integrity and confidentiality, service trust and privacy of users. The solution proposed in [30] identified the requirements (i.e., interoperability, automation, decentralization and contextualization) for resources to be more resilient in IoT and proposed an architectural model of Self Managed Security Cell, which leverages on current knowledge in large scale security systems, information management and autonomous systems. This model supports policy based access control on IoT resources. In [31] we find a virtualization and abstract model based on IoT to build the data security mechanism that could protect the privacy of user data and the security of personal information.

2.4.3 Data perception trust

Data perception trust concerns IoT data trust during collection and pre-process in the physical perception layer. Javed and Wolf addressed one data perception trust problem:

how to verify sensor information that is gathered from multiple sensors that are managed by different entities using outlier detection [32-33]. They developed a technique for automatically deriving a model of the physical phenomenon that is measured by the sensors. This model is then used to compare sensor readings and to identify outliers through spatial and temporal interpolation. The system was evaluated in the context of weather sensing and is applicable to any application domains where the underlying phenomenon is continuous. Another solution [34] provided many key technologies to resist against different attacks temper proong of the embedded devices in IoT by applying the concept of trusted. Security and privacy issues have been studied in [35-36] with RFID technology and analyzed various threats of the RFID system components (e.g., blocking and jamming devices, relay attack, eavesdropping, replay attack, tag cloning, personal and location privacy intrusion) and elucidated how these issues can be resolved or risks can be mitigated. [37] proposed DARE, a hybrid architecture combining wireless sensor networks (WSNs) with wireless mesh networking paradigm in order to provide secure data aggregation and node reputation in WSNs. DARE uses a secure veriable multilateration technique that allows the network to retain the trustworthiness of aggregated data even in the presence of malicious node. DARE can effectively reduce the amount of data exchanged over the wireless medium and achieve battery lifetime improvement to the wireless sensors. This work is advanced in supporting TRD, DPT, DFMT, and SSR. But it did not consider data privacy although data trustworthiness during fusion is enhanced efciently.

2.4.4 Identity trust and privacy preservation

A number of studies aim to improve identity trust and achieve privacy preservation in IoT. In [37], a solution that studied identity management for user location privacy adapted to situations such as emergency and non-emergency cases based on policy management and user authentication, as well as access control on user location information. [38] proposed a trust extension protocol to support secure mobility management for extending and adapting the network to changes of location and infrastructure, increase fault tolerance capacity, connectivity, dependability and scalability in IPv6-based Wireless Sensor Networks (6LoWPAN). [39] analyzed the threat of unauthorized tracking by a compromised RFID discovery service in the current industrial standard and proposed a pseudonym-based RFID discovery service architecture that provides practical privacy protection against unauthorized tracking to mitigate this threat. This design protects against database reading attack by a semi-trusted discovery service and provides efficient key management and access control. Data privacy preservation is an important aspect for achieving data trust in IoT. [40] applied information flow control techniques and tagged data in IoT with their privacy properties, which allows a trusted computing base to control data access based on privacy policies. In addition, computing performance issue about tagging within resource-constrained sensors was also considered in this study. This work is significant with regard to data trust and privacy preservation. However, it relies on trusted computing technologies. Its execution performance in the physical perception layer needs extensive investigation. A privacy protection solution was proposed by [41] for IoT. It contains a user controlled privacy-preserving access control protocol, context aware anonymity privacy policy mechanisms in order to control which of personal data is being collected and accessed, who is collecting and accessing such data, and when these are happening.

2.4.5 Transmission and communication trust

Data transmission and communication is important for achieving IoT trust. Existing advances in networking and communications can be used in order to achieve trust. In particular, the trustworthy IoT networking and communication protocols should support the heterogeneous and specific IoT networking context, which rises new issues and challenges.

A security protocol for data transfer amongst the things was proposed in [42], together with a security framework for enhancing trust and privacy for IoT system infrastructure. Lightweight symmetric encryption (for data) and asymmetric encryption (for key exchange) in Trivial File Transfer Protocol (TFTP) were suggested in order to make the proposed protocol applicable in the context of IoT.

The applicability and limitations of existing Internet protocols and security architectures in the context of IoT have been discussed in [43]. They presented challenges and requirements for IP-based security solutions and highlighted specific technical limitations of standard IP security protocols. There was underlined that trust-ensuring solutions should take into account the resource-constrained nature of things and heterogeneous communication models. Lightweight security mechanisms and group security that are feasible to be run on small things and in IoT context should be developed, with particular focus on possible DoS/DDoS attacks.

2.4.6 User trust

User trust in IoT applications is essential for the success of IoT. [44] investigated trust in an IoT setting with considerable aspects: transitivity and reflexivity, psychological aspects of risk and risk assessment, distrust, deception, retaliation and altruism, reputations, association and brands, and human brain. It was pointed that it is obvious that one cannot fully trust any of the IoT components (e.g., software, hardware, communications, etc.), but this does not mean that humans cannot or should not trust IoT services at all. An interesting solution proposed in [45] provides a differential game model to study user behaviors in IoT interactions between selfish and malicious nodes. They obtained optimal amount of network resource to invest in information security and packet forwarding and studied how the vulnerability of information and the potential loss from such vulnerability affects the optimal amount of resources that should be devoted to securing that information. The evaluation of the solution showed that malicious behaviors could be discovered with a high probability.

2.4.7 IoT application trust

There are quite a number of IoT applications in a variety of areas of our life with some support on trust by satisfying partial trust management objectives. In the following, we present an examples. privacy preserving smart meter based load management system was proposed by using secure multi-party computation and homomorphic encryption as its security primitives [46]. It fully achieved preservation of the detailed user data, kept the data resolution for proposed smart grid control and management functionalities with a verification process, and did not need the support of a trusted third party. Secure multiparty computation based techniques are often used to perform audio database search tasks, such as music matching, with privacy preservation [47] explained the security flaws of secure multi-party computation and analyzed the resulting tradeoff between privacy and computational complexity in the music matching application. [48-49] developed a lightweight framework for ensuring security, privacy and trustworthiness of life-logging in smart environments including the use of lightweight versions of IP protocols.

2.4.8 Related works

This table shows related works table 2.2

Searcher	Solution	Explanation
[19]	a trust management protocol considering both social trust and QoS trust metrics and using both direct observations and indirect recommendations to update trust.	three trust properties: honesty, cooperativeness, and community interest are considered in the trust evaluation of IoT things.

[22]	a trust management protocol considering both social trust and QoS trust metrics and using both direct observations and indirect recommendations to update trust.	three trust properties: honesty, cooperativeness, and community interest are considered in the trust evaluation of IoT things.
[23]	studied how the information provided by other members of the Social IoT has to be processed to set a reliable system on the basis of the behavior of the objects.	defined a model for trust management where each node computes the trust of its friends on the basis of its own experience and the opinion of common friends with potential service providers.
[24]	provides an IoT system architecture	offers a solution to the broad array of challenges in terms of general system security, network security, and application security with respect to the basic information security requirements of data confidentiality, integrity, and availability, authority, non-repudiation, and privacy preservation.
[26-27]	briefly reviewed the research progress of IoT, paying special attention to security.	analyzing the security architecture and features, security requirements were given in each layer of IoT, such as lightweight cryptographic algorithm and protocol, integrity and authenticity of sensor data, key agreement in the physical perception layer; identity authentication, anti-DDoS (Distributed Denial of Service), encryption mechanism and communication security in the network layer; secure multi-party computation, secure cloud computing and anti-virus for data processing; authentication and key management, security education and management, and privacy preservation in the application layer.

[43]	The applicability and limitations of existing Internet protocols and security architectures in the context of IoT	presented challenges and requirements for IP-based security solutions and highlighted specific technical limitations of standard IP security protocols. There was underlined that trust-ensuring solutions should take into account the resource-constrained nature of things and heterogeneous communication models. Lightweight security mechanisms and group security that are feasible to be run on small things and in IoT context should be developed, with particular focus on possible DoS/DDoS attacks.

Table 2.2: Table related works

2.5 Conclusion

In the present chapter, we have presented some aspects related to trust issue in the context of the Internet of Things. The next chapter will deal with the presentation of our work's trust issue as well as the solution we have been working on.

Chapter 3

Presentation of the realized solution

3.1 Introduction

In this chapter, we describe our work's context in terms of threat model and motivations. Then, we present the model of our solution. After that we detail the realization step description of the simulation tool. This chapter reviews the simulation scenarios and the results obtained. The latter will be the subject of a discussion.

3.2 Context and threat model

3.2.1 General context

Drones are unmanned aerial vehicles that are capable of interconnecting networks in a highly flexible manner. Indeed, drones present three-dimensional mobile relay entities that gather data in many sensitive IoT applications. These applications can be military (e.g. monitoring of land borders and unattended areas) as they can be civil (such as disaster management, military applications, agricultural environment monitoring). Regardless the details of the application scenario, the core communications are usually taking place between Drones and sensor nodes deployed to accomplish a well-determined mission. The Drone flies over the sensor nodes so that to collect their sensorial data. Then, it travels back to the requesting or also called decision maker station and give it the collected data. At this stage, it is worth mentioning that

3.2.2 Threat model

As a flying node, the drone do not keep staying around the wireless sensor network. It rather visits the WSN and flies away alternatively. From a security perspective, this inherent feature can be seen as a vulnerability that can be exploited to threaten the overall network and disrupt its mission. Wireless sensor networks are recognized for being sensitive to compromising acts that aim at poisoning the sensor's behavior and consequently, exercise different cyber-attack scenarios. Furthermore, according to [50,51], drones can be also prone to be maliciously captured and compromised during their missions. This time, the aim of the adversary is either to steal the sensitive data carried by the drone and thereafter spy on the application secrecy. Another possibility is to make of the drone an adversary that threatens the Sensor Networks it interacts with. In our work, we focus on the latest threat possibility; the case when the wireless sensor network carries the risk of getting attacked by a corrupted drone. And since WSN-Drone communication are likely in critical IoT applications, DoS (Denial of Service) threats are in this case the most dangerous attacks. We recall that DoS attacks have many possible scenarios that all share the same goal that is to make a sensitive service unavailable. The figure bellow depicts the assumed threat model.figure 3.1

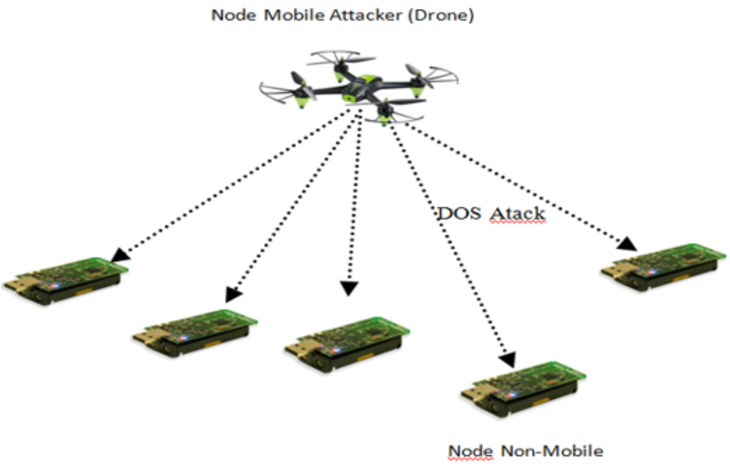


Figure 3.1: threat model

3.3 Scope of our solution's model

Our solution relies on a trust-based security mechanism that is meant to protect the wireless sensor network against possible DoS attack originated from malicious Drone. We assume a request-response communication model between the drone and the WSN, where the drone is the requester (client) and the WSN nodes: the sensors are the responders (servers). A simple way to perform DoS attack is to send massive requests to the sensor nodes in order to exhaust their resources.

3.3.1 Solution's algorithm

```
Trust_value <-0 ; is_hacker<-0; nb_requests <- 0;
Threshold : fixed amount of normal number of requests that can be
received in a well-determined period of time.
    BEGIN
Do { if ( is_hacker [DroneID] = 0 )
    nb_request <- nb_request + 1;
    if ( nb_requests > threashold )
    Trust_value <- Trust_value – punishment_amount;
    if (Trust_Value < 0)
    is_hacker [DroneID] = 1;
    else
    Trust_value <- Trust_value + reward_amount;
    Respond to the drone's request.
    While(request_received ())
    END
```

3.4 Discussion

The problem we are dealing with is still a new issue in ioT security field, and has not yet been thoroughly addressed. Our solution adopts a traditional trust estimation policy and adapts it to our context. Such a policy increases dynamically the trust score of the drone as long as he behaves normally, and punishes it otherwise. A negative trust score induces the complete isolation of the Drone. At this stage, it is important to underline that malicious Drone that combines DoS with identity spoofing attacks may compromise the presented solution.

3.5 Implementation context

3.5.1 Contiki OS

Contiki is an open source operating system. It is a modular configurable system for Internet of Things networks. The hybrid architecture of the Contiki kernel allows two modes of operation: either multitasking or event-based. Contiki is an operating system designed to take up the least space possible, with a low memory footprint. For this, the code is written in C language. A system using Contiki contains processes, which can be applications or services, ie. a process that provides functionality to one or more applications. Communication between processes is done by sending events. The Contiki kernel remains, natively, an operating system based on events. To get the multitasking mode, a library must be installed. Functions associated with this library do not directly access all wireless object resources. In some cases, they must use the part of the kernel dedicated to event management. This two-level structure results in a degradation of system performance when multitasking is enabled. [52]

Contiki provides mechanisms that help in the programming of smart object applications. It gives. Libraries for memory allocation, handling of chained lists and communication abstraction. Developed in C with all its applications, and it is portable towards different architectures [53].

3.5.2 Architecture

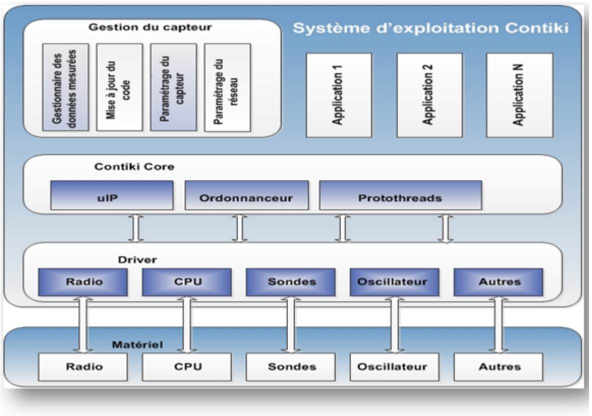


Figure 3.2: Contiki operating system architecture.

Contiki consists of a kernel, libraries, a scheduler and a process set. Like any operating system, its role is to manage physical sheet metal resources as processor, memory, computer peripherals (input / output) figure 3.2.

3.5.3 The applicative protocol CoAP

Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.[54].in the figure 3.3 CoAP protocol architecture.

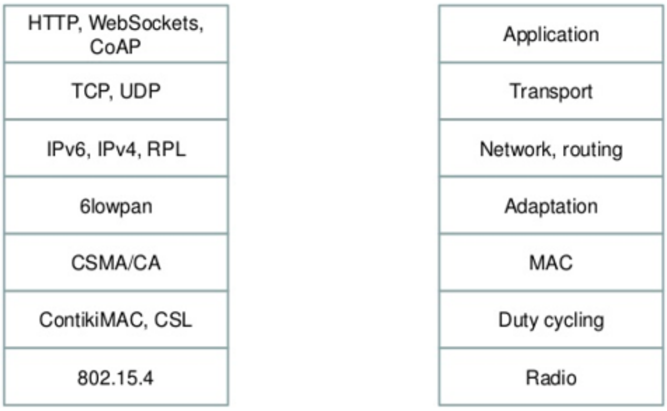


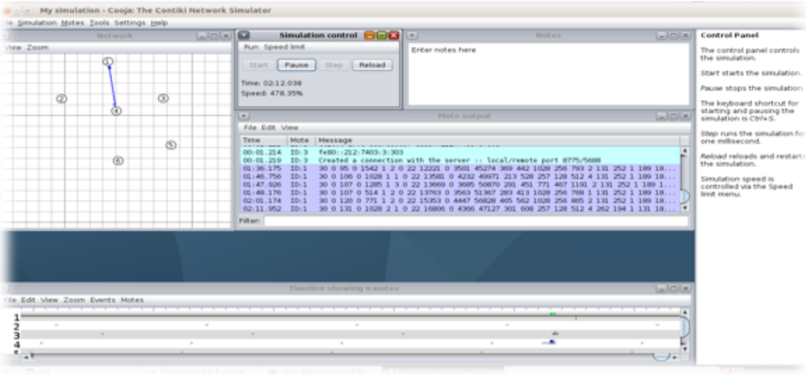
Figure 3.3: The structure of ContikiCOAP.

3.5.4 Cooja

Cooja is a network simulator for Contiki. It can simulate sensor networks (Internet of Things) regardless of their size. The sensors can be emulated: at the hardware level, which is slower, but allows precise control of system behavior, or at a less detailed level, which is faster and allows the simulation of large networks (larger networks) [55].figure 3.4

3.6 The different tools needed

We will work under Ubuntu, for this we use the following tools:



Time: 02:12.038
Speed: 478.35%

Enter notes here

```
00:01.214  ID:3  fe80::212:7403:3:303
00:01.219  ID:3  Created a connection with the server :: local/remote port 8775/5688
01:36.175  ID:1  30 0 95 0 1542 1 2 0 22 12221 0 3501 45274 369 442 1028 256 793 2 131 252 1 189 18...
01:46.756  ID:1  30 0 106 0 1028 1 1 0 22 13581 0 4232 49971 213 528 257 128 512 4 131 252 1 189 18...
01:47.926  ID:1  30 0 107 0 1285 1 3 0 22 13669 0 3685 50870 291 451 771 467 1191 2 131 252 1 189 1...
01:48.176  ID:1  30 0 107 0 514 1 2 0 22 13763 0 3563 51367 283 413 1028 256 768 1 131 252 1 189 18...
02:01.174  ID:1  30 0 120 0 771 1 2 0 22 15353 0 4447 56828 465 562 1028 256 865 2 131 252 1 189 18...
02:11.952  ID:1  30 0 131 0 1028 2 1 0 22 16806 0 4366 47127 301 608 257 128 512 4 262 194 1 131 18...
```

Control Panel

The control panel controls the simulation.

Start starts the simulation.

Pause stops the simulation.

The keyboard shortcut for starting and pausing the simulation is Ctrl+S.

Step runs the simulation for one millisecond.

Reload reloads and restarts the simulation.

Simulation speed is controlled via the Speed limit menu.

Figure 3.4: Image capture of Cooja.

3.6.1 ANT

Ant is a software created by the Apache Foundation that aims to automate the repetitive operations of software development such as compilation, document generation (Javadoc) or archiving in JAR format, like the Make software.

To install this first package, you will have to use the following command in a terminal:

sudo apt-get install ant

3.6.2 The MSP430 compiler

Cooja simulates network communications using the MSPSim emulator to finely emulate (at the instructional level) the execution of a program on an MSP430-based platform.

We will need a compiler adapted to this architecture. The following command installs it:

sudo apt-get install gcc-msp430

3.6.3 The JDK

Since Apache Ant is written in Java, it needs a virtual machine (JVM: Java Virtual Machine) to work.

We will install Open-JDK (Java Development Kit) to use it.

sudo apt-get install openjdk-8-jdk-headless

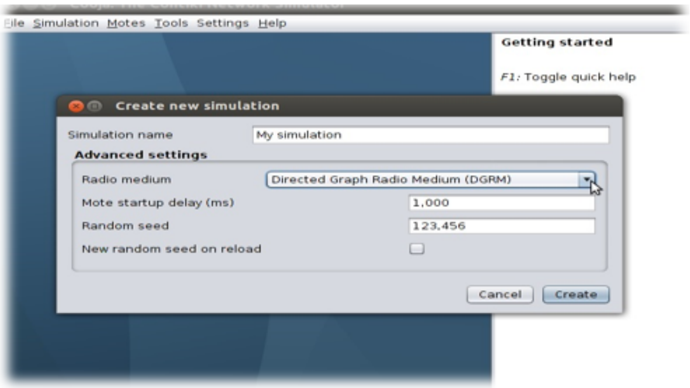
3.6.4 Download and installation of Contiki

Contiki 2.7 can be downloaded at the following address:
https://sourceforge.net/projects/contiki/files/Instant%20Contiki/Instant
%20Contiki%203.0/InstantContiki2.7.zip/download

3.7 Starting the application

Navigate to the Contiki folder (contiki 2.7) and navigate to the / tools / cooja directory. Run the sudo ant run command to open the cooja interface graph.
$ cd contiki2.7/tools/cooja
$ cd contiki3.0/tools/cooja
Then, we will create a new simulation from the menu File / NewSimulation.figure 3.5

Give the simulation a name in the Simulation Name field. Choose Directed Graph Radio Medium (DGRM) from the Radio Medium drop-down menu under Advanced Settings. Click the Create button. The new simulation is unleashed and it opens many windows as shown in the figure bellow. figure 3.6



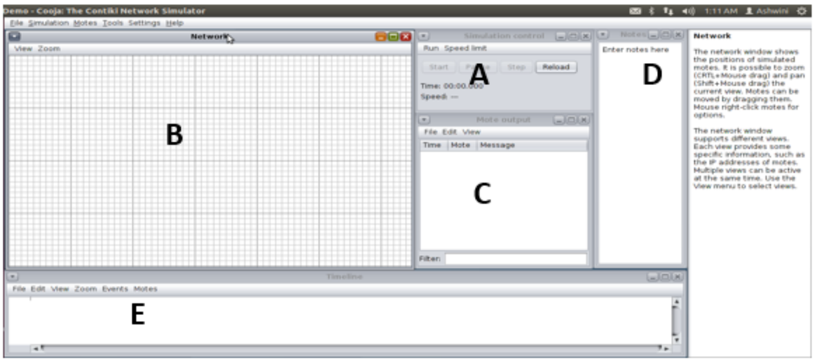
Simulation name: My simulation
Radio medium: Directed Graph Radio Medium (DGRM)
Mote startup delay (ms): 1,000
Random seed: 123,456
New random seed on reload: □

Figure 3.5: Create a new simulation.

A. The Timeline window: displays all the communication events in the simulation, very useful for understanding what is going on in the network.
B. The Network window: at the top left of the screen, shows us all the nodes in the simulated network.
C. The Mote Output window, on the right side of the screen, shows us all the impressions on the serial port of all the nodes. D. The Notes window at the top right is where we can

put notes for our simulation.

E. The Simulation control window is where we can start, pause and load our simulation.



Network

The network window shows the positions of simulated motes. It is possible to zoom (CRTL+Mouse drag) and pan (Shift+Mouse drag) the current view. Motes can be moved by dragging them. Mouse right-click motes for options.

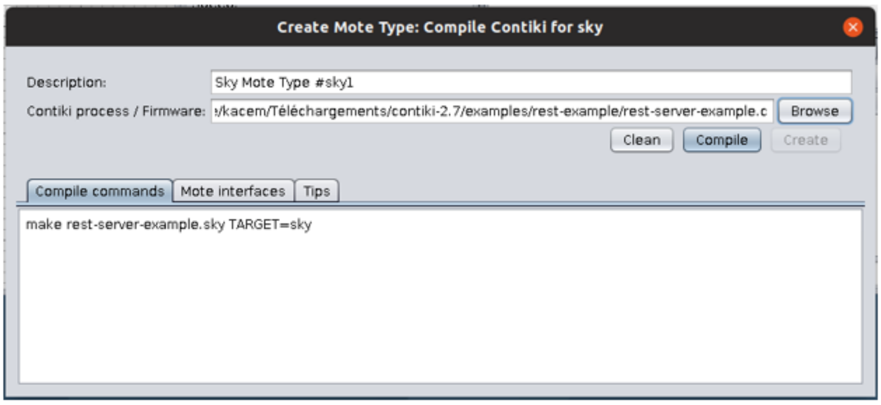
The network window supports different views. Each view provides some specific information, such as the IP addresses of motes. Multiple views can be active at the same time. Use the View menu to select views.

Figure 3.6: Screenshot of the simulator interface.

3.8 Add nodes to create a network

3.8.1 Add the node "Server-CoAP"

To add a Server Type Node, the code rest-server-example example rest-example (/examples/rest-example) is used. However, you can download any code that you want to implement based on your application. Click the Compile button. A "Creat" button appears on a successful compilation that adds a number of Nodes as desired in the network.figure 3.7



Description: Sky Mote Type #sky1

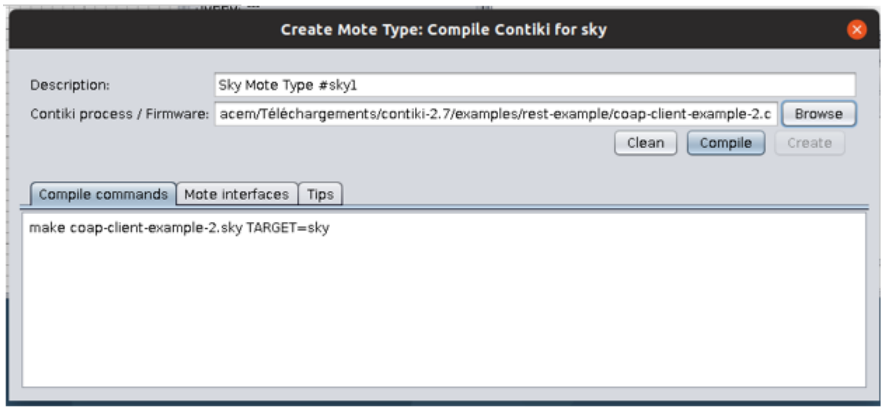
Contiki process / Firmware: :/kacem/Téléchargements/contiki-2.7/examples/rest-example/rest-server-example.c

```
make rest-server-example.sky TARGET=sky
```

Figure 3.7: Add the node "Server-COaP".

3.8.2 Add the node "Client-CoAP"

To add a Client Node, use the client code from the coap-client-example-2 example (/examples/rest-example) figure 3.8



Create Mote Type: Compile Contiki for sky

Description: Sky Mote Type #sky1

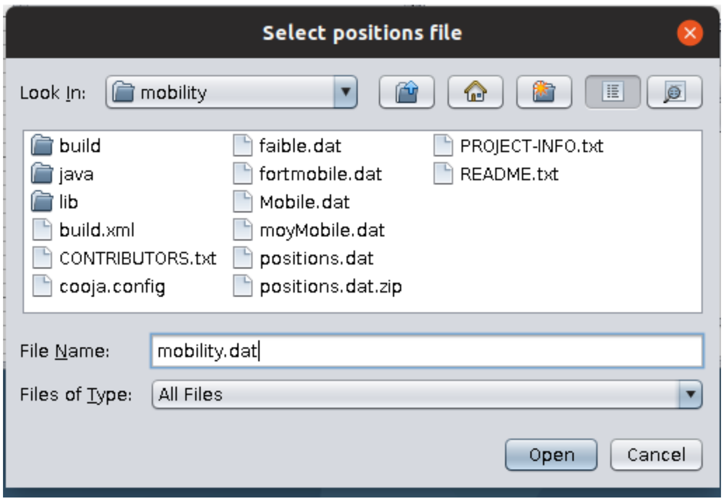
Contiki process / Firmware: acem/Téléchargements/contiki-2.7/examples/rest-example/coap-client-example-2.c

```
make coap-client-example-2.sky TARGET=sky
```

Figure 3.8: Add the node "Client-COaP".

3.8.3 Adding the mobility

Access: tool->mobility. then open the file (mobility.dat) .figure 3.9



Select positions file

Look In: mobility

build
java
lib
build.xml
CONTRIBUTORS.txt
cooja.config
faible.dat
fortmobile.dat
Mobile.dat
moyMobile.dat
positions.dat
positions.dat.zip
PROJECT-INFO.txt
README.txt

File Name: mobility.dat

Files of Type: All Files

Figure 3.9: Add the mobility.

3.9 Evaluation parameters

3.9.1 Energy consumption

We assess here the performance of the solution regarding the required energy amounts throughout the simulation time.

Energy consumption is directly related to power. In our case, the power depends on time parameter T. The latter is in the form of a percentage linked to the use of the radio in transmission and in reception. The energy consumption remains proportional to the use of the radio. The following equation expresses how we quantify energy consumption in Contiki:

Energy (milliJule)= [[(tempsLPM*0.545 milliampère) + (tempsCPU*1.8 mA)+ (tempsTX*17.7 mA) + (tempsLISTEN * 20 mA)] * 3 volt] / 32768

Energy (milliJule) = Power x time of execution of the simulation (seconds). The times are expressed in ticks (number of clock pulses). 32768 is the number of ticks generated per second, in contiki 2.7. The operational voltage of the TmoteSky sensors for example is 3V.

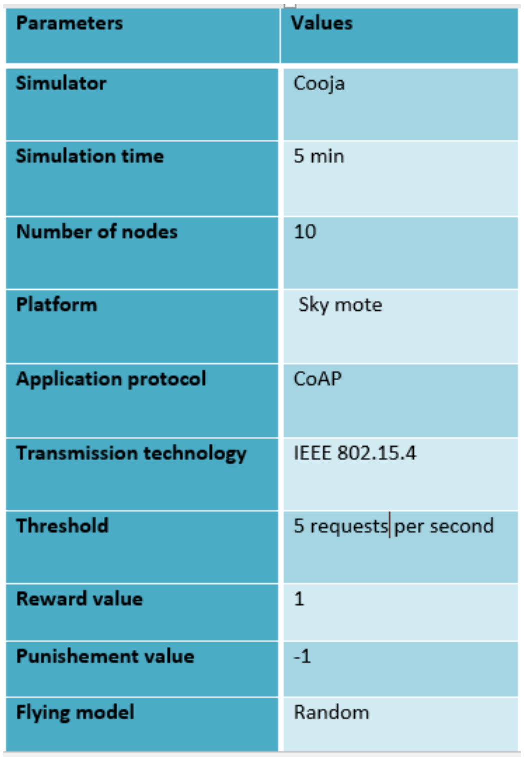
3.9.2 Security efficiency

Besides the energy consumption parameter that is a very important in the assessment of any iot-destined solution , the evaluation of security effectiveness is also important. In our case we have estimated the isolation delay with each sensor node in the network within different movement scenarios of the malicious Drone.

3.10 Simulation Results

3.10.1 Simulation parameters

The main simulation parameters are summarized in the table bellow.table 3.1



Parameters	Values
Simulator	Cooja
Simulation time	5 min
Number of nodes	10
Platform	Sky mote
Application protocol	CoAP
Transmission technology	IEEE 802.15.4
Threshold	5 requests per second
Reward value	1
Punishement value	-1
Flying model	Random

Table 3.1: Simulation parameters.

3.10.2 The obtained results

3.10.3 Energy Vs Time(before and after the solution)

This figure above depicts the total energy consumption in terms of time before and after the solution, where we note that the amount of energy consumed with the trust solution enhances the amount of energy consumption. This is mainly due to the fact that massive responding to the malicious drone's requests is curbed with the integration of the trust-based solution. figure 3.10

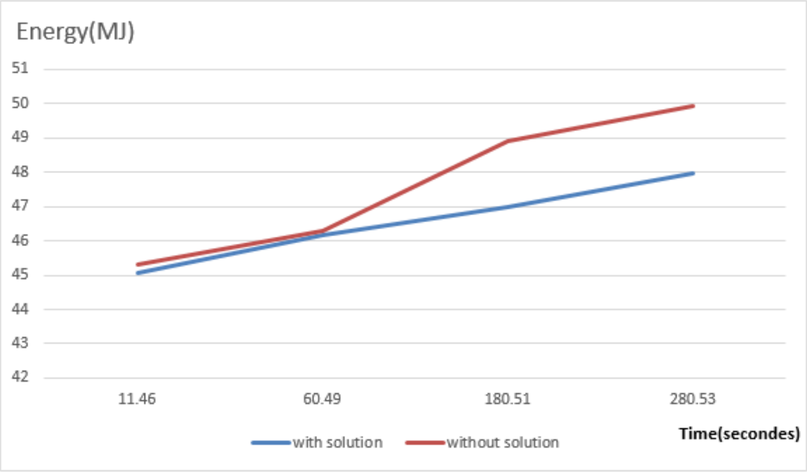


Figure 3.10: Energy Vs Time(with and without solution)

3.10.4 Energy vs Communication threshold

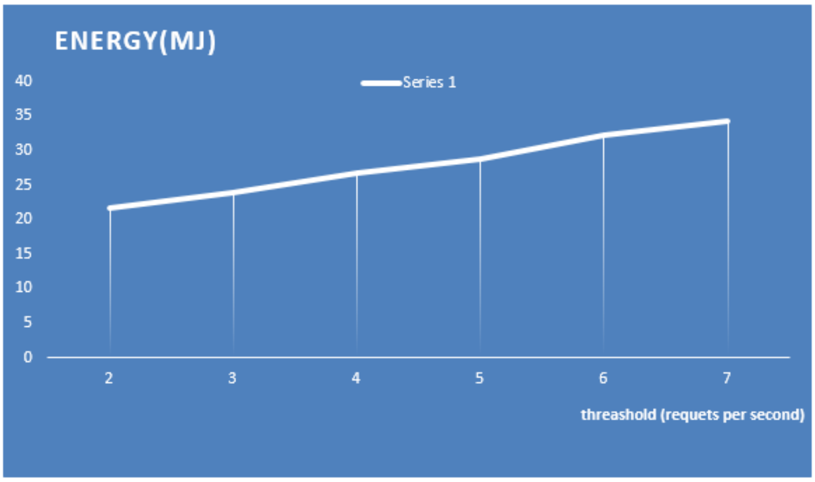


Figure 3.11: Energy Vs Communication Threshold

This figure represents the total energy consumed by one sensor node, with the solution turned on and an increasing amount of requests sent per second. We can notice that the energy consumption augments slightly each time the threshold rises up.

3.10.5 Isolation delay

The isolation delay of the compromised drone is assessed according to random and different movement (low, medium and high) speeds of that drone.

figures 3.12 3.13 3.14 and 3.15 show client (drone) movements over the network. Accordingly, we have obtained the following results:

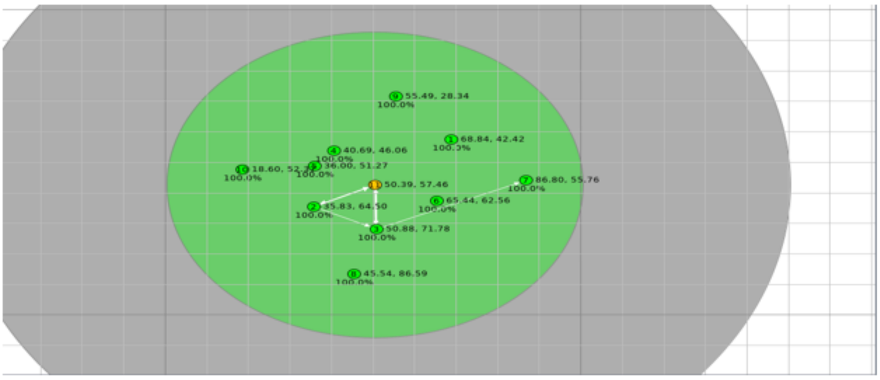


Figure 3.12: position number 1.

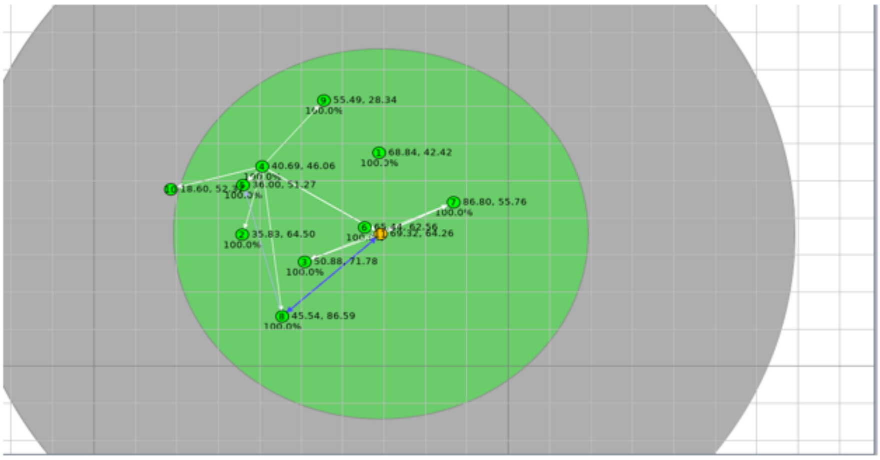


Figure 3.13: position number 2.

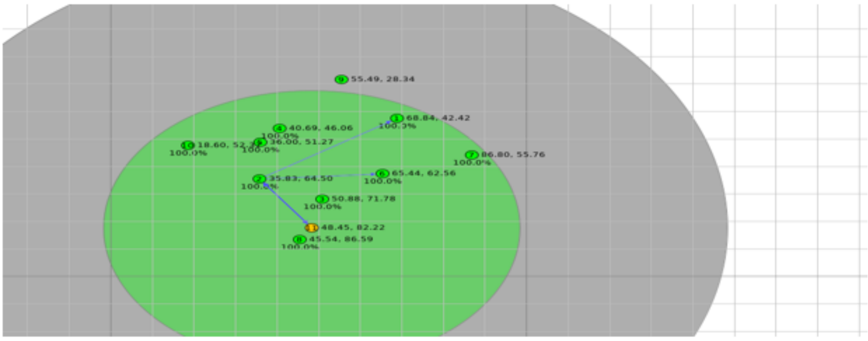


Figure 3.14: position number 3.

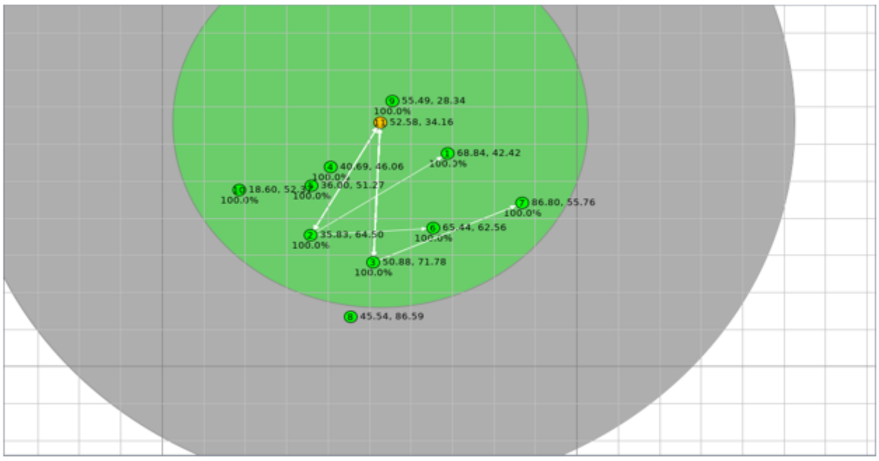


Figure 3.15: position number 4.

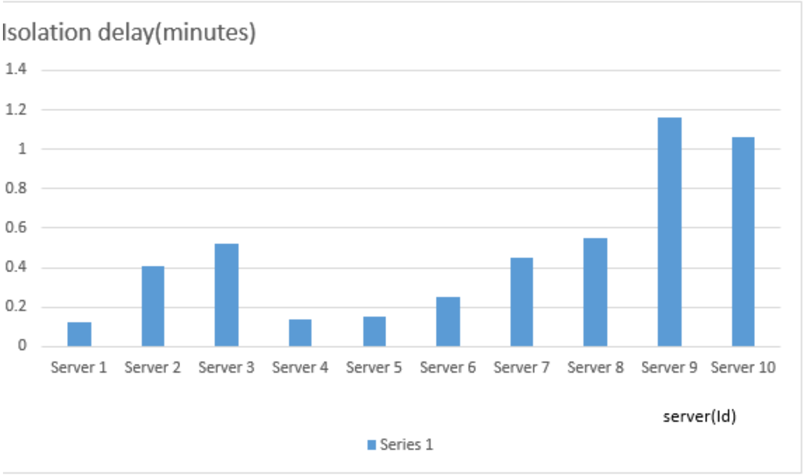


Figure 3.16: delay isolation with low mobility

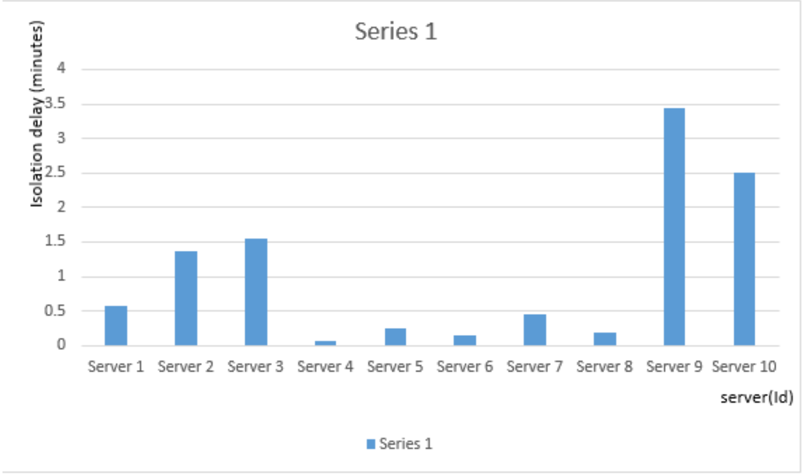


Figure 3.17: delay isolation with average mobility

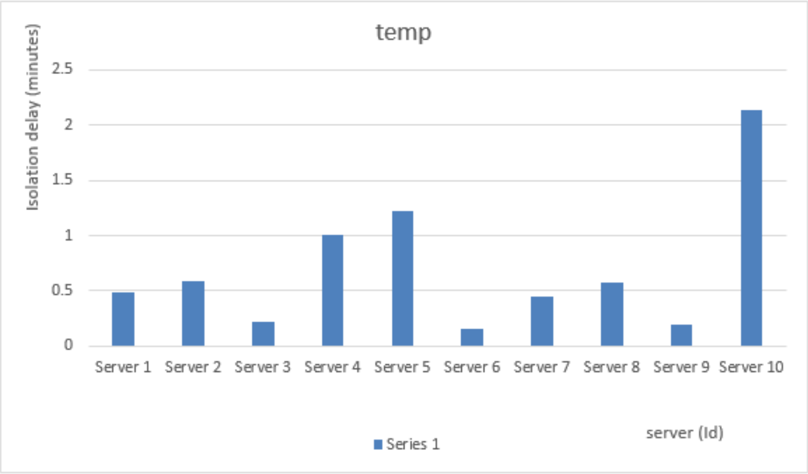


Figure 3.18: delay isolation with hight mobility

From the three last result figures, we can see that the isolation delay is relatively reduced with the majority of sensor nodes that are the most frequented by the Drone.

3.11 Conclusion

In this chapter, we have presented the essence of the realized solution and its implementation context. Then, we have presented the performance evaluation of the solution under Cooja simulator. The obtained results show that the solution brings good resiliency against DoS attacks while being suitable to IoT constraints.

General conclusion

Throughout this document, we have presented some generalities related to the area of Internet of Things. We have then, presented some basic aspects in a trustful IoT before presenting our work that consists in a trust based security policy for WSN-Done communication in the IoT. The solution relies on the dynamic and autonomous trust estimation of the connections between a DoS launcher Drone and each sensor node in the WSN.

The preliminary obtained results show that the solution is brings good DoS-resiliency for the IoT integrated WSN. However, the present work might be enhanced by the consideration of smarter attack scenarios (such as the combination of DoS with identity spoofing attacks) that can be exercised by not necessarily one unique Drone. In addition, the cooperative trust estimation among the sensor nodes might bring significant efficiency to the solution.