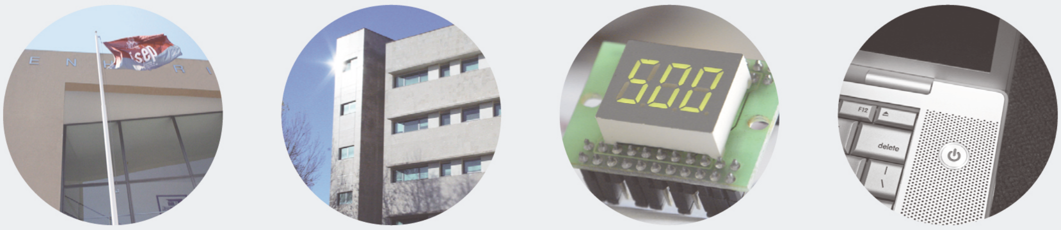
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



Explorando Oportunidades Esteganográficas sobre o IEEE 802.15.4

MAGDA ALEXANDRA OLIVEIRA PEREIRA

Novembro de 2022

POLITÉCNICO
DO PORTO

Politécnico do Porto

Instituto Superior de Engenharia do Porto

Exploring Steganographic Opportunities over the IEEE 802.15.4

Magda Alexandra Oliveira Pereira

Master in Electrical and Computer Engineering
Specialization Area of Automation and Systems

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

November 1, 2022

This dissertation partially satisfies the requirements of the Thesis/Dissertation course of the program Master in Electrical and Computer Engineering, Specialization Area of Automation and Systems.

Candidate: Magda Alexandra Oliveira Pereira, No. 1170753, 1170753@isep.ipp.pt

Scientific Guidance: Ricardo Severino, rar@isep.ipp.pt

Scientific Co-Guidance: Carlos Campos, crc@isep.ipp.pt

Research Group: PORTIC - Porto Research, Technology and Innovation Center - Polytechnic Institute of Porto

Advisor: João Rodrigues, joao.rodrigues@portic.ipp.pt

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200–072 Porto

November 1, 2022

Acknowledgements

Firstly, I would like to start by thanking Instituto Superior de Engenharia do Porto (ISEP) and all the professors that passed their knowledge to me during the last five years.

I would like to give special thanks to Dr. Ricardo Severino, Dr. Carlos Campos and João Rodrigues for all the help they have given me over the last few months to make this work possible.

To all my friends, thanks for the support provided when it was most needed, the moral strength given, the long talks and for just being there when I truly needed. Thank you all for being there!

Lastly, to my family (my parents and sister), especially my mother. It is always a little difficult to consolidate work, studies, and social life, and I know that without their support and life lessons it would be impossible to finish this stage of my life. For that and everything leading to this project, you have my eternal gratitude.

Abstract

The advancements in information and communication technology in the past decades have been converging into a new communication paradigm in which everything is expected to be interconnected with the heightened pervasiveness and ubiquity of the Internet of Things (IoT). As these technologies mature, they are increasingly finding its way into more sensitive domains in which safety and cybersecurity are paramount.

While the number of deployed IoT devices continues to increase annually, up to tens of billions of connected devices, IoT devices continue to present severe cybersecurity vulnerabilities, which are worsened by challenges such as scalability, and heterogeneity at different levels.

Steganography is the practice of representing information in a cloaked fashion, in such manner that it is not evident to a computing or communications system. These are being used increasingly to support malware with stealthy behaviours, aiming at exfiltrating data or orchestrating nodes of a botnet in a cloaked fashion. Nevertheless, the attention to this problem regarding underlying and pervasive IoT protocols such as the IEEE 802.15.4 has been scarce.

Therefore, in this thesis, we intend to explore different steganographic opportunities of the IEEE 802.15.4 protocol and to analyze their *performance*. This will enable a better understanding of the threat, and to open new pathways to further support the development of new mechanisms and add-ons, that can effectively contribute to improve the current state-of-art of IoT systems which rely on such, or similar underlying communication technologies.

Keywords: IoT, Steganography, IEEE 802.15.4, MAC, OMNeT++

Resumo

Os avanços nas tecnologias de informação e comunicação nas últimas décadas têm convergido para um novo paradigma de comunicação em que se espera que tudo esteja interligado com o aumento da omnipresença e da difusão da Internet das Coisas (IoT). À medida que estas tecnologias amadurecem, vão encontrando cada vez mais o seu caminho em domínios mais sensíveis, a proteção e a segurança cibernética são primordiais.

Embora o número de dispositivos de IoT implantados continue a aumentar anualmente, chegando a dezenas de biliões de dispositivos conectados, os dispositivos IoT continuam a apresentar graves vulnerabilidades de cibersegurança, que são agravadas por desafios como a escalabilidade, e a heterogeneidade a diferentes níveis.

A esteganografia é a prática de representar a informação de uma forma camuflada, de tal forma que não seja evidente para um sistema informático ou de comunicações. Estes estão a ser cada vez mais utilizados para dar suporte a malware com comportamentos furtivos, visando a exfiltração de dados ou a orquestração de nós de uma botnet de forma camuflada. No entanto, a atenção a este problema em relação aos protocolos de IoT subjacentes e abrangentes, tais como o IEEE 802.15.4 tem sido escasso.

Portanto, nesta tese, pretendemos explorar diferentes oportunidades esteganográficas do protocolo IEEE 802.15.4 e analisar o seu *performance*. Isto permitirá uma melhor compreensão da ameaça, e explorar novos meios para apoiar ainda mais o desenvolvimento de novos mecanismos e add-ons, que podem efectivamente contribuir para melhorar o actual estado da arte dos sistemas IoT que dependem de tais, ou de tecnologias de comunicação subjacentes semelhantes.

Palavras-Chave: IoT, Esteganografia, IEEE 802.15.4, MAC, OMNeT++

Contents

List of Figures

List of Tables

List of Acronyms

ACT	Allocation Counter Table
AMCA	Asynchronous Multi-Channel Adaptation
BI	Beacon Interval
CAP	Contention Access Period
CFP	Contention Free Period
CPS	Cyber Physical Systems
CSL	Coordinated Sampled Listening
CSMA/CA	Carrier-sense multiple access with collision avoidance
DSME	Deterministic Synchronous Multi-channel Extension
DSSS	Direct Spread Spectrum Sequence
FCF	Frame Control Field
FFD	Full Function Device
GACK	Group Acknowledgements
GIF	Graphic Interchange Format
GTS	Guaranteed Time Slots
HVS	Human Visual Systems
IAT	Inter Arrival Time
IE	Information Elements
IoT	Internet of Things
JPEG	Joint Photographic Expert Group
LLDN	Low Latency Deterministic Network
LQI	Link Quality Indication

LSB	Least Significant Bit
MAC	Medium Access Control
MCPS-SAP	MAC Common Part Sublayer
MD	Multi-superframe Duration
MLME-SAP	MAC Sublayer Management Entity
OMNeT ++	Objective Modular Network Testbed in C++
OSI	Open Systems Interconnection
P.PORTO	Polytechnic Institute of Porto
PAN	Personal Area Network
PDU	Power distribution unit
PHY	Physical Layer
PNG	Portable Network Graphics
PORTIC	Porto Research, Technology and Innovation Center
RFD	Reduced Function Device
RFID	Radio Frequency Identification
RIT	Receiver Initiated Transmissions
SAP	Service Access Points
SD	Superframe Duration
SO	Superframe Order
STOIC	Secure Trustworthy Omnipresent Cyber-defense
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TSCH	Time Slotted Channel Hopping
UDP	User Datagram Protocol
WPAN	Wireless Personal Area Network

Chapter 1

Introduction

1.1 Overview

With the increased pervasiveness and ubiquity of the Internet of Things (IoT) paradigm, information and communication technology advancements over the past several decades have been conveniently creating a new communication paradigm in which everything is expected to be interconnected. Industry 4.0 refers to the rapid advancement of technology, which results in new methods of interconnecting gadgets and systems, new data insights, customizable goods, and technical autonomy. In order to enable what is now known as Industry 4.0, which integrates IoT, Cyber Physical Systems (CPS), and Cloud technologies into the factory floor, these technologies are progressively making their way into the industrial sector as they develop. Industry 4.0 refers to the rapid advancement of technology, which results in new methods of interconnecting gadgets and systems, new data insights, customizable goods, and technical autonomy. Since its inception, it has been used to characterize a significant digital manufacturing shift where networked equipment and processes enable mass customization of goods and quicker market reaction [Masood and Sonntag 2020]. Researchers concur that projects like Industry 4.0 and digital transformation are essential for a company's ability to compete in the future and successfully meet consumer expectations. This is crucial for SMEs, which account for 99 percent of all businesses in Europe.

Although the number of deployed IoT devices continues to increase annually and is estimated to reach 75 billion by 2025 [Statista 2022], as can be seen in the figure 1.1, the amount of interconnected devices follows the same trend. Fueled by the large IoT scale, heterogeneity, and its fast adoption, beyond business opportunities, this fact also unlocks a variety of new security threats [Lin et al. 2017, Frustaci et al. 2018, "A Novel Timing-based Network Covert Channel Detection Method" 2019], with heightened security and privacy risks, particularly in industrial and medical scenarios [Liu et al. 2020].

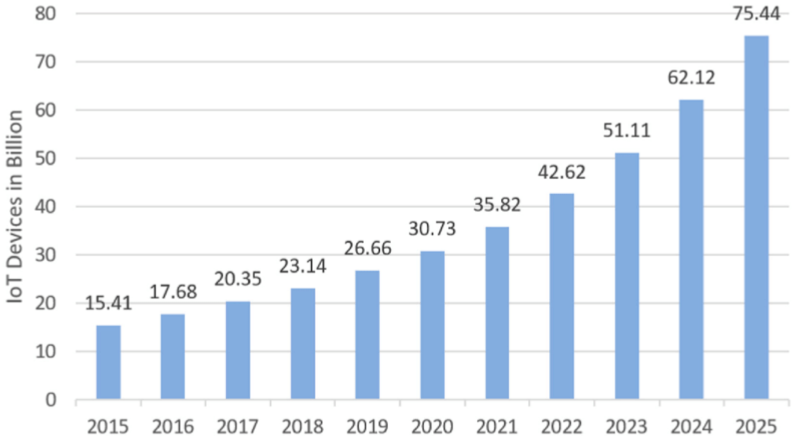


Figure 1.1: IoT Devices Statistics by 2025 [Statista 2022]

IoT risks and vulnerabilities are caused by the enormous number of intrusion targets that the IoT paradigm has generated (some of this vulnerabilities can be seen in figure 1.2), as well as the systems' inherent greater susceptibility when they are coupled to other systems (such as legacy systems connected to Internet gateways) (e.g., home connected appliances, wearables, connected vehicles, etc.). With this scenario, we must consider that each compromised system may be used for at least three different malicious purposes. For example, with regard to Industry 4.0 sensors and appliances, devices may be used for leaking sensitive data (for example, to steal secrets and engage in industrial espionage), endangering a plant physically (for example, by saturating the processing power of nodes controlling machinery), or mounting an attack to more critical infrastructure (e.g., enrolling the compromised node into a remotely controlled botnet) [Caviglione, Merlo, and Migliardi 2018].

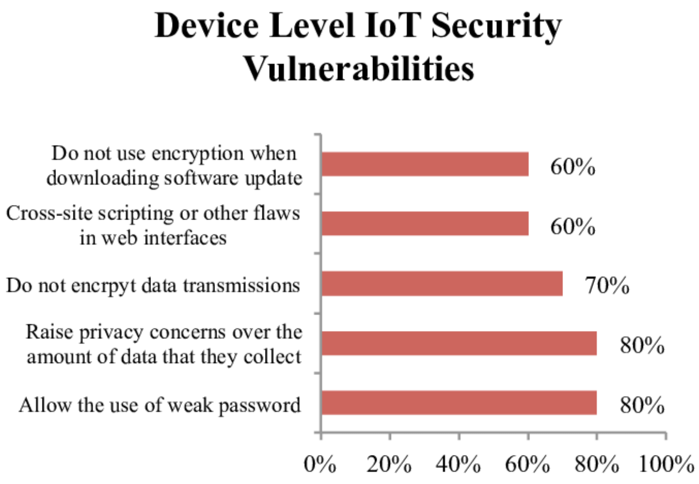


Figure 1.2: IoT devices vulnerabilities [Gamundani, Phillips, and Muyingi 2018]

Steganography can be effectively leveraged to support most of these attacks, therefore the capacity of the attacker to utilize a covert channel and information concealment techniques is depicted as a very significant factor in any assault in such a problematic scenario, especially when physical access to the IoT infrastructure is conceivable.

No matter if the goal is exfiltration of critical information or if it is to induce unintended behavior of a node, there exists a need for communicating with the compromised node without disclosing the fact that it has been compromised. Indeed, network covert channels are increasingly being used to support malware with stealthy behaviours (stegomalware), for instance to exfiltrate data or to orchestrate nodes of a botnet in a cloaked fashion [Caviglione 2021a]. However, the detection of such attacks is difficult as it is unknown in advance where the secret information has been hidden, and on the other hand, network covert channels usually feature low data-rates which complicates the detection. Also, neutralization or mitigation is not straightforward, as it is hard not to disrupt legitimate flows or degrade the quality of service, particularly at the perception layer of an IoT application. Consequently, countermeasures are tightly coupled to specific channel architectures, leading to poorly generalized and often scarcely scalable approaches.

Therefore, it is quite relevant to explore steganography opportunities in such common communications protocols. This Thesis aims not only at accomplishing this, but also at disclosing the threat such implementations can impose, by analyzing their data throughput under different network scenarios.

To this end, this Thesis begins by offering the IoT and cyber-security communities a deployment - and analysis - friendly simulation model that includes different implementations of hidden network storage channels. This implementation is then used as a benchmark to compare the effectiveness of the various covert channel approaches in various networking contexts. This understanding will be essential for better understanding the risk, and to support new mechanisms that may make use of these covert strategies in benefic ways, putting innovative information security concepts into practice.

1.2 Research Context

This work was carried out in alignment with the Secure Trustworthy Omnipresent Cyber-defense (STOIC) research framework and supported by the CybersSecIP project, at the Porto Research, Technology and Innovation Center (PORTIC) of the Polytechnic Institute of Porto (P.PORTO). The STOIC framework aims at developing innovative mechanisms and technologies for achieving improved cyber-security in distributed wireless infrastructures. This research is tightly connected with the work being pursued in the CybersSecIP project, which aims at further strengthening the scientific competences and innovation potential of the North region of the country, to tackle the cyber-security challenge, through investment in a small set of enabling technologies and knowledge, in a coherent program organized in two research lines: one related to the design and protection of secure digital systems, and a second centered on data security and privacy.

1.3 Objectives

1. Understand the fundamentals of IoT communications, major IoT cybersecurity threats and their impact in such networks, particularly the risk of covert channels.

2. Realize and present an argumentation about the severity of the problem addressed in the proposal in regards to IoT cybersecurity.

3. Survey covert channel techniques and overview current state of the art. Compare different approaches.

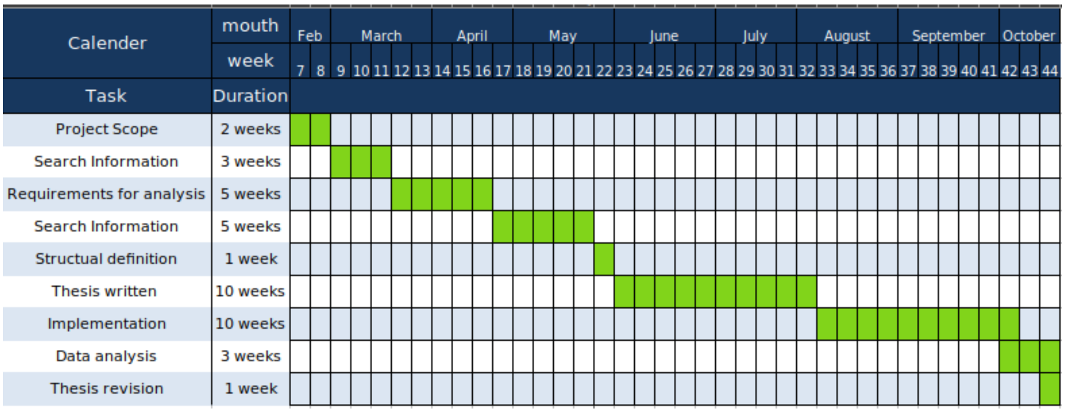
4. Overview the IEEE 802.15.4-2020 standard and investigate new covert storage channel opportunities to exploit in the protocol.

5. Carryout a preliminary simulation analysis of different covert-channel techniques.

6. Implement different covert-channel techniques over a IEEE 802.15.4 communications stack and evaluate the performance and ease of construction of different techniques in the IEEE 802.15.4.

1.4 Scheduling

The project plan, can be observed in figure 1.3, where is visible all the tasks, planned for the study during the project.



Calender	mouth	Feb		March					April				May					June					July				August					September					October		
	week	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Task	Duration																																						
Project Scope	2 weeks	🟩	🟩																																				
Search Information	3 weeks			🟩	🟩	🟩																																	
Requirements for analysis	5 weeks						🟩	🟩	🟩	🟩	🟩																												
Search Information	5 weeks											🟩	🟩	🟩	🟩	🟩																							
Structual definition	1 week																🟩																						
Thesis written	10 weeks																	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩												
Implementation	10 weeks																											🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
Data analysis	3 weeks																																				🟩	🟩	🟩
Thesis revision	1 week																																						🟩

Figure 1.3: Thesis Schedule

1.5 Structure of the Dissertation

This Thesis is composed of introduction (chapter 1), where the motivation and context is presented. Next, it provides a research background,where covert channels as a concept are introduced, along with their history and relevant research works (chapter 2), particularly those which focus the usage of Steganography. This is followed by an overview of the IEEE 802.15.4 communication protocol in the next chapter (chapter 3). In the next chapter it will be discussed in more detail what stenography is and the opportunities it can have as well as dangers (chapter 4). Next, this Thesis introduces the simulation tools used in this Thesis (chapter 5). This chapter presents all the simulation tools and the purpose throughout this work. The final architecture of the solution is then explained together with possible design alternatives. Finally, the results and the performance analysis this study will be shown detailed in chapter 6. In here, a brief description of the simulation setup appears, along with the obtained results and its analysis. In chapter 7 it will have featuring several observations and conclusions of the performance of the simulations. Finally, in chapter , several general conclusions are drawn from the obtained results, and a few ideas are proposed for future work.

Chapter 2

Research Background

In this chapter introduces the concepts of steganography and covert communication channels.

2.1 Steganography

2.1.1 History of Steganography

While traditional encryption focuses on hiding the content of messages, steganography hides the very fact that messages even exist [Anderson and Petitcolas 1998]. Steganography is an old method that has been used for centuries. A message hidden behind the wax of a writing tablet was used by the Greeks to send and receive warnings of enemy movements, according to Herodotus, a historian who lived in the fifth century B.C.. Other instances were the employment of covert ink to hide information on white paper or the use of micro-dots by intelligence agents during World War 2. The phrase *"cover what I write"* or, more simply put, *"hide data"* is the direct translation of the Greek terms *"stegano"*, which means *"I cover"*, and *"graphô"*, which means *"I write"* [Martins and Guyennet 2010]. The purpose of traditional steganography is to conceal a secret message—such as a copyright mark, covert communication, or serial number—in a cover message that is disguised as a video or audio file, computer code, or other kind of media. The embedding is often parametrized by a key, making it challenging for a third party to find or remove the embedded material without knowledge of this key (or a related one). A cover object is referred to be a stego object if material has been embedded in it, therefore, figure 2.1 is a good example of this, as the difference between the original and the hidden image is 2 bits, giving the illusion that there is no change in the message [Anderson and Petitcolas 1998].

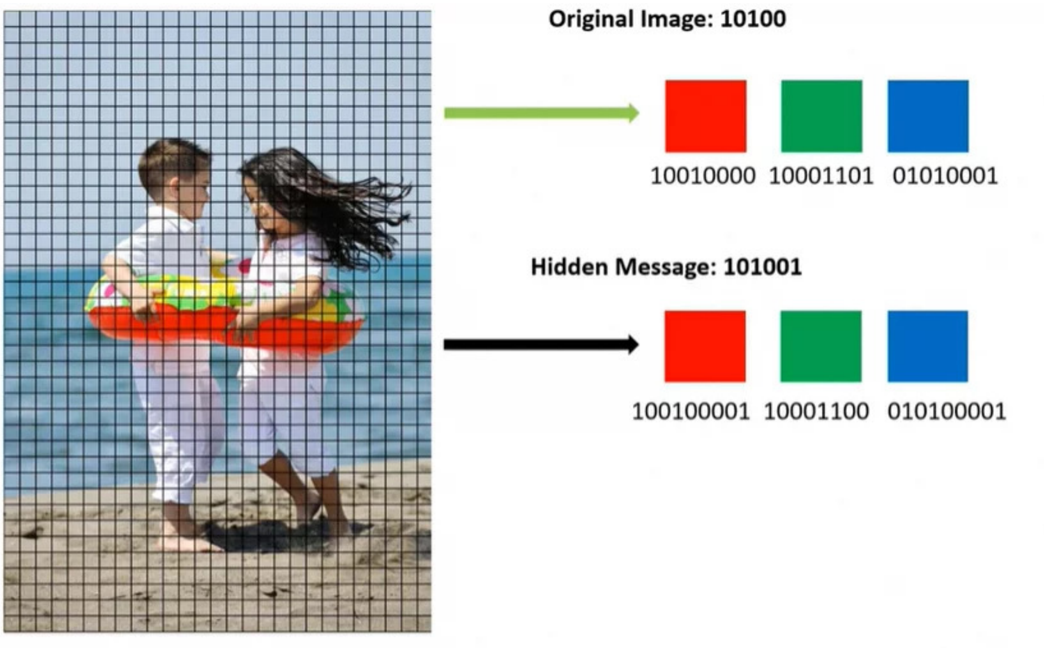


Figure 2.1: Steganography Example [VOTIRO n.d.]

Steganography should not be confused with cryptography, in which the message is changed to make it difficult for someone to decipher its contents. Such defense is frequently insufficient. A recent UK police force concerned about criminal monitoring of police radios has discovered that it is not enough to simply encipher the traffic, as criminals detect, and react to, the presence of encrypted communications nearby. The detection of encrypted message traffic between a soldier and a hostile government, or between a known drug smuggler and someone not yet under suspicion, has obvious implications. In some circumstances, it suffices to conceal the sender or recipient of the communication rather than the message itself. Criminals frequently believe that the anonymity of the caller is adequate. Traffic selection-choosing which calls to intercept is, in fact, the primary practical challenge confronting law enforcement and intelligence organizations, and due to the enormous volume of traffic, this must often be done in real time [Anderson and Petitcolas 1998].

The Prisoners' Problem's paradigm of classical steganography and copyright marking are very different from one another. A successful attack in the former requires the warden to notice that a specific object is marked. This paradigm will be explained further in the next section. In the second scenario, all participants in the scheme may be aware that marks are being used, making the impacts of the marks visible (marks should stay below the perceptual threshold, but they may change the statistics of the material in ways that are readily measured). Therefore, an effective attack does not involve finding a mark but rather making it useless. To prevent a judge from determining whether mark was genuine, this may be accomplished by either erasing it or adding several other markings. A customer's signature included in the document or the use of a public timestamping service during the marking process might also help to prevent such assaults [Anderson and Petitcolas 1998].

Steganalysis is the process of identifying and often decoding concealed data inside a particular medium. It is the recognized countermeasure against steganography. Information theory and statistical analysis, two key steganalysis methods, clearly demonstrate the enormous potential for hidden information in Internet data. As long as a collection of data can be compressed to a lower size, hidden data can exist inside the medium. As a result, the path to uncovering buried information is perilous and unclear. It may be very hard to discover concealed data in the carrier data unless it is encoded in a standard, well-defined format. In other words, how one chooses to specify the encoding, makes the whole difference. A bit is merely a bit to a steganalysis specialist who cannot identify the selected encoding [Artz 2001].

2.1.2 Steganography algorithms

Steganography algorithms can be classified into various categories as shown in Figure 2.2 based on factors such as the type of cover object used, the type of domain (spatial or transform domain), the type of file format or compression used, and the type of embedding method used to modify the cover object [Kaur, S. Bansal, and R. K. Bansal 2014].

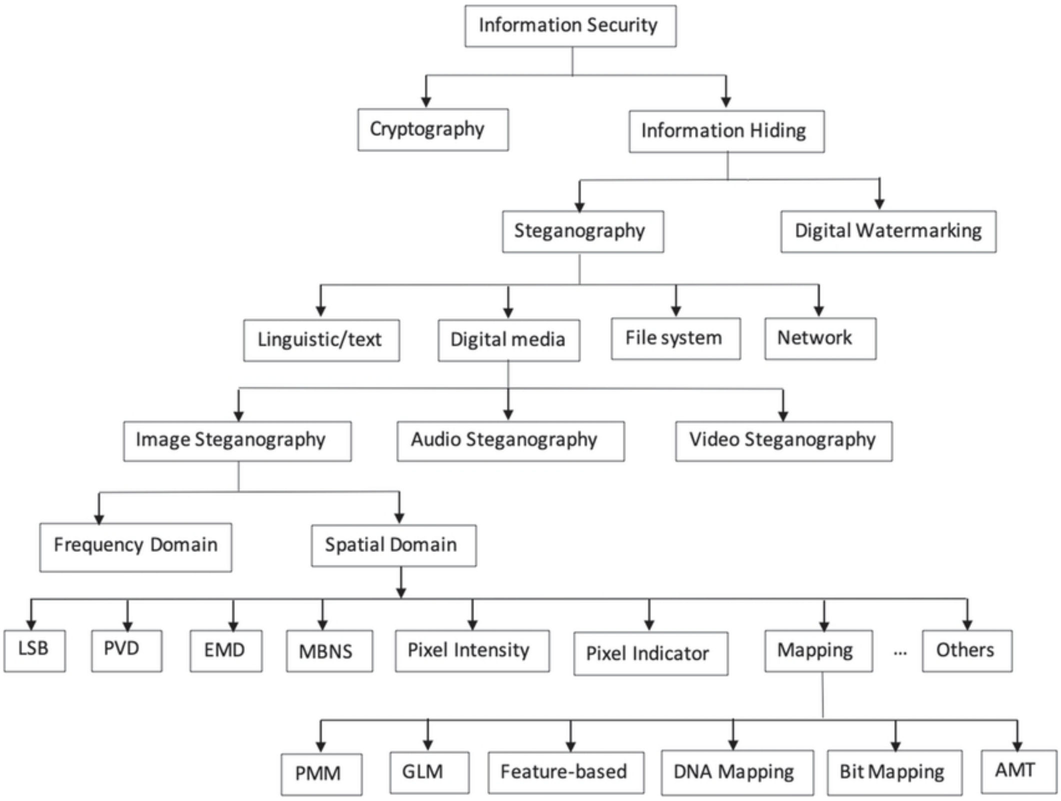


Figure 2.2: Classification of various steganography techniques [Elshoush, Mahmoud, and Altigani 2022]

Before explaining the different types of steganography it is necessary to understand the requirements for a steganograpy algorithms. They are the capacity, in other words the volume of data that has to be embedded in a cover medium and that can be properly retrieved later without the cover medium being significantly changed. The undetectabily, as there should be no obvious difference between a stego and a cover object. A stego system needs to bear any attack. For instance, in the case of a digital image, if it undergoes transformation such as rotation, lossy compression, scaling, etc. it should remain intact, and this is robustness. Finally, the security is fundamental because an embedding algorithm can only be deemed secure if the embedded information cannot be easily removed or changed if the attacker detects it.

As represented in the figure 2.2 there are different types of steganography. In the following section it will be explain the most well-known.

According to the Cover Object

Digital Media

This category is divided into 3 subcategories of steganography types which are [Kaur, S. Bansal, and R. K. Bansal 2014]:

- **Image**

  The most often used type of steganography is image steganography. Here, a hidden message is embedded in noise that can hardly be seen by the naked eye. Data hidden in static images offers certain difficulties for Human Visual Systems (HVS) to handle. Still pictures are also vulnerable to a variety of processes, ranging from straightforward to complex nonlinear transformations including cropping, blurring, filtering, and lossy compression, among others. Data concealing techniques should be resilient to these kinds of operations. As computer graphics capability increases, images will continue to be a popular medium on the internet. Additionally, images have a high degree of redundancy, offer more capacity, and can tolerate more distortion. There are currently a lot of applications available that use picture steganography to conceal text as steganography tools.

- **Audio**

  The message is concealed in a cover audio track using audio steganography at a frequency that is unheard by humans. Secret message encoding in digital sound is typically more challenging than in other forms of media, and it is more sensitive to additive noise. Least Significant Bit (LSB) coding, parity coding, phase coding, spread spectrum, and echo concealing are often employed techniques for audio steganography.

- **Video**

  In video files, which are often collections of sound and pictures, information is concealed. Video files may use the same steganography techniques as music and picture files. The benefit of this strategy is that due to the continual flow of information, a significant quantity of data may be concealed inside videos with less distortion—data that may go unnoticed by observers.

Linguistic/Text

It is among the oldest and trickiest varieties of steganography. It is a technique for hiding a secret message in written natural language. Due to the fact that text documents have less redundancy than images and audio files, text steganography is the most difficult [Kaur, S. Bansal, and R. K. Bansal 2014].

Network

It has to do with hiding information in fields of network control protocols that are optional or empty during transmission over a network. There are covert channels in the OSI network model's layers where steganography can be applied. In several fields that are either optional or never utilized in the header of a Transmission Control

Protocol/Internet Protocol (TCP/IP) packet, information can also be concealed. Humans read some fields in headers only occasionally, so these fields are an ideal location for data hiding. The disadvantage of data hiding in headers is that for safety reasons, we often configure firewalls to filter out packets where reserved fields contain unusual information, and thus hidden information may also get lost.

According to the Domain Type

In accordance to the domain type, most used steganography techniques are either on the spatial domain and the transform domain [Kaur, S. Bansal, and R. K. Bansal 2014].

Spatial Domain Techniques

Bitwise pixel intensity modification and noise manipulation are examples of methods used in the spatial domain. Data may be embedded in the spatial domain using a variety of methods. LSB methods are the most popular and basic methods for spatial domain. This method is based on the replacement of the least significant bits of cover object with a secret message. It is most popular and simple technique when dealing with images. It has low computational complexity and high embedding capacity. Since the change's amplitude is small, modifying the LSB does not produce a difference that is audible to humans. As a consequence, the final stego-image will appear to the human eye to be an exact replica of the cover-image. High perceptual transparency of LSB is made possible by this. Although it is a fairly straightforward approach, stego-images can be destroyed by noise or lossy compression in addition to lossy compression and picture modification including scaling, rotation, cropping, etc. The picture should be grayscale with progressive variations in shades, and the image file should be larger than the message file for it to operate optimally. Both fixed type and variable bit LSBs are possible [Kaur, S. Bansal, and R. K. Bansal 2014].

Transform Domain Techniques

Frequency domain methods are another name for transform domain techniques. In order to insert a hidden message, transform domain techniques first convert an image from the spatial domain to the frequency domain. These methods employ mathematical operations to conceal the facts. These methods of concealing a hidden message in the transform space of the cover object are frequently used in compression algorithms. Secret data is embedded into transform coefficients in frequency domain schemes after being converted into the frequency domain by a variety of methods, including discrete cosine transformation, discrete wavelet transformation, discrete fourier transformation, etc. [Kaur, S. Bansal, and R. K. Bansal 2014].

According to the File Format and type of Compression used

The most used images on internet are Graphic Interchange Format (GIF), Joint Photographic Expert Group (JPEG) and Portable Network Graphics (PNG) and most of the steganography techniques exploit these image formats and some of the techniques are based on Bitmap format (BMP). Image compression is used to store and transmit huge image files in an acceptable period of time. Lossy and lossless compression are the two different types of picture compression techniques. Both methods are space-saving, but they have differing effects on buried, uncompressed data. Understanding the compression and type of compression utilized in the cover object is crucial when using steganography. The file sizes of uncompressed formats (GIF and BMP) are greater than those of any other format and they have more visual redundancy, which allows them to hold more hidden data and makes them more useful for data concealing methods [Kaur, S. Bansal, and R. K. Bansal 2014].

Embedding Methods

Here are some of the often employed techniques based on certain approaches to alter the cover object to conceal hidden data [Kaur, S. Bansal, and R. K. Bansal 2014]:

- Spread Spectrum: This method, which spreads a narrow band signal's bandwidth, is based on spread spectrum communication. Spread spectrum steganography generates a stego image by combining a cover image with a secret message that is first embedded in noise. The power of the embedded signal is substantially lower than that of the cover image, and the resulting stego image cannot be seen by HVS.

- Masking: By changing the luminance of particular regions, this technique covers original data with secret data. It incorporates the message within significant bits of the cover image. Masking, unlike LSB, is resistant to lossy techniques because it provides redundancy to the hidden information, which protects it from being affected by image modification. As a result, the masking technique is more suited for lossy JPEG images than LSB. Additionally, it could aid in safeguarding against some image processing procedures like cropping and rotating.

- Statistical: Here, data extraction and hiding are based on certain statistical characteristics of the cover object. It makes use of the fact that "1-bit" steganography is possible and modifies the cover in such a manner that "1" is transferred by altering some of the statistical characteristics of the cover; otherwise, the cover is left unaltered.

- Distortion: Here, a secret message is concealed via cover distortion and measurement of the difference between the original cover and the stegos during the decoding process. Due to the possibility that a steganalyst may have access to the original cover object for comparison, distortion methods are less secure and are rarely employed in many situations. Techniques for text-based steganography often utilize distortion type for embedding.

2.1.3 Limitations

The same presumption that applies to encryption also limits steganography. Alice and Bob must first privately decide on a steganography technique before Alice may send Bob an image containing a secret message. In accordance with the encryption concept, Bob may be reasonably certain that he has some ciphertext. But in the steganography paradigm, Bob won't always be able to tell when a picture is merely an image.

The size of the medium itself usually limits the amount of data that can be properly hidden in it. Also, the opportunity for concealing data increases with the lack of restrictions on the medium's integrity [Artz 2001].

2.2 Network Covert Channels

2.2.1 Principles and History of Covert Channels

The meaning of covert channel can be explained by an attack or evasion technique that is used to transfer information in a unauthorized, illicit or secretive manner. A covert channel can also be used to extract information from or implant information into an organization [Piscitello 2016].

The term of covert channels were introduced by Lampson's "Note on the Confinement Problem" but the meaning was restricted to a subclass of leakage channels that exclude storage channels and “legitimate” channels [Millen 1999]. A covert channel can be also described as "any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy", like it is said in [“Department of Defense Trusted Computer System Evaluation Criteria” 1985]. Basically, it is a method of communication used to transfer information that normally is not allowed to be access [Rowland 1997].

The best example to understand what is a covert channel, is to rely on the example normally used to explain cryptography, the Alice and Bob example. Alice and Bob are two prisoners in different cells and both have a single communication method that is the transmission one to another, but every message is analysed by a warden (Figure 2.3). Both of them want to escape the prison so they must communicate with each other and the warden can't know what is truly being said between them. So they must establishing a "subliminal channel" between them (this can be called steganography, as explained above). In conclusion, Alice and Bob have created a covert channel to communicate in order to escape from the prison [Simmons 1984].

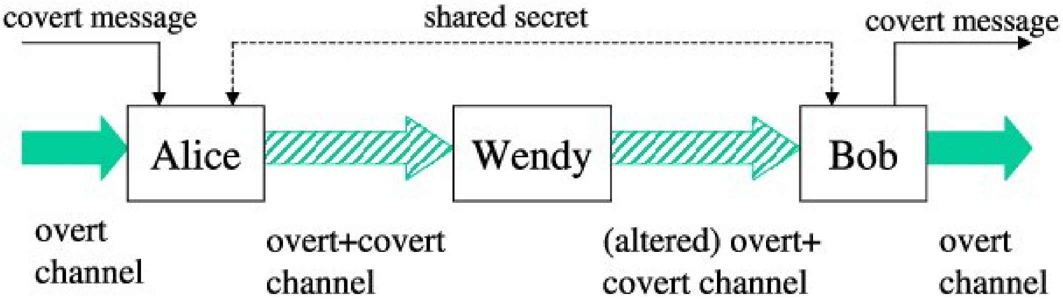


Figure 2.3: Alice and Bob example [Zander, Armitage, and Branch 2007]

2.2.2 Covert Channels Characteristics

Coverts Channels have a common set of characteristics and metrics that aim to establish discrete and differential, distinguish and assess various implementations of covert channels. Some characteristics are shown in the following topics [Johnson et al. 2010]:

- Mechanism: specifies how the channel is constructed and how it carries the information.
- Type: classify the cover channels by a type.
- Throughput: measures the amount of data the channel is able to transmit over a certain period of time.
- Robustness: this characteristic describes the ability of the covert channel to persist in different circumstances (store-and-forward devices, firewalls, proxy servers or other similar devices).
- Detection: when measuring covert channels the probability of detection is essential, however for a large number of covert channels, values for probability of detection can be derived from the entropy exhibited by the covert channel.
- Prevention: is the ability to take explicit actions (user or administrator intervention) to disrupt or degrade the covert channel.

2.2.3 Storage Covert Channel Techniques

Without anyone in the network (such as a warden) being aware that certain data was covertly sent from one side to the other, information might be transmitted from a sender to a recipient, figure 2.4 is an example of this. By examining the network's protocol standard, various reserved fields may be discovered. Using some strategies, that same storage covert channel can be optimized to ensure that the most data can be sent covertly.

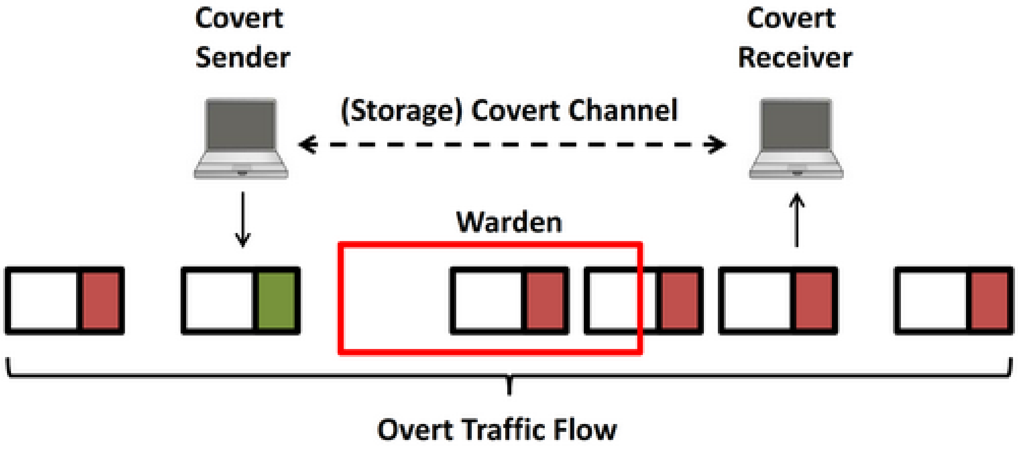


Figure 2.4: Storage Covert Channel [Caviglione 2021b]

As it explained in "A Pattern-based Survey and Categorization of Network Covert Channel Techniques" [Wendzel et al. 2015] the classification of covert channels is mane into several different patterns that these were obtained from a selection of 109 covert channel techniques:

Size Modulation Pattern

The hidden message is incorporated into the size of a header element or Power distribution unit (PDU), with paddings or offsets applied to adjust the overall size of a given element, with different sizes (or size differences) being assigned to readable information.

Sequence Pattern

In order to encode covert data, the covert channel modifies the header's or PDU's element order. This can be done by changing the position of one (or more) of the header's elements, the quantity of these elements in the header, or the PDU.

Add Redundancy Pattern

These kinds of techniques create (or enlarge) additional header elements to provide more empty space that may be used to covertly transfer information.

PDU Corruption/Loss Pattern

The system creates corrupted PDUs that are embedded with covert information or drops network packets to signal the hidden information in order to send covert information. Another method involves dropping a predetermined number of network packets in order to provide a set packet loss rate that is covert meaning to the receiver.

Random Value Pattern

After the system generates a random value, the covert channel checks for a header in the packet that has that value and replaces it with the information that will be covertly transmitted.

Value Modulation Pattern

The covert channel will modulate the values of a specific element in a packet's header, changing it to a value between 0 and the maximum it might just contain. One technique used in this pattern is case-modification, where a letter's case (upper or lower) in a packet header element might infer concealed significance if the network warden overlooked it.

Reserved/Unused Pattern

Many elements constitute a packet header, some of which are reserved or unusable in a particular network or protocol. This implies that any information present in those fields will be overlooked by the regular channel. This creates a chance to insert hidden data into those exact fields.

2.2.4 Timing Covert Channel Techniques

Secret messages are incorporated into the timing behavior at the sender via covert timing channels, which are subsequently extracted at the receiver, as is seen in figure 2.5. Network packet delays are typically utilized to transmit covert messages [Tian et al. 2020].

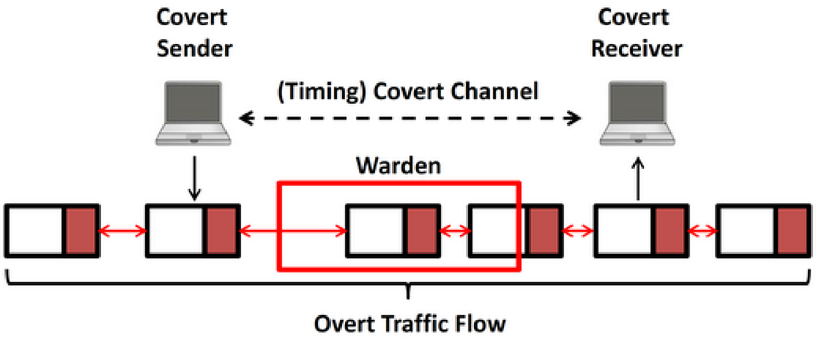


Figure 2.5: Timing Covert Channel Scheme [Caviglione 2021b]

On-Off

A relatively basic covert channel known as the "On-Off" was created by the Institute of Physics Publishing [“A Novel Timing-based Network Covert Channel Detection Method” 2019, where the key to the information shared is a timing interval Tc pre-negotiated between the sender and receiver. The receiver will interpret the jitter as a 1 bit if the sending delay is Tc. Bit 0 will indicate that the packet was dispatched without waiting the required Tc.

L-bits to N-packets

Another method is the "L-bits to N-packets" technique, which has been utilized frequently in scientific publications throughout the years [“A Novel Timing-based Network Covert Channel Detection Method” 2019].With this one, L <= N <= Channel length, the user is able to send L bits hidden in N packets.

Jitterbug

The "Jitterbug" [“A Novel Timing-based Network Covert Channel Detection Method” 2019] technique works by adding real latency to the packets that are being sent across the channel. To adjust the final latency to the desired one, this extra latency should be calculated given the previously existing network normal latency. If the receiver side’s channel latency can be divided by a predetermined W value, then it means the bit 0. On the other hand, if is divisible by W/2, then it means the bit 1.

Time Replay

This is a method for covertly transmitting information using time intervals. The latency and the packet interarrival timings are crucial to this method, just as they are to other methods. The values of the potential time intervals, therefore, were divided into two lists to add an additional layer of undetectability. As a result, when the delay was a value included in the S0 list, the covert bit was 0, and when it was present in the S1 list, the bit sent was 1 [“A Novel Timing-based Network Covert Channel Detection Method” 2019].

Inter Arrival Time

A network receiver node might receive some covert information by using the Inter Arrival Time (IAT) between packets.Successfully authenticated the node providing the information and, as a result, made the information transmitted over the typical channel trustworthy through a covert timing channel based on the IAT between packets.With the aid of silence bits, which are sent in an equal number at the start and end of the authentication frame and serve to encapsulate the relevant message in the middle, the authentication message is sent in the exact middle of the authentication frame using this method [“TACAN: Transmitter Authentication through Covert Channels in Controller Area Networks” 2019].

Packet Length

It is also possible to translate the possible variability of the packet lengths being transmitted through a network into covert information. An additional piece of information is purposelessly added to the payload after determining the weight of the packet header and the existing payload to alter the full packet length [“Covert timing channel detection method based on time interval and payload length analysis” 2020]. When receiving the packet, the receiver will know its length. If the length is even, the covert bit will be set to zero.If the length is an odd number, the bit is 1 [Elsadig and Fadlalla 2018].

L-Bits to N-packets adaptive

This method was developed based on the "L-Bits to N-Packets" technique. Before any covert information being transmitted, an analysis of the network is performed in order to get the delay values of the channel and, after them being analysed, they are split into two lists (much like in the Time Replay technique). These delays are then applied to the packets and, according to the list they are placed in, they transmit different information. The analysis performed beforehand allows for an extra layer of undetectability for all the used delay values will be congruent with the regular delays of the network [Tahmasbi, Moghim, and Mahdavi 2016].

Packet Sequence Number

The "sequence number" field in the packet header indicates the order in which a series of sequential packets will be sent. If this order changes to 1,2,4,3 instead of 1,2,3,4 at the receiver node, the packet receiver may interpret this differently [Zhang et al. 2019]. This method is used deliberately tp encrypt the sequence of data by purposely altering it in the emitting process to make sure that the order won't be the same in the other node [“Covert Timing Channels for IoT over Mobile Networks” 2018].

Bit-Rate

A channel's performance may be evaluated using the metric of bit-rate. It translates the amount of bits a channel can send in a given amount of time. A channel is more efficient than another if it has a higher bit rate than the other. Variations in this field can be considered as normal because this bit rate might depend on a variety of factors, some of which may be hardware related. With the advantage of this feature was implemented a covert channel that would mean things different to the network's receiver node depending on whether the bit rate of the channel was changed to a value higher or less than the pre-accorded one [“Covert Timing Channels for IoT over Mobile Networks” 2018].

Packet Loss

A network may suffer a variety of difficulties that may have an impact on the transmission quality. One of these consequences is packet loss, which occurs when a packet (or a series of packets) is transmitted through the network but not received by the receiver node, resulting in the packets being lost. It really is required to modify some packets so that they are supposed to be "lost" in the network in order to create a covert channel from the number (or percentage) of lost packets. The receiver node will be on the lookout for inconsistencies on the sequence and will interpret the change in there in light of that and the "sequence number" stated above [“Covert Timing Channels for IoT over Mobile Networks” 2018].

Inter-Arrival Time Probabilistic

The last technique is based on probabilistic methodologies. Using a formula based on a matrix and its values, an array of bits is encoded. Then, this encrypted value is further shaped to assign it an of inter arrival time and impose a delay between two packets. Using probabilistic methods, the receiver node will then de-shape this delay to obtain the information's encrypted value. A decryption will occur with the help of the previous matrix, exposing the original information meant for transmission [“A Timing Channel Spyware for the CSMA/CA Protocol” 2013].

2.2.5 Covert Channels in 802.15.4 Protocol

It has been attempted in the past to modify the 802.15.4 protocol to include covert channels. At the physical layer, for instance, it has been developed a covert channel Direct Spread Spectrum Sequence (DSSS) variant of steganography, where they also provide a covert acknowledgement and error detection technique to guarantee trustworthiness in the covert channel [Nain and Rajalakshmi 2017]. In the end, they came to the conclusion that they could reliably transfer confidential information using their technology at a significant rate without adversely affecting primary data reception.

A covert channel based on link quality was proposed by the authors of "Covert channel attacks in pervasive computing" [Tuptuk and Hailes 2015]. The Link Quality Indication (LQI), a metric used in the 802.15.4 protocol, is frequently used in low-power radio devices to give a measure of signal strength. Two covert channel-based attacks—modulation of transmission power and modulation of sensor data—are developed in this study. They were deemed successful ways for surreptitiously obtaining information from an exposed system after a study of the data obtained from the attacks execution.

The disadvantage of such approaches is that physical layer access and control are necessary, which is frequently not possible because these features are deeply integrated into the firmware of the radio transceiver. In this context, covert techniques that operate at the MAC sub-layer are far more attractive, especially since they are frequently implemented by a software stack that an application can much easily gain access to.

In the 802.15.4 protocol, numerous types of frames exist that may be altered to incorporate some secret information to pass along. For instance, a typical data frame has several loopholes, including the Frame Control field that contains 16 bits, each of which has a distinct meaning. There are 16 fields total, but only 5 (fields 7-9 and 12–13) are reserved, which means that information can be passed covertly in those fields even if they do not carry information and are disregarded by the computer [“Steganography in MAC layers of 802.15.4 protocol for securing wireless sensor networks” 2010].

The field for Sequence Number is another example. Each packet is numbered there so that it can be recognized in the acknowledgement message. This field, which can include 8 bits of hidden information, is not initialized during transmission in the case that the packet is signaled as not being a part of a sequence. It is instead allocated solely after reception. [Martins and Guyennet 2010].

The Address Info field comes last. This holds information about the packet’s source and destination, including addresses and PAN coordinator references. If a packet’s source address is declared as being nonexistent, the source address field is expressly not examined or used, but its empty space remains. It is possible to transmit significant amounts of covert data between source and destination nodes using this space, which ranges in size from 16 bits (short) to 64 bits (extended)[Martins and Guyennet 2010].

Some of the same fields are applicable to different frame types in the protocol. The Sequence Number field, for example, can still be left empty and utilized covertly for a beacon frame. This Sequence Number field on an acknowledgement frame as well as the Frame Control field, which was previously mentioned before, are applicable.[Martins and Guyennet 2010]. However, storage covert channels are far simpler to detect and mitigate than timing covert channels.

Little focus has been given to covert channels in the updated IEEE 802.15.4 - 2020 protocol, a *de facto* standard for the underlying WSN infrastrucutre of many IoT and Industry 4.0 systems. Additionally, there is no simulation model or assessment that includes such covert channel implementations, making it difficult for security experts and network designers to assess risks and further support the development of detection and mitigation strategies.

Chapter 3

Overview of the 802.15.4 protocol

3.1 General Description

Many remote sensors need to operate on battery power without attention for years, and this low power is one of the key of the IEEE 802.15.4 standard. The reason why, is this protocol was developed to provide low cost communications and low power wireless connectivity networks in the Open Systems Interconnection (OSI) [notes n.d.].

3.1.1 IEEE 802.15.4-2003

The first protocol was written in 2003 and was designed for a low-data rate, low-power, low cost, short range radio frequency and low-complexity for Wireless Personal Area Network (WPAN) [Ramonet and Noguchi 2019]. A WPAN system is composed of several devices that can be either Full Function Device (FFD) or Reduced Function Device (RFD). The first one can communicate with both FFD or RFD and can implement the all standard while the second one only operates as a simple device and can communicate with only FFD and can implement only a small portion of the standard ["IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4" 2006].

This standard defines the Physical Layer (PHY) and the Medium Access Control (MAC) sublayer specifications in the OSI model , as seen in figure 3.1

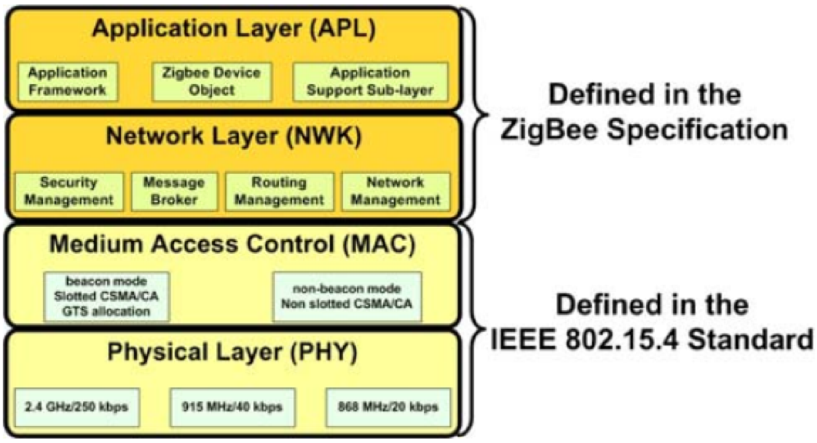


Figure 3.1: IEEE 802.15.4 layers [Cunha 2007]

Physical Layer

This standard defines three PHYs: the most used operating a 2.4 Ghz, with a maxium data rate of 250 Kp/s,and the less commonly used: 916 MHz and 868 MHz [Ramonet and Noguchi 2020]. The first transmission uses 16 channels, the second 10 channels and the third just one channel [Ramonet and Noguchi 2019].

MAC Sublayer

The MAC sublayer is the most important one since the study of this thesis is mostly focused on this layer, mainly because of the two major components which will be explained below.

- Beacon: is a periodic signal emitted by the Personal Area Network (PAN) coordinator of the network and the communication between nodes is synchronized and is the only broadcasted message [Ramonet and Noguchi 2020, Valero, Bourgeois, and Beyah 2010].

- Superframe: has a Contention Access Period (CAP) and a Contention Free Period (CFP). In the CAP, a Carrier-sense multiple access with collision avoidance (CSMA/CA) algorithm is used for the transmission of the data allowing the network to compete in order to allocate slots in the CPF [Ramonet and Noguchi 2020, Valero, Bourgeois, and Beyah 2010]. The CPF is divided into Guaranteed Time Slots (GTS) which are assigned to specific nodes for transmission without contention, in other words a node can allocate it to sen a message to another node, and when its time arrives, that superframe slot will responsible for the transmission of the packets from the node allocating to its destination. This superframe has 16 time slots, but 7 of them are reserved for CFP and the rest are allocated for the CSMA/CA integration in the CAP.

3.1.2 IEEE 802.15.4 - 2006

In 2006 the first revision of the 2003 standard was made and in this a field in the Frame Control Field (FCF) of the MAC Header was added to make it easier to check the version in use. The most significant changes in this revision were in the physical layer was on the 868 MHz and 915 Mhz bands because they operated with a data rate of 100 kp/s and in this version they can operate in a data rate to 250 kp/s lihe the 2.4 GHz band. As for the MAC layer this revision enables to specify beacons to start via parameter in the MAC layer. This way, with pre-establishing the start time will help to reduce beacon collision in the network [Ramonet and Noguchi 2020].

3.1.3 IEEE 802.15.4e (enhanced)

In the previous standards, commercial and industrial applications didn't have the necessary support so this resulted to the creation of the amendment IEEE 802.15.4-2011, named IEEE 802.15.4e (in 16 April 2012). With this standard was added five MAC behaviours to facilitate industrial applications [Ramonet and Noguchi 2019].

- Deterministic Synchronous Multi-channel Extension (DSME): This new behaviour is targeted to applications that require high levels of reliability or deterministic latency.
- Time Slotted Channel Hopping (TSCH): Intended for robustness, in other words to applications prone to collisions caused by the saturation of the network (like chemical and pharmaceutical production)TSCH applications
- Low Latency Deterministic Network (LLDN): Designed for factory automation where it is necessary low-latency
- Asynchronous Multi-Channel Adaptation (AMCA): Designed to work with low channel quality either by noise or the presence of a large number of devices. The AMCA selects the channel with the highest link quality either listening or transmitting.
- Radio Frequency Identification (RFID): This standard specifies a MAC behaviour named BLINK which is a specific kind of RFID that transmits encrypted data and is well suited for applications that has sensitive information.

The case of the DSME and TSCH behavior will be explained further on, since they are main points of study in this thesis.

Besides the new additions in the protocol, it also brought some improvements:

Multi-channel access

In the legacy IEEE 802.15.4 one of the main disadvantages was the lack of multi-channel access because this only supports a single channel for communication, restricting the capability to accommodate a large number of nodes without contention. With the IEEE 802.15.4e this limitations were overcome by supporting multi-channel operations. With channel hopping, the sequence is statically predetermined in advance. Contrariwise, with the channel adaptation the PAN coordinator has the ability to allocate different channels for data transmission [Kurunathan 2021] .

Information Elements (IE)

IEs already existed in the original IEEE 802.15.4, but, the IEEE 802.15.4e brought new addition features, such as a DSME behaviour.The IE has the information regarding the superframe specification, such as the number of superframes in a multi-superframe, number of channels time synchronization specification, Group Acknowledgement and channel hopping specifications [Kurunathan 2021].

Low latency and low energy

With this new standard was an improvement for some features like the low latency communications, more suitable for industrial control applications and providing a trade-off between latency and energy efficiency. The IEEE 802.15.4e allows devices to operate at a very low duty-cycle and also provides deterministic latency, which is a main requirement for time-critical applications. The amendment brought two low energy mechanisms based on the latency requirements. One, is the Coordinated Sampled Listening (CSL) and it is used normally for applications with very low latency. And the other, is the Receiver Initiated Transmissions (RIT) that it user for latency-tolerant applications (i.e tolerating latency of more than 10 seconds) [Kurunathan 2021].

Multi-purpose frames

All the MAC behaviours in the IEEE 802.15.4 protocol are based on specific features of each behavior and target application. The DSME variant was designed to support applications where determinism and scability are fundamental. This frame can supports guaranteed time-slot with multi-channel capability. It also provides a Group Acknowledgements (GACK) to reduce the overall delay for several GTS based transmissions [Kurunathan 2021].

Enhanced Beacons

The Enhanced Beacons are a revision of the standard beacons used in the legacy IEEE 802.15.4 protocol. This new feature provides applications-specific content to the DSME and the TSHC. They contain information on whether TSCH/DSME and low-energy mode, and information about the respective channel hopping sequences [Kurunathan 2021].

MAC performance metrics

The new protocol provides information on the link performance (quality and channel) to help the network layer to take efficient routing decisions, in order to reduce the overall power consumption and latency of the network. The information transmitted includes [Kurunathan 2021]:

- number of frames that required one or more retries before acknowledgment;
- number of frames that did not result in acknowledgment;
- number of frames that were acknowledged properly;
- number of received frames that were discarded due to being considered insecure.

Fast Association

In the IEEE 802.15.4, a device association to a network must wait till the end of the MAC response wait time before requesting the association data from the PAN Coordinator, this results in a delay on network. The new enhanced protocol, introduces a Fast Association mechanism, which results in the removal of the delay from the connection, i.e the PAN Coordinator allocates a short address to the device and also sends an association response that contains the assigned short address and the status indicator of the association [Kurunathan 2021].

Group Acknowledgement

In the IEEE 802.15.4e the DSME and the LLDN behaviors allows several successful transmission into a single GACK either within the adjacent beacon interval or as a separate Group Acknowledgment frame. The PAN Coordinator can, only for a dedicated time-slot, assigned a certain GACK that will deal just with that time-slot. This will improve the efficiency by reducing the number of single acknowledgements travelling the network, so it will saves energy and time of the network [Kurunathan 2021].

3.1.4 IEEE 802.15.4-2020

Compared to the previous standard, the standard released in 2020 (the IEEE 802.15.4-2020) brought the PHY and MAC sublayer spececifications for low-data-rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements. Therefore, the standard provides modes that allow for precision ranging and the PHY are also defined for devices operating in a variety of geographical areas [“IEEE Standard for Low-Rate Wireless Networks” 2020].

3.2 Deterministic Synchronous Multi-channel Extension (DSME)

DSME was a MAC behaviour introduced in the enhanced standard, and made official int the IEEE 802.15.4-2020 [“IEEE Standard for Low-Rate Wireless Networks” 2020]. Several improvements to DSME were proposed, but namely [Kurunathan 2021]:

- Multi-superframe;
- CAP reduction;
- Beacon Scheduling
- Channel diversity

Multi-superframe

The Figure 3.2 shows an example of a DSME superframe structure. The superframes are composed by 16 time slots: 1 slot for the beacon containing network and time information, 8 slots for the CAP and 7 slots for the CFP. During the CAP, nodes exchange control messages via CSMA/CA using a single channel. The CFP is subdivided into GTS which are spread over time and frequency and grant an exclusive access to the shared medium [Meyer, Mantilla, and Turau 2020].

With the DSME implemented it was also necessary to develop a structure that better suits it and with that came the multi-superframe. In figure3.2 it is possible to see what a multi-superframe is and the superframes present in a multi-superframe varies according to the defined Superframe Order (SO). The standard defines the structure of the superframe by the values of the Multi-superframe Duration (MD), Superframe Duration (SD) and the Beacon Interval (BI). These parameters are defined in the following equations Kurunathan 2021:

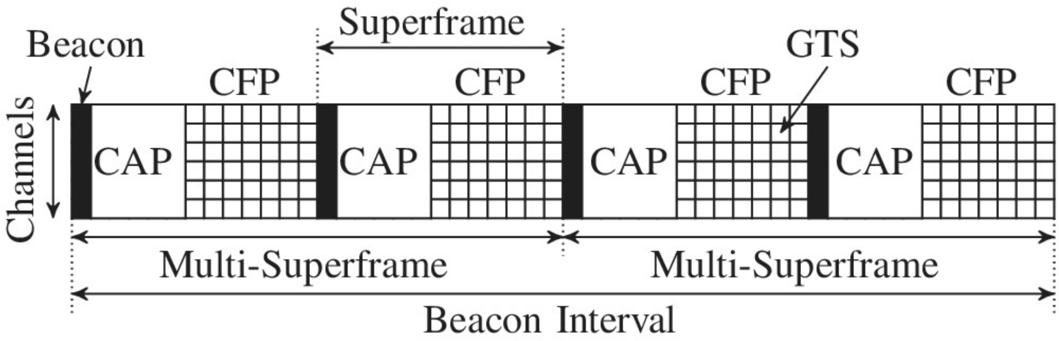


Figure 3.2: DSME Superframe example [Meyer, Mantilla, and Turau 2020]

$$MD = aBaseSuperframeduration * 2^{MOsymbols}, for 0 \leq SO \leq MO \leq BO \leq 14 \quad (3.1)$$

$$BI = aBaseSuperframeduration * 2^{BOsymbols}, for 0 \leq BO \leq 14 \quad (3.2)$$

$$SD = aBaseSuperframeduration * 2^{SOsymbols}, for 0 \leq SO \leq BO \leq 14 \quad (3.3)$$

In the previous definitions the BO is the MAC Beacon Order and this value defines the transmission interval of a beacon in a superframe. In the case of the MO, this is the MAC Multi-superframe order and this value represents the beacon interval of the superframe. And the SO is the superframe Order andis used to compute the amount of superframes are present in a given network's multi-superframe. The "aBaseSuperframeDuration" is the minimum duration of a superframe corresponding to the initial order of the superframe, having a duration fixed to 960 symbols (each symbol represents 4 bits) corresponding to 15.36 ms.The total number of superframes and multi-superframes in a DSME network can be determined by 2^{BO-SO} and 2^{BO-MO} , respectively [Kurunathan 2021].

CAP Reduction

CAP reduction was introduced to the DSME in order to possibility of the protocol remove the CAP on all but the first superframe of a multi-superframe needs to have it. (this permits the other superframes to have a longer CFP as seen in figure 3.3). This will permit that only the first superframe of the multi-superframe will have de the beacon slot occupied, so the other 15 slots will be dedicated to GTS. that only With this feature the performances of the network is increased and the standard provides a CAP reduction flag. If the flag is enable than the CAP Reduction is being used ["IEEE Standard for Low-Rate Wireless Networks" 2020, Kurunathan 2021].

Figure 3.3: CAP reduction [Cunha 2007]

Beacon Scheduling

A DSME network has all the devices time-synchronized using the values of the Timestamp field of the received beacons from the device they are associated with by keeping global synchronization int the PAN. If it is a node need to join the network, the other nodes try to communicate sending their beacon schedule information and this shedule is update as a bitmap sequence. The new node try to find a vacant beacon slot and claim it to use to send its own beacons. The outcome of this situations is overlapping of the transmission beacon slot because it results in collisions. This is solver with the DSME-Beacon allocation notification because it is a command frame that it is sent to all nodes in the network by broadcast. This permits to all nodes collect the allocation information and add it to their Allocation Counter Table (ACT) and conflict is avoided Kurunathan 2021, “IEEE Standard for Low-Rate Wireless Networks” 2020].

Channel Diversity

This feature permits two channel diversity mechanisms. The first one is channel adaptation that consist in the PAN Coordinator has the capability to allocate the DSME guaranteed timeslots in a single channel or through different channel to an end device, Although if the link quality of an allocated DSME GTS becomes degrades, the PAN Coordinator can also deallocate a specific DSME GTS. The second one is the channel hopping that is a methodology by which several devices hop over different channel with a predefined order. This technique is used in radio communication systems when many receivers can select a channel from a predefined set to receive the required broadcast information. This mechanism is used to provide orthogonality among all devices employing the same channel hopping sequence list Kurunathan 2021, “IEEE Standard for Low-Rate Wireless Networks” 2020].

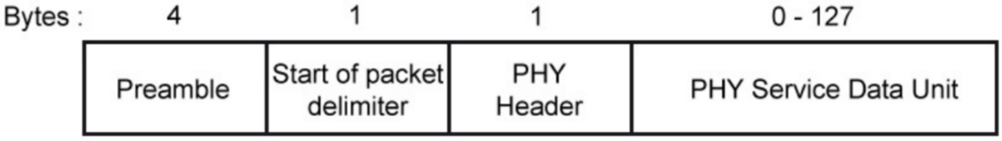
Chapter 4

Steganography Opportunities

By comparing [Martins and Guyennet 2010] and the current standard (IEEE 802.15.4-2020) it was possible to understand that there are several new hypotheses for hiding data. In this Chapter, we show the significant differences between the protocols in hiding data as well as the amount of possible information that can be accommodated.

4.1 PHY Layer

The physical layer is the first and lowest layer of the OSI Model and deals with bit-level transmission between different devices and supports electrical or mechanical interfaces connecting to the physical medium for synchronized communication. Comparing the information between the article and the IEEE 802.15.4-2020 standard we can conclude that there aren't differences when it comes to hiding information. The general format of the PHY frame can be seen in Figure 4.1. The PHY header field provides the length of the PHY Service Data Unit Field and with this information is possible to know that the field is encode on one bite, but only seven are used (so the eight is reserved). If this bit is never used because it is reserved, it is possible to hide some data.

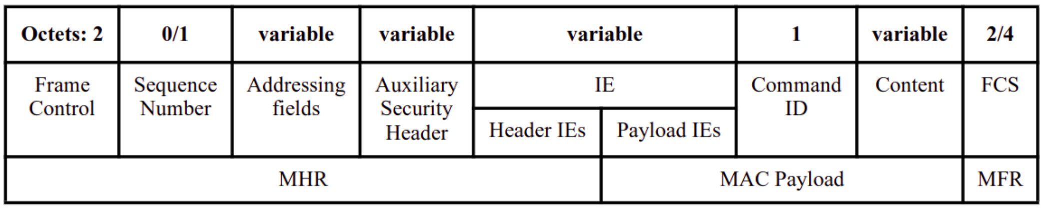


Bytes :	4	1	1	0 - 127
	Preamble	Start of packet delimiter	PHY Header	PHY Service Data Unit

Figure 4.1: General PHY Frame Format ["IEEE Standard for Low-Rate Wireless Networks" 2020]

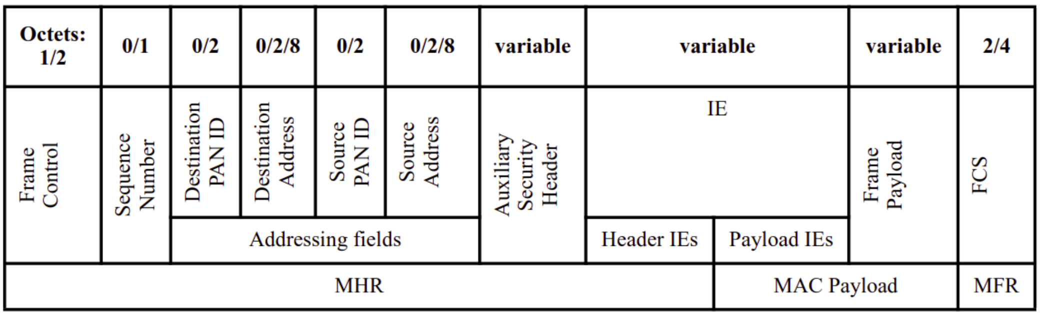
4.2 MAC Layer

The MAC Layer is one of the two sublayers into which the data-link layer of the OSI model is subdivided in accordance to the specifications of the IEEE 802.15.4. Regarding the MAC layer there were some differences that will be presented in this document. First of all it is necessary to understand the general MAC format that is possible to see in the figure 4.2. Currently, the format of the MAC layer format is shown in figure 4.3.



Octets: 2	**0/1**	**variable**	**variable**	**variable**		**1**	**variable**	**2/4**
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	IE		Command ID	Content	FCS
				Header IEs	Payload IEs			
MHR					MAC Payload			MFR

Figure 4.2: General MAC Format ["IEEE Standard for Low-Rate Wireless Networks" 2020]



Octets: 1/2	**0/1**	**0/2**	**0/2/8**	**0/2**	**0/2/8**	**variable**	**variable**		**variable**	**2/4**
Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Auxiliary Security Header	IE		Frame Payload	FCS
		Addressing fields					Header IEs	Payload IEs		
MHR								MAC Payload		MFR

Figure 4.3: Actual General MAC Format ["IEEE Standard for Low-Rate Wireless Networks" 2020]

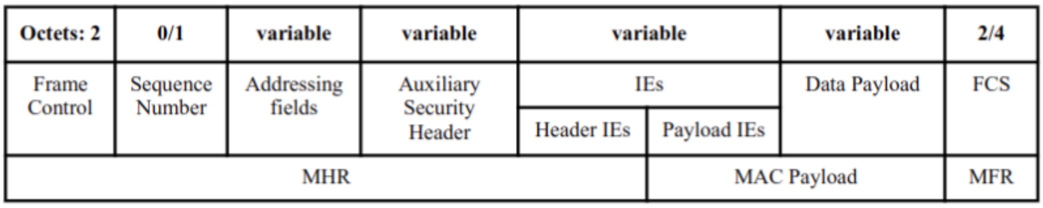
In short, the format of the MAC layer has not changed, so it is necessary to study the different kind of frames. The Mac layer uses different kind of frames and the main ones will be studied in the next sections. Some of the frames that will be studied are present in the next table 4.1.

Table 4.1: Frame type field

Frame Type value	Description
000	Beacon
001	Data
010	Acknowledgment
011	MAC command
100	Reserved
101	Multipurpose
110	Fragment or Frak
111	Extended

Data Frame

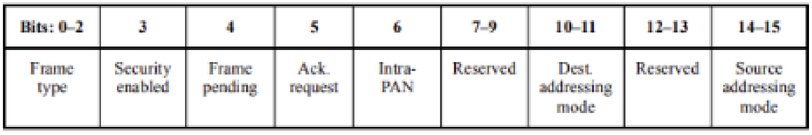
The Data frame shall be formatted as illustrated in Figure 4.4. There are opportunities in different sections of it.



Octets: 2	0/1	variable	variable	variable		variable	2/4
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	IEs		Data Payload	FCS
				Header IEs	Payload IEs		
MHR					MAC Payload		MFR

Figure 4.4: Data Frame Format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

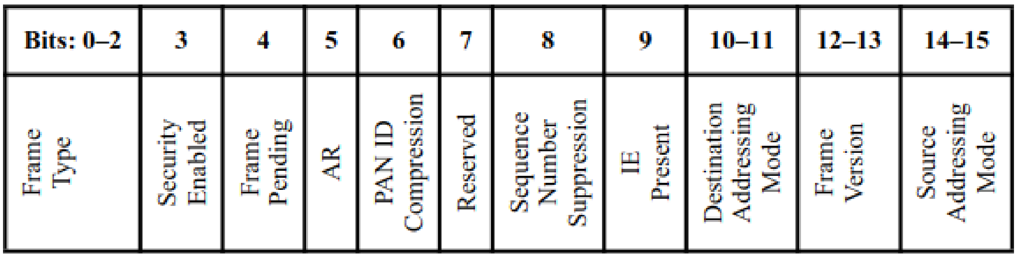
- **Frame Control Field**



Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Figure 4.5: General frame control field before [“IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4” 2006]

In the first figure we could see that the the 7-9th bits and the 12-13th bits were reserved and could be used to hide a stego object. So, it was possible to



Bits: 0–2	3	4	5	6	7	8	9	10–11	12–13	14–15
Frame Type	Security Enabled	Frame Pending	AR	PAN ID Compression	Reserved	Sequence Number Suppression	IE Present	Destination Addressing Mode	Frame Version	Source Addressing Mode

Figure 4.6: General frame control field now [“IEEE Standard for Low-Rate Wireless Networks” 2020]

encode three and two bits, respectively, in these fields like it is seen in Figure 4.5.

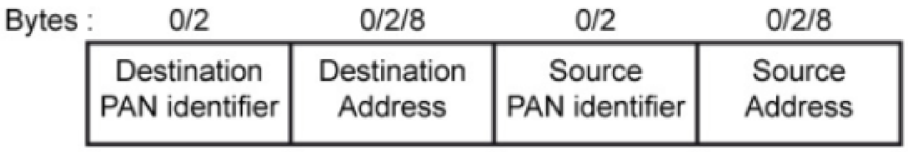
Now, only the 7th bit is reserved, so maybe it’s possible just to encode one bit there like it is seen in Figure 4.6. The IE Present field shall be set to one if IEs are contained in the frame. If not the Frame Version is 0b00 or 0b01, the IE field shall be zero.

- **Sequence Number Field**

In the Sequence Number field is possible to know that each packet contains 8 bits, used in particular with packet acknowledgements to specify which packet has been acknowledged. The value of this number corresponds to the PIB mac DSN variable. The value of this variable is initialized randomly, then incremented after each received packet. If we could choose the initialized number of the PIB variable, it would be possible to hide a stego object (or a part of the stego object) inside (one byte of data could be hide).

- **Adress Info Field**

The size of Address Info field varies between four and 20 bytes, so it is possible choose to have a short (16 bits) or an extended source address (64 bits) Figure 4.7. It is possible to hide data in this field, for example, if we specify a nonexistent source address. With this nonexistent address, we can hide a stego object with a size up to 64 bits. This steganographic technique can be particularly undetectable if the network does not know the exact number of nodes present in the network, especially in a big network where nodes can be added over time.

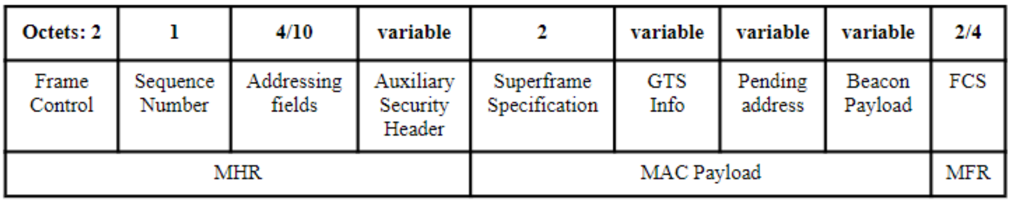


Bytes :	0/2	0/2/8	0/2	0/2/8
	Destination PAN identifier	Destination Address	Source PAN identifier	Source Address

Figure 4.7: Address Info field structure [“IEEE Standard for Low-Rate Wireless Networks” 2020]

Beacon Frame

The frame format can be seen in the Figure 4.8. It can be conclude that exist the same possibilities for hiding information as in the Data Frame, in the Frame Control and in the Source Address Information field. The Beacon Sequence Number field give us another way to use the cover object. The Beacon Sequence Number field contains the sequence number of the Beacon node. This number is given by the macBSN variable. This variable is ordinarily initialized randomly. As in the Sequence Number field of the MAC data frame, we can voluntarily choose the value of this number and then hide up to one byte of data.

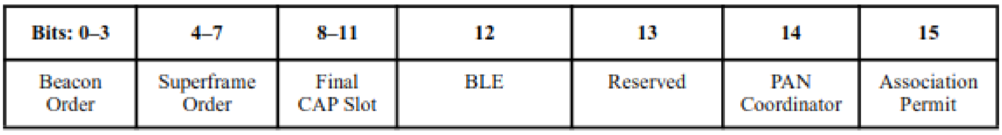


Octets: 2	1	4/10	variable	2	variable	variable	variable	2/4
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	Superframe Specification	GTS Info	Pending address	Beacon Payload	FCS
MHR				MAC Payload				MFR

Figure 4.8: Beacon Frame Format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

- **Superframe Specification field**

  The Superframe Specification field shall be formatted as illustrated in Figure 4.9



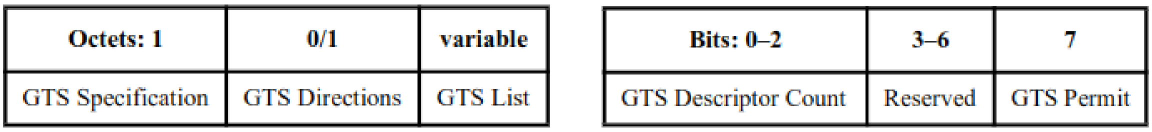
Bits: 0–3	4–7	8–11	12	13	14	15
Beacon Order	Superframe Order	Final CAP Slot	BLE	Reserved	PAN Coordinator	Association Permit

Figure 4.9: Format of the Superframe Specification field [“IEEE Standard for Low-Rate Wireless Networks” 2020]

In this superframe the 13th bit is reserved so it is possible to hide data.

- **GTS Info field**

  The GTS Info field shall be formatted as illustrated in Figure 4.10 - (a) and the GTS Specification field shall be formatted as illustrated in Figure 4.10 . (b)



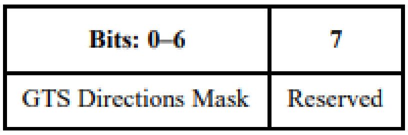
Octets: 1	0/1	variable
GTS Specification	GTS Directions	GTS List

Bits: 0–2	3–6	7
GTS Descriptor Count	Reserved	GTS Permit

Figure 4.10: (a) GTS information field (b) GTS Specification field ["IEEE Standard for Low-Rate Wireless Networks" 2020]

So like we can see in the figure 4.10 - (b) the bits between 3 to 6 are reserved so it is possible to hide 4 bits of data here.

If the GTS Direction is set to one so the format will be formatted as illustrated in Figure 4.11 the seven bit will be reserved.

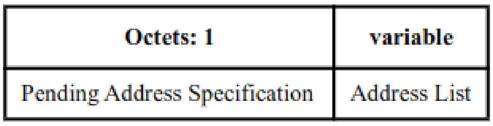


Bits: 0–6	7
GTS Directions Mask	Reserved

Figure 4.11: Format of the GTS Directions field ["IEEE Standard for Low-Rate Wireless Networks" 2020]

- **Pending Address field**

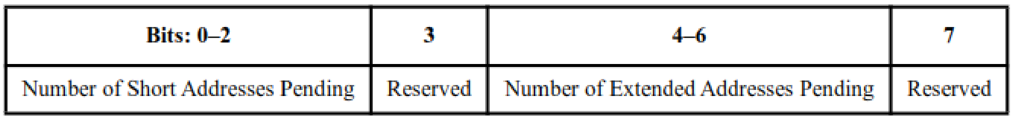
  The Pending Address field shall be formatted as illustrated in Figure 4.12



Octets: 1	variable
Pending Address Specification	Address List

Figure 4.12: Pending Address field format ["IEEE Standard for Low-Rate Wireless Networks" 2020]

The Pending Address Specification field shall be formatted as illustrated in Figure 4.13. So, if the 3rd and the 7th bit are reserved it will be possible to hide 2 bits of data there.

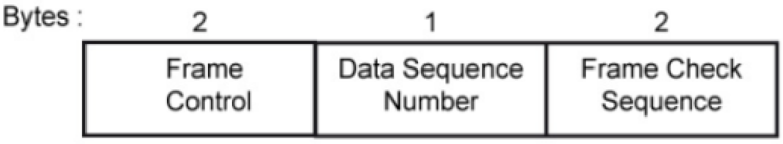


Bits: 0–2	3	4–6	7
Number of Short Addresses Pending	Reserved	Number of Extended Addresses Pending	Reserved

Figure 4.13: Format of the Pending Address Specification field ["IEEE Standard for Low-Rate Wireless Networks" 2020]

Acknowledgment frame

The structure of this frame can be see in Figure 4.14. The possibilities to hide data are the same then the previous standard. So the concussions remain the same, in other words we have the same possibilities to hide data in the Frame Control field and Data Sequence field because both are identical to fields in MAC data frame.

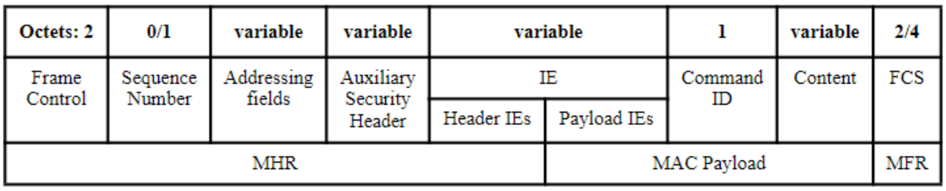


Bytes :	2	1	2
	Frame Control	Data Sequence Number	Frame Check Sequence

Figure 4.14: Acknowledgement frame structure ["IEEE Standard for Low-Rate Wireless Networks" 2020]

MAC command frame

The MAC command frame format can be see in figure 4.15. So it has the possibility to hide information in the fields Frame Control, Data Sequence Number and Address Information such as the MAC data frame.

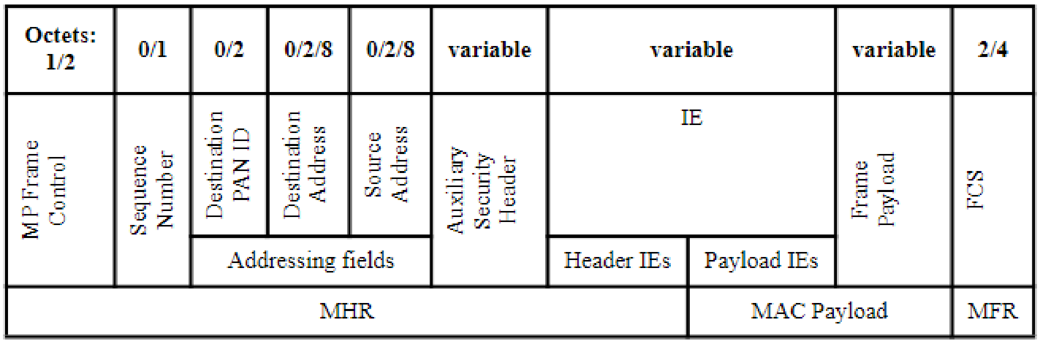


Octets: 2	0/1	variable	variable	variable		1	variable	2/4
Frame Control	Sequence Number	Addressing fields	Auxiliary Security Header	IE		Command ID	Content	FCS
				Header IEs	Payload IEs			
MHR					MAC Payload			MFR

Figure 4.15: MAC command frame format ["IEEE Standard for Low-Rate Wireless Networks" 2020]

Multipurpose frame format

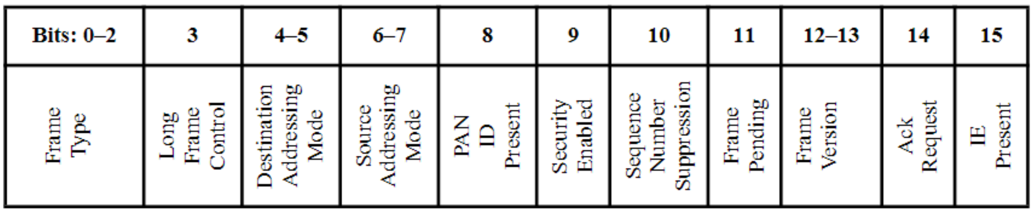
The Multipurpose frame type was introduced in the new standard and provides a flexible format that may used for a variety of purposes. The format supports a short and long form of the Frame Control field, and allows for all the fields in the MHR to be present or omitted as specified by the generating service. Multipurpose frames are treated in the same manner as Data frames, and their content is passed to/from the next higher layer using the MAC common part sublayer (MCPS) DATA primitives. The multipurpose frame shall be formatted as illustrated in Figure 4.16



Octets: 1/2	0/1	0/2	0/2/8	0/2/8	variable	variable		variable	2/4
MP Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source Address	Auxiliary Security Header	IE		Frame Payload	FCS
		Addressing fields				Header IEs	Payload IEs		
MHR							MAC Payload		MFR

Figure 4.16: Multipurpose frame format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

The MP Frame Control Field can be seen in the figure 4.17



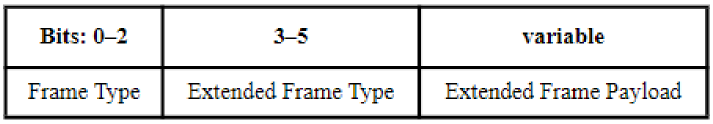
Bits: 0–2	3	4–5	6–7	8	9	10	11	12–13	14	15
Frame Type	Long Frame Control	Destination Addressing Mode	Source Addressing Mode	PAN ID Present	Security Enabled	Sequence Number Suppression	Frame Pending	Frame Version	Ack Request	IE Present

Figure 4.17: Format of the MP Frame Control field [“IEEE Standard for Low-Rate Wireless Networks” 2020]

It will be possible to hide data in the IE field.

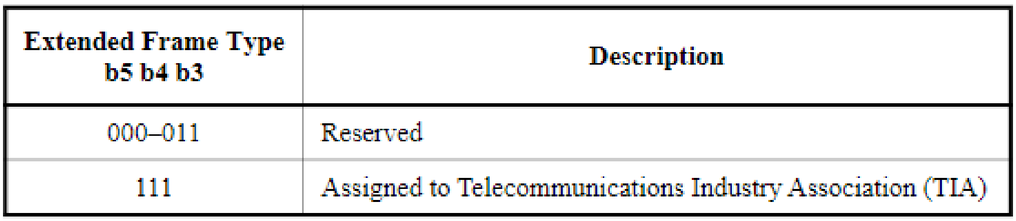
Extended frame format

The Extended frame format can be see in figure 4.18. In the Extended Frame Type has values that is represented in the figure 4.19. Currently, the standard does not have any frame formats defined for the reserved values Extended Frame Type fields, so there are 4 bits not used that could hide data.



Bits: 0–2	3–5	variable
Frame Type	Extended Frame Type	Extended Frame Payload

Figure 4.18: Extended frame format ["IEEE Standard for Low-Rate Wireless Networks" 2020]



Extended Frame Type b5 b4 b3	Description
000–011	Reserved
111	Assigned to Telecommunications Industry Association (TIA)

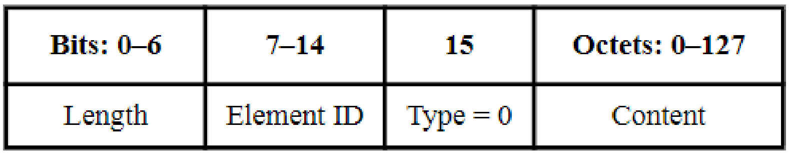
Figure 4.19: Extended Frame Type values ["IEEE Standard for Low-Rate Wireless Networks" 2020]

IE

The next subsections will be about possibilities to hide data in some IE terminations. So, all frames that had the possibility to have the IE ending will have the possibility to hide data there.

Header IE format

The Header IE shall be formatted as illustrated in Figure4.20.The Length field specifies the number of octets in the Content field.The Element IDs are defined in for each of the Header IEs. The element ID form 0x01 to 0x19 and from 0x80 to 0xff are reserved so it is possible to hide some data here (7 bits in the element ID that is reserved)(see attachment A).



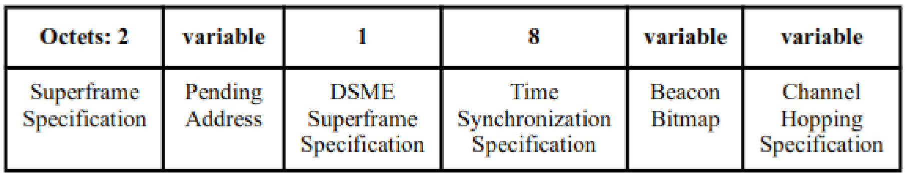
Bits: 0–6	7–14	15	Octets: 0–127
Length	Element ID	Type = 0	Content

Figure 4.20: Format of Header IEs ["IEEE Standard for Low-Rate Wireless Networks" 2020]

DSME PAN descriptor IE

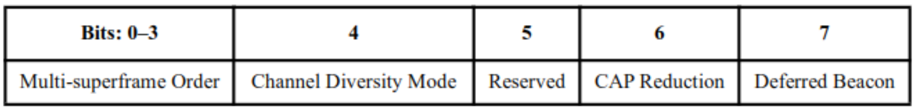
The DSME PAN Descriptor IE shall be included in enhanced beacons that are sent every beacon interval ina DSME-enabled PAN.

The DSME PAN Descriptor IE Content field shall be formatted as illustrated in Figure 5.4.The Superframe Specification field is described in figure A.1. The 5th bit is reserved so ti may have the possibilitie to hide data.



Octets: 2	variable	1	8	variable	variable
Superframe Specification	Pending Address	DSME Superframe Specification	Time Synchronization Specification	Beacon Bitmap	Channel Hopping Specification

Figure 4.21: DSME PAN Descriptor IE Content field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

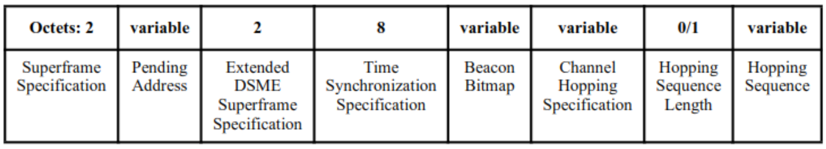


Bits: 0–3	4	5	6	7
Multi-superframe Order	Channel Diversity Mode	Reserved	CAP Reduction	Deferred Beacon

Figure 4.22: DSME Superframe Specification field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

Extended DSME PAN descriptor IE

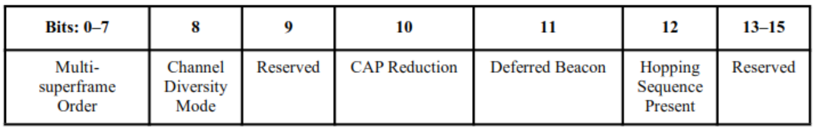
The format of the Extended DSME PAN Descriptor IE Content field shall be as illustrated in Figure 4.23.



Octets: 2	variable	2	8	variable	variable	0/1	variable
Superframe Specification	Pending Address	Extended DSME Superframe Specification	Time Synchronization Specification	Beacon Bitmap	Channel Hopping Specification	Hopping Sequence Length	Hopping Sequence

Figure 4.23: Extended DSME PAN Descriptor IE Content field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

The superframe can be seen in Figure 4.24 and it is possible to see that the 9th, the 13th, 14th and 15th bits are reserved so it is possible to hide 4 bits of data in this superframe.

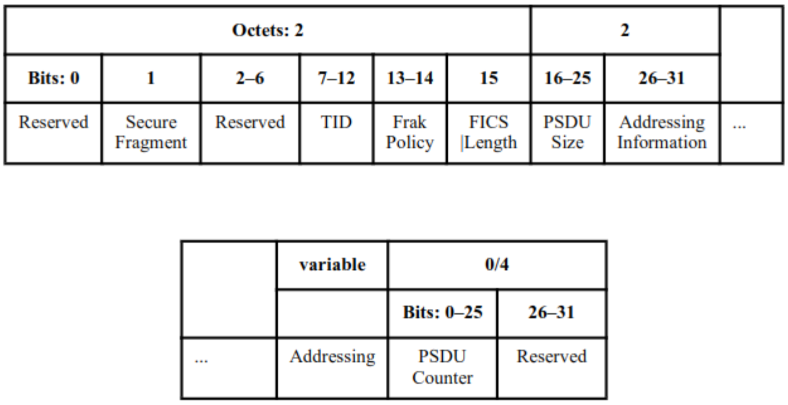


Bits: 0–7	8	9	10	11	12	13–15
Multi-superframe Order	Channel Diversity Mode	Reserved	CAP Reduction	Deferred Beacon	Hopping Sequence Present	Reserved

Figure 4.24: Format of the Extended DSME Superframe Specification field [“IEEE Standard for Low-Rate Wireless Networks” 2020]

Fragment Sequence Context Description (FSCD) IE

In Figure 4.25 it is possible to see the IE Content field format. In conclusion, if we analyse the figure we can see that 12 bits int total are reserved so it maybe will possible to hide data there.



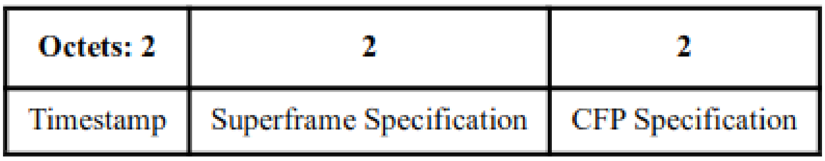
Octets: 2						2		
Bits: 0	1	2–6	7–12	13–14	15	16–25	26–31	
Reserved	Secure Fragment	Reserved	TID	Frak Policy	FICS |Length	PSDU Size	Addressing Information	...

	variable	0/4	
		Bits: 0–25	26–31
...	Addressing	PSDU Counter	Reserved

Figure 4.25: FSCD IE Content field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

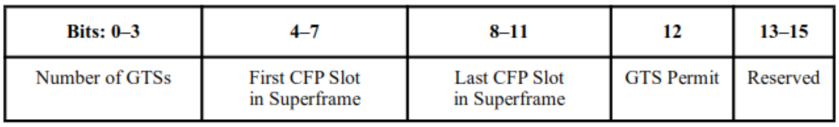
Simplified Superframe Specification IE

The Simplified Superframe Specification IE Content field shall be formatted as illustrated in Figure 4.26 and the CFP Specification field shall be encoded as illustrated in Figure 4.27. The bits 13th, 14th and 15th are reserved so it is possible to hide data here.



Octets: 2	2	2
Timestamp	Superframe Specification	CFP Specification

Figure 4.26: Simplified Superframe Specification IE Content field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]



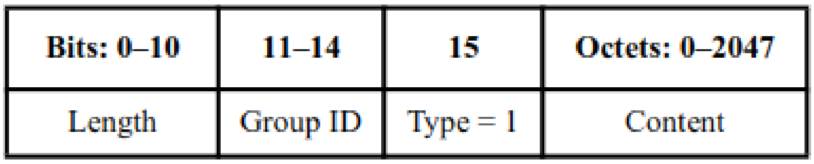
Bits: 0–3	4–7	8–11	12	13–15
Number of GTSs	First CFP Slot in Superframe	Last CFP Slot in Superframe	GTS Permit	Reserved

Figure 4.27: CFP Specification field format [“IEEE Standard for Low-Rate Wireless Networks” 2020]

Payload IEs

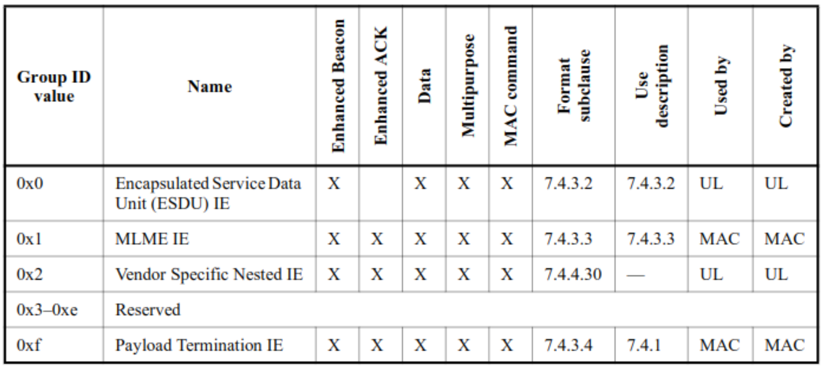
- **General**

  The general format of the Payload IE is shown in Figure 4.28



Bits: 0–10	11–14	15	Octets: 0–2047
Length	Group ID	Type = 1	Content

Figure 4.28: Format of Payload IEs [“IEEE Standard for Low-Rate Wireless Networks” 2020]



Group ID value	Name	Enhanced Beacon	Enhanced ACK	Data	Multipurpose	MAC command	Format subclause	Use description	Used by	Created by
0x0	Encapsulated Service Data Unit (ESDU) IE	X		X	X	X	7.4.3.2	7.4.3.2	UL	UL
0x1	MLME IE	X	X	X	X	X	7.4.3.3	7.4.3.3	MAC	MAC
0x2	Vendor Specific Nested IE	X	X	X	X	X	7.4.4.30	—	UL	UL
0x3–0xe	Reserved									
0xf	Payload Termination IE	X	X	X	X	X	7.4.3.4	7.4.1	MAC	MAC

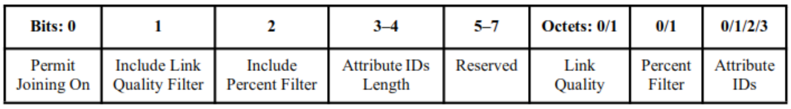
Figure 4.29: Payload IE Group ID [“IEEE Standard for Low-Rate Wireless Networks” 2020]

The Group ID from 0x3 - 0xe is reserved so in this values it is possible to hide some data. In this 4 bits exists 11 possibilities to hide data. So, maybe, in the IE field it will be possible to hide this data in payload IE.

Nested IE

- **Enhanced Beacon Filter IE**

The Enhanced Beacon Filter IE Content field shall be formatted as illustrated in Figure 4.30. No more than one Enhanced Beacon Filter IE shall be conveyed per Enhanced Beacon Request frame. The 5th, 6th and 7th are reserved so it will be possible to hide 3 bits of data.



Bits: 0	1	2	3–4	5–7	Octets: 0/1	0/1	0/1/2/3
Permit Joining On	Include Link Quality Filter	Include Percent Filter	Attribute IDs Length	Reserved	Link Quality	Percent Filter	Attribute IDs

Figure 4.30: Enhanced Beacon Filter IE Content field format ["IEEE Standard for Low-Rate Wireless Networks" 2020]

4.3 Summary of storage channel opportunities

To summarize, taking into account that at least 3 bits of the element ID of Header IE the possibilities in study in this chapter according the table 4.1 are:

- Example of a sent MAC data frame: This frame has the possibilities to use IE, so to the values obtain in table 4.2 it is necessary to sum the values of this. Using the 7 bits reserved from the element ID the total number of bits to be hidden will be: 73 + 7 = 80 bits

- Example of a sent MAC beacon frame: This frame has the possibility to enable the enhanced beacon filter IE, so this give 3 more bits to hide - 65 + 3 = 68 bits

- Example of a sent MAC acknowledgment frame: The acknowledgement frame only has the possibility to hide 9 bits.

- Example of a sent MAC command frame: The command frame has the same amount of bits as the data frame so the number of bits to be hidden is the same. 73 + 7 bits of IE = 80 bits.

- Example of a sent MAC Multipurpose frame: This will give the opportunity to hide 4 + 3 from IE = 8 bits

- Example of a sent MAC Fragment or Frak: This frame type doesn't include any type of reserved bits so it shouldn't be a hypothesis in study,

- Example of a sent MAC Extendend Frame: To finalize this frame it will only give 4 bits to fill out with data to hide.

In other words, theoretically it is expected to be possible to have the following number of bits to use steganographically, as shown in the table 4.2.

Table 4.2: Number of bits comparation

	802.15.4-2020	802.15.4-2010
PHY Layer	1	1
Data Frame (total)	73	77
a) Frame control field	1	5
b) Sequence Number field	8	8
c) Address Info field	64	64
Beacon frame (total)	72	69
a) Frame control field	1	5
b) Address Info field	64	64
c) Superframe Specification field	1	ND
d) GTS	4	ND
e) Pending Adress field	2	ND
Acknowledgment frame (total)	9	13
a) Frame control field	1	5
b) Sequence Number field	8	8
Command frame (total)	73	77
a) Frame control field	1	5
b) Sequence Number field	8	8
c) Address Info field	64	64
Multipurpose frame format	4	ND
Extended frame format	4	ND
IE	28	ND
a) Element ID	16	ND
hline b) Extended DSME PAN descriptor IE	1	ND
c) Extended DSME Superframe Specification	4	ND
d) Payload	4	ND
e) Enhanced Beacon Filter IE	3	ND

Chapter 5

Simulation tools

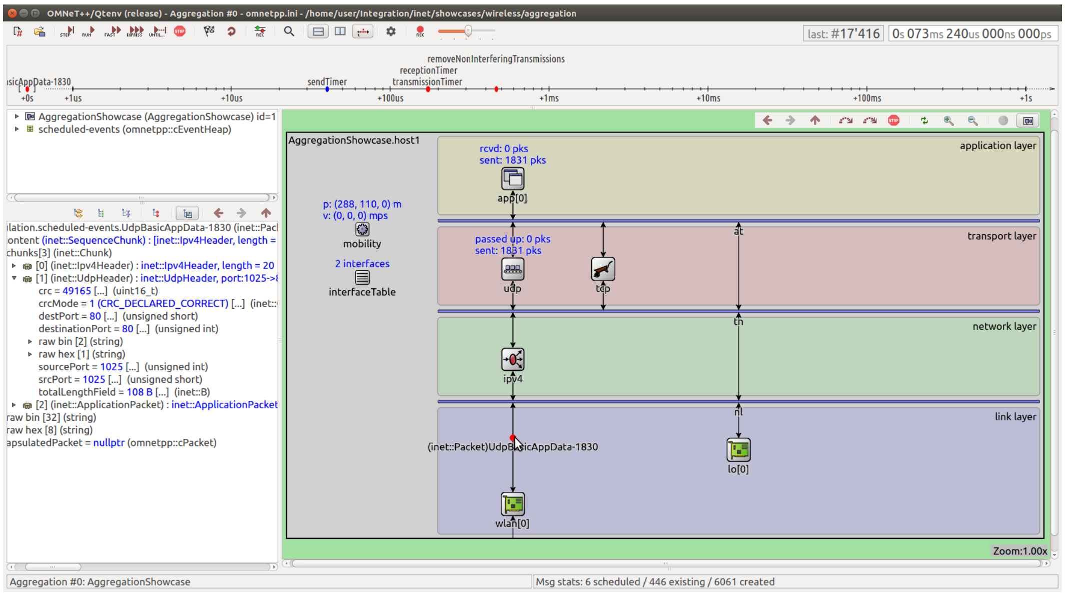
The simulation tools used in this thesis are described in general in this chapter. In addition to the underlying simulation model for the performance assessment, it also contains a thorough comparison and selection of simulation frameworks.

5.1 Network Simulators

It is essential to rely on already existing and updated IEEE 802.15.4 protocol simulation models when implementing a set of network covert channels to avoid the time-consuming task of implementing many of the complex protocol communication features. As a result, one of the key considerations in choosing a simulation framework for this assessment is whether or not there are any existing, established, and accessible models that are typically linked to that specific framework. The protocol was developed using a number of technologies and has been in use since 2003 (with many revisions, the most recent of which dates back to 2020). What follows is a summary of relevant simulation frameworks and the models that are presently available.

5.1.1 OMNeT ++

Objective Modular Network Testbed in C++ (OMNeT ++) is a is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators [*Screenshots* n.d.]. This open-source simulator can be model projects that involved communications (wireless and wired) [Varga 2003]. The



AggregationShowcase (AggregationShowcase) id=1
scheduled-events (omnetpp::cEventHeap)

```
lation.scheduled-events.UdpBasicAppData-1830 (inet::Pacl
ontent (inet::SequenceChunk) : [inet::Ipv4Header, length =
chunks[3] (inet::Chunk)
  [0] (inet::Ipv4Header) : inet::Ipv4Header, length = 20
  [1] (inet::UdpHeader) : inet::UdpHeader, port:1025->
      crc = 49165 [...]  (uint16_t)
      crcMode = 1 (CRC_DECLARED_CORRECT) [...]  (inet::
      destPort = 80 [...]  (unsigned short)
      destinationPort = 80 [...]  (unsigned int)
    raw bin [2] (string)
    raw hex [1] (string)
      sourcePort = 1025 [...]  (unsigned int)
      srcPort = 1025 [...]  (unsigned short)
      totalLengthField = 108 B [...]  (inet::B)
  [2] (inet::ApplicationPacket) : inet::ApplicationPacket
raw bin [32] (string)
raw hex [8] (string)
apsulatedPacket = nullptr (omnetpp::cPacket)
```

Aggregation #0: AggregationShowcase
Msg stats: 6 scheduled / 446 existing / 6061 created

Figure 5.1: OMNeT++ INET Simulation [*Screenshots* n.d.]

OMNeT++ alone couldn't process communication between nodes, so with the aid of the INET Framework, that is an open-source model library, it provides protocols, agents and other models for researchers and students working with communication networks. This framework INET contains models for the Internet stack (like Transmission Control Protocol (TCP), User Datagram Protocol (UDP), etc.) wired and wireless link layer protocols (Ethernet, IEEE 802.11, etc) and many other protocols and components [*What Is INET Framework?r* n.d.] The figure 5.1 is an example from a simulation with the OMNeT++.

The use of a simulation framework such as the OMNeT++ was only half the solution when it came to developing a covert channel. Since there had to be a model that could reproduce the various features of the DSME protocol, it was necessary to do a research and find an open-source implementation. This was done through the openDSME project, which was developed by a team of researchers from the *Institute of Telematics* at *Hamburg University of Technology* [Köstler et al. 2016].

Since DSME is only one of the variants of the 802.15.4 protocol, TSCH also needed to be present in the framework that was chosen in the form of a simulation model. An open-source TSCH model that integrates OMNeT++ and the INET framework to create interactions and send packets between simulation nodes was created and made available by a team of researchers from the *Hamburg University of Technology*, this time from the *Institute of Communication Networks* This concept was developed to have an effect on the advancement of wireless avionics intra-communications.

5.1.2 NS-3

ns-3 is a discrete-event simulator for Internet systems and target primarily for educational use and research. Also, this is a free, open-source-software, licensed under the *GNU GPLv2* license, and maintained by a worldwide community.It was created as an improvement on the existing widely utilized ns-2. It is successful in increasing the models' realism to bring them closer to the real-world software implementations they reflect , like the example in figure 5.2.

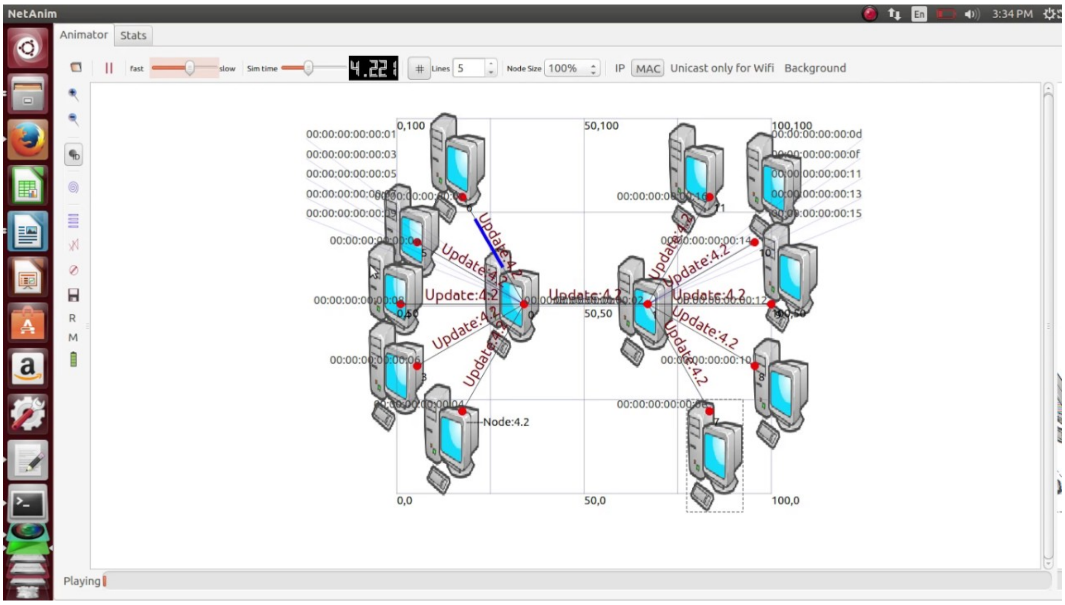


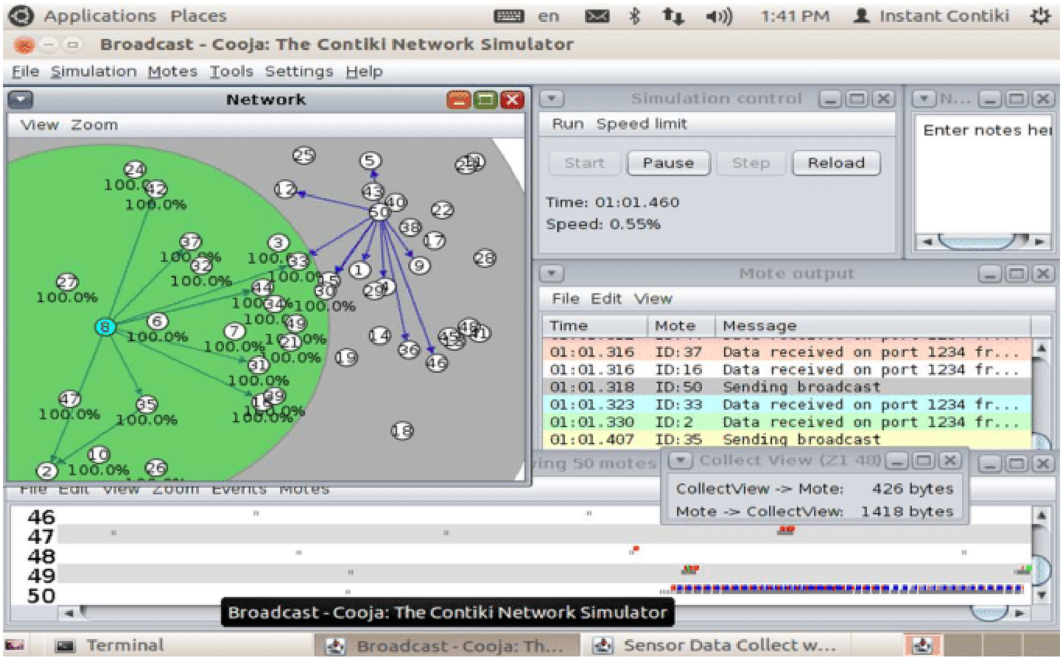
Figure 5.2: ns-3 Simulation [otosection n.d.]

To schedule the beacons and so avoid collisions and information loss in a beacon collision environment, the authors of "Analysis and Enhancement of IEEE 802.15.4e DSME Beacon Scheduling Model" successfully developed the DSME variant of the enhanced protocol IEEE 802.15.4e. This model was also tested and, based on those experiments, improvements to the beacon scheduling model present in DSME are proposed and result in an enhanced DSME [Hwang and Nam 2014].

[kourzanov 2022] developed the upgraded protocol 802.15.4e's TSCH operating mode in the ns-3 simulation framework. After that, the model was completely made available in a public GitHub repository for use by the scientific community in extending the works created.

5.1.3 CooJa

A flexible Java-based network simulator created specifically for Wireless Sensor Networks is called CooJa Simulator. This simulation framework is exclusive to the Contiki operating system and is particularly targeted at low power IoT devices.Because many components of the simulator may be readily swapped or expanded with new features, CooJa is adaptable. The simulated radio medium, the simulated node hardware, and the plug-ins for the simulated input/output are some examples of pieces that may be expanded The figure 5.3 is an example of a simulation from this software [Osterlind et al. 2006].



Time: 01:01.460
Speed: 0.55%

```
Time        Mote    Message
01:01.316   ID:37   Data received on port 1234 fr...
01:01.316   ID:16   Data received on port 1234 fr...
01:01.318   ID:50   Sending broadcast
01:01.323   ID:33   Data received on port 1234 fr...
01:01.330   ID:2    Data received on port 1234 fr...
01:01.407   ID:35   Sending broadcast
```

CollectView -> Mote: 426 bytes
Mote -> CollectView: 1418 bytes

Figure 5.3: CooJa Simulation [Omoniwa et al. 2019]

The same team behind the implementation of the OMNeT++ simulation's DSME by researchers at *Hamburg University of Technology* also created the reference model (openDSME) for the Cooja simulation framework [Köstler et al. 2016].

According to the article "Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks"[Shih, Xhafa, and Zhou 2015], an implementation of the TSCH version of 802.15.4 was achieved successfully and fully tested.

5.1.4 Simulator Selection

After a lengthy analysis of the simulation tools, it was decided to use OMNeT++ not only because it was created in the most familiar programming language, but also because it supports the OpenDSME simulation model, an open source model of the IEEE 802.15.4 DSME protocol and possesses a large online support community.

5.2 OMNeT++ Simulation Model

As previously stated, simulations will be carried out as a complementary study in this thesis on the aforementioned framework together with INET on OMNeT++. This model was provided by a team of researchers from the *Institute of Telematics* and it is composed by two fundamental layers integrated into the MAC link layer, the DSMELayer and the DSMEAdaptationLayer, as shown in figure 5.4).

- DSMELayer: is responsible for implementing the newly released DSME MAC behaviour and all its features;
- DSMEAdaptationLayer: is responsible for the IEEE 208.15.4 functions that are necessary to perform a cohesive ink with the rest of the OSI layers;

The higher layer communicates directly with the DSME layer and sends all instructions to the DSME layer via two interfaces called Service Access Points (SAP). While the MAC Common Part Sublayer (MCPS-SAP) is responsible for sending and receiving messages composed with payload (the availability of queuing messages for transmission is included). The MAC Sublayer Management Entity (MLME-SAP) is in charge of the network tasks. The DSMEPlataform interconnect the DSMELayer and the DSMEAdaptationLayer and it is responsible for the emission of times and for the flow of packets throughout the network [Köstler et al. 2016].

Unfortunately, the model, although the most complete available does not include the implementation of the IEEE 802.15.4 Information Elements. Such implementation was carried out in this Thesis and is presented in the following Chapter.

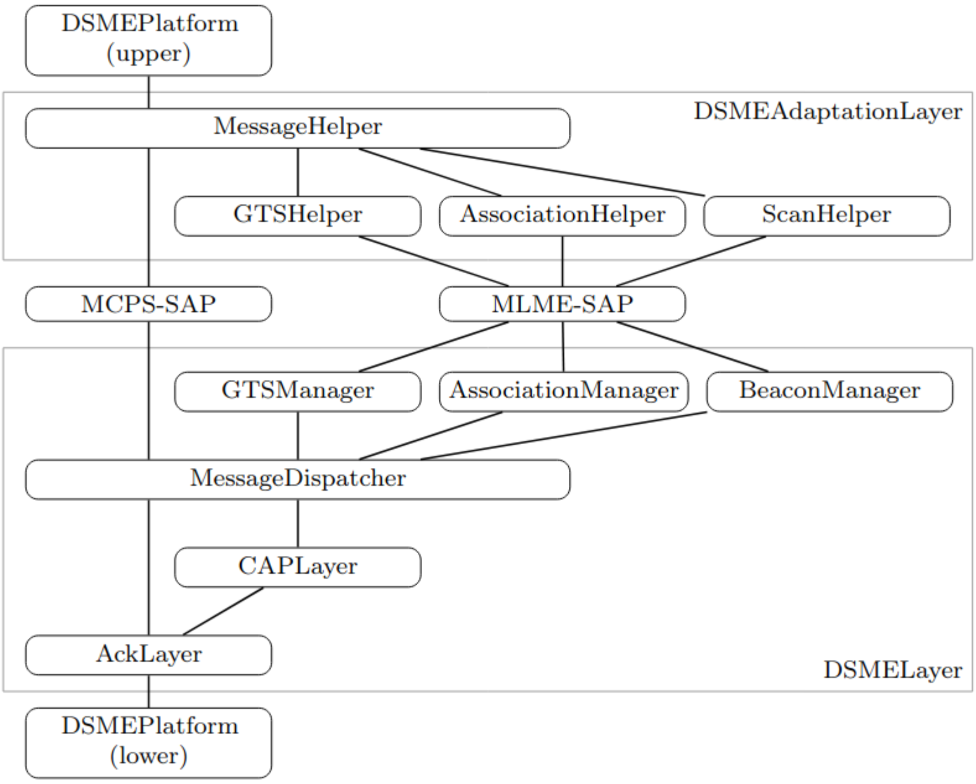


Figure 5.4: openDSME architecture [Köstler et al. 2016]

Chapter 6

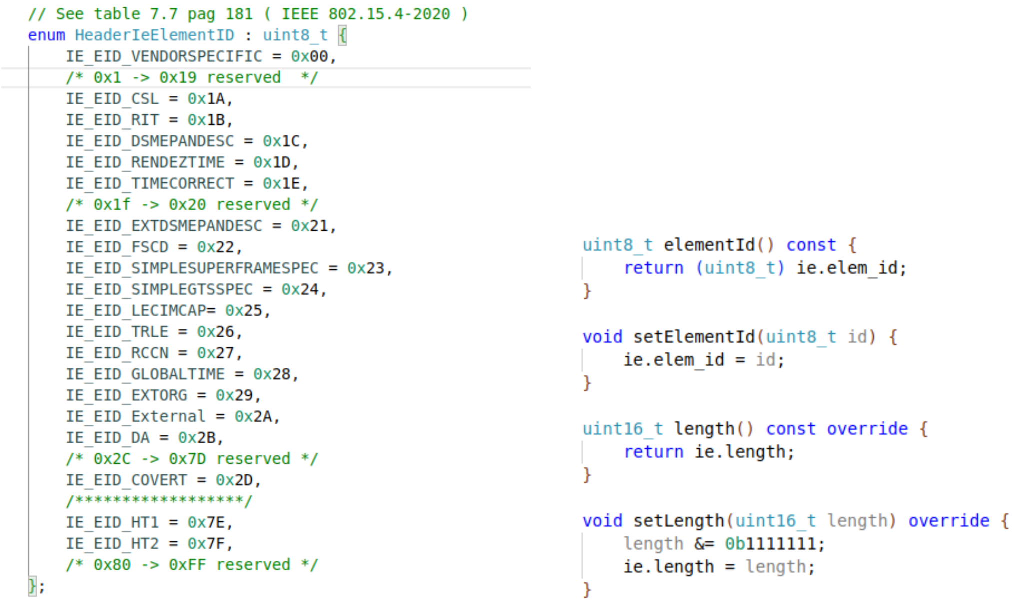
Simulation Setup and Model Extension

This chapter describes the extension of the IEEE 802.15.4 DSME simulation model with IE support.

We believe IEs have great potential to pass covert information. Because IE support was lacking from the OpenDSME model, we decided to carry out its implementation, in order to be able to study the performance limits of such solution.

First, we created an IE library, to manage the IE information, particularly, available IEs, their identification and length. Importantly, the setup of Element IDs enables us to hide 2 bytes of data there(Figure 6.1 - (a)). As there are some reserved Element IDs it was necessary to find out which ones would be free. As result, a specific Element ID was created to use during the performance assessment.

In addition to the IE storage library, we also created a set for interface functions that enable to set and retrieve the length and IE ID of data to be sent (Figure 6.1 - (b)). These functions allow the program to know the Element ID in use and the total length of the IE.



```
// See table 7.7 pag 181 ( IEEE 802.15.4-2020 )
enum HeaderIeElementID : uint8_t {
    IE_EID_VENDORSPECIFIC = 0x00,
    /* 0x1 -> 0x19 reserved  */
    IE_EID_CSL = 0x1A,
    IE_EID_RIT = 0x1B,
    IE_EID_DSMEPANDESC = 0x1C,
    IE_EID_RENDEZTIME = 0x1D,
    IE_EID_TIMECORRECT = 0x1E,
    /* 0x1f -> 0x20 reserved */
    IE_EID_EXTDSMEPANDESC = 0x21,
    IE_EID_FSCD = 0x22,
    IE_EID_SIMPLESUPERFRAMESPEC = 0x23,
    IE_EID_SIMPLEGTSSPEC = 0x24,
    IE_EID_LECIMCAP= 0x25,
    IE_EID_TRLE = 0x26,
    IE_EID_RCCN = 0x27,
    IE_EID_GLOBALTIME = 0x28,
    IE_EID_EXTORG = 0x29,
    IE_EID_External = 0x2A,
    IE_EID_DA = 0x2B,
    /* 0x2C -> 0x7D reserved */
    IE_EID_COVERT = 0x2D,
    /*****************/
    IE_EID_HT1 = 0x7E,
    IE_EID_HT2 = 0x7F,
    /* 0x80 -> 0xFF reserved */
};
```

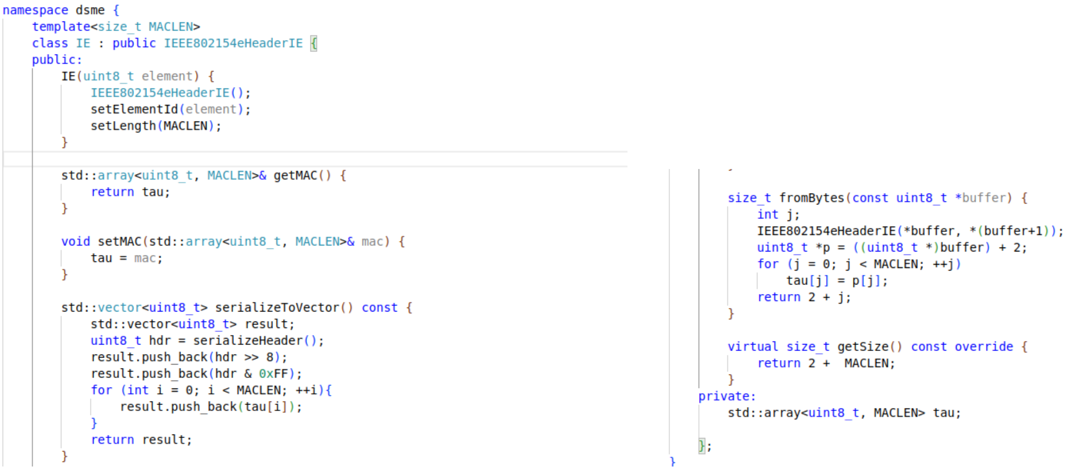
(a)Element ID

```
uint8_t elementId() const {
    return (uint8_t) ie.elem_id;
}

void setElementId(uint8_t id) {
    ie.elem_id = id;
}

uint16_t length() const override {
    return ie.length;
}

void setLength(uint16_t length) override {
    length &= 0b1111111;
    ie.length = length;
}
```

(b) IE's functions

Figure 6.1: IE library

In addition, an auxiliary library was created to assist on the IE data setup for each frame. The node chooses the type of IE, to get the length and the message (Figure 6.2). With that purpose, we created a class named "IE", because with this class which makes it more effective to call the necessary methods when needed.

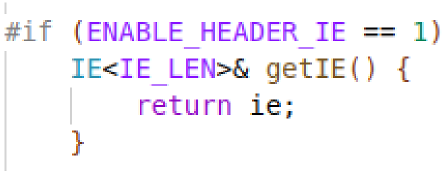


```
namespace dsme {
    template<size_t MACLEN>
    class IE : public IEEE802154eHeaderIE {
    public:
        IE(uint8_t element) {
            IEEE802154eHeaderIE();
            setElementId(element);
            setLength(MACLEN);
        }

        std::array<uint8_t, MACLEN>& getMAC() {
            return tau;
        }

        void setMAC(std::array<uint8_t, MACLEN>& mac) {
            tau = mac;
        }

        std::vector<uint8_t> serializeToVector() const {
            std::vector<uint8_t> result;
            uint8_t hdr = serializeHeader();
            result.push_back(hdr >> 8);
            result.push_back(hdr & 0xFF);
            for (int i = 0; i < MACLEN; ++i){
                result.push_back(tau[i]);
            }
            return result;
        }
```

```
        size_t fromBytes(const uint8_t *buffer) {
            int j;
            IEEE802154eHeaderIE(*buffer, *(buffer+1));
            uint8_t *p = ((uint8_t *)buffer) + 2;
            for (j = 0; j < MACLEN; ++j)
                tau[j] = p[j];
            return 2 + j;
        }

        virtual size_t getSize() const override {
            return 2 +  MACLEN;
        }
    private:
        std::array<uint8_t, MACLEN> tau;

    };
}
```

Figure 6.2: IE class

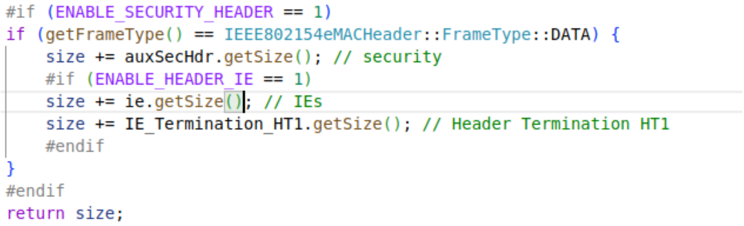
Before using IEs in a message, it is first necessary to check if the IE is used and to do so this check is done as shown in the Figure 6.3.



```
#if (ENABLE_HEADER_IE == 1)
    IE<IE_LEN>& getIE() {
        return ie;
    }
```

Figure 6.3: IE verification

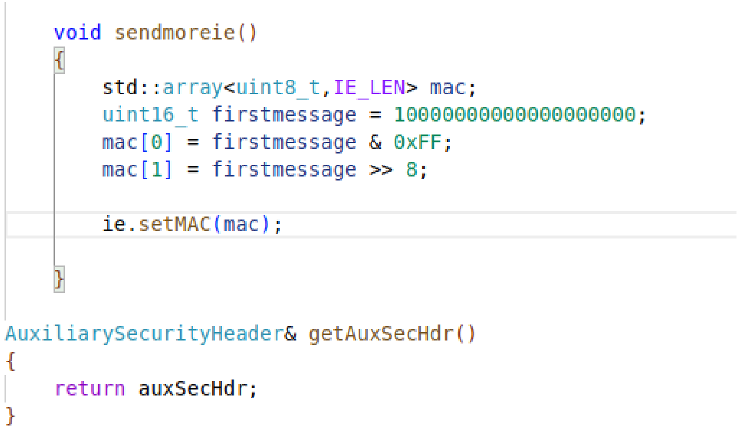
To allocate in the header the respective IE information and to compute the total space of the MAC header it is mandatory to check if the respective IE exists. As shown in the figure 6.4, the total size needed to implement an IE will be the 2 bytes given by the element ID which will be variable content, plus the 16 bytes needed from the IE itself plus 1 byte needed from the security header that is necessary once the IE is used.



```
#if (ENABLE_SECURITY_HEADER == 1)
if (getFrameType() == IEEE802154eMACHeader::FrameType::DATA) {
    size += auxSecHdr.getSize(); // security
    #if (ENABLE_HEADER_IE == 1)
    size += ie.getSize(); // IEs
    size += IE_Termination_HT1.getSize(); // Header Termination HT1
    #endif
}
#endif
return size;
```

Figure 6.4: IE size

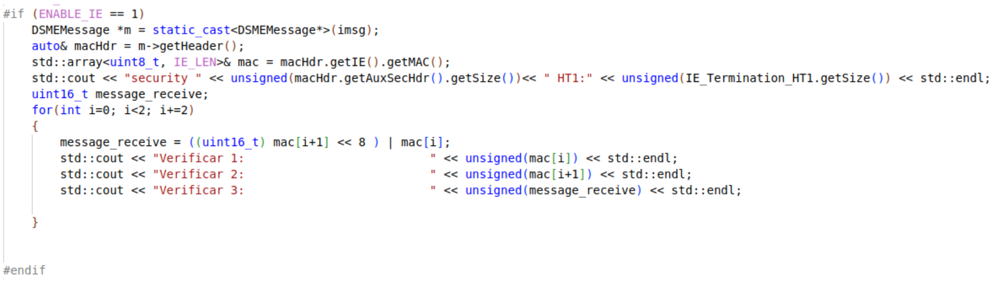
To close this description of the implementation, it is necessary to show how IE are sent and received. First, it is mandatory to send the IE like is shown in the figure 6.5. Unfortunately, it is not possible to send 2 bytes at once with the IE implementation performed, so 1 byte is sent at a time and the message is assembled in the receiver's side. This way you can send the whole message without loss.



```
    void sendmoreie()
    {
        std::array<uint8_t,IE_LEN> mac;
        uint16_t firstmessage = 10000000000000000000;
        mac[0] = firstmessage & 0xFF;
        mac[1] = firstmessage >> 8;

        ie.setMAC(mac);

    }
AuxiliarySecurityHeader& getAuxSecHdr()
{
    return auxSecHdr;
}
```

Figure 6.5: Send IE

On the receiver side, as said before, the message is assembled as shown in the figure 6.6. If the message arrives exactly the same as what was sent then the transmission was successful.



```
#if (ENABLE_IE == 1)
    DSMEMessage *m = static_cast<DSMEMessage*>(imsg);
    auto& macHdr = m->getHeader();
    std::array<uint8_t, IE_LEN>& mac = macHdr.getIE().getMAC();
    std::cout << "security " << unsigned(macHdr.getAuxSecHdr().getSize())<< " HT1:" << unsigned(IE_Termination_HT1.getSize()) << std::endl;
    uint16_t message_receive;
    for(int i=0; i<2; i+=2)
    {
        message_receive = ((uint16_t) mac[i+1] << 8 ) | mac[i];
        std::cout << "Verificar 1:                          " << unsigned(mac[i]) << std::endl;
        std::cout << "Verificar 2:                          " << unsigned(mac[i+1]) << std::endl;
        std::cout << "Verificar 3:                          " << unsigned(message_receive) << std::endl;

    }

#endif
```

Figure 6.6: Receive IE

In short, sending IEs in the MAC header can be translated into the following pseudocode. First it is necessary to check if the IE exists. If it does, then we need to write the information to pass and increase the size of the MAC header. When this is done, the IE is sent. On the receiver side, it checks if IE exists and if so it receives the IE and then reads the message sent.

Algorithm 1 Send IE

```
if $IE$ is not 0 then
    $IE \leftarrow message$
    $headersize \leftarrow headersize + IE$
end if
Send Message
```

Algorithm 2 Receive IE

```
if $IE$ is not 0 then
Receive Message
    $IE \leftarrow message$
end if
```

Chapter 7

Performance Analysis

This Chapter reports on the performance assessment of a set of network storage covert channel opportunities.

7.1 Simulation Setup

To carry out the simulation assessment we relied on the openDSME model [Köstler et al. 2016] supported by the INET framework and runned in the OMNeT++ event. We considered a star network topology with 11 nodes, organized in a circular shape, where the distance between the exterior nodes and the node[0], i.e. the PAN Coordinator, is approximately similar. This node, being placed at the center, acts as sink as presented in figure 7.1. All the data sender nodes will be scheduled according to an incrementing schedule (node[1] gets slot 1, node[2] gets slot 2, etc.). This data begins being generated in the 30th second of the simulation to ensure all the network is properly setup and slots are available. From this point onward, the simulation will run for 200 seconds, recording several metrics of interest described below. The default setup of the simulation (unless changed for a metric comparison) was a packet payload length of 1 byte, a rate of 0.01 seconds per packet created and a channel bitrate of 250k bits per second. In terms of the protocol standard metrics, the beacon order is set to 6, multi superframe order is 4 and the superframe order is also 4.

The metric under test was mostly the Covert Capacity, which reflects the amount of covert data that can be communicated in a specific portion of time. The integrity

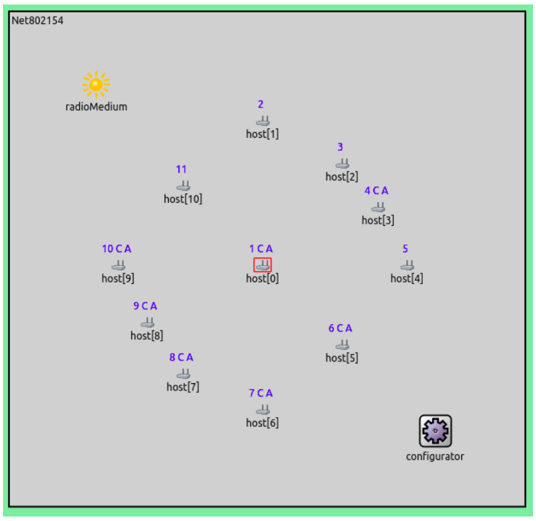


Figure 7.1: OMNeT++ Simulator Model

of the data is also tested in order to completely assess the capabilities of steganography. Regarding data extraction, an algorithm was setup to retrieve analysable data from the simulations. First, is was embedded in the code an input/output script in C++ that retrieved several raw data, including the number of packets per simulation time to a .csv file. After this, data was analysed using Excel.

It was considered MO = SO for all test cases and BO = SO+1, so to guarantee that we allocate two superframes per multi-superframe. To evaluate the behaviour of the covert channel we consider varying traffic rates from 0.01 to 1 seconds, and the MO, SO, BO combinations represented in table 7.1

	SO	MO	BO
A	4	5	6
B	5	6	7
C	6	7	8
D	7	8	9
E	8	9	10
F	10	11	12
G	12	13	14

Table 7.1: Network DSME configurations

7.2 Analysis of IE Exfiltration Capacity in Beacons

In this section, we analyze the exfiltration capacity of Beacon enabled IEs under different network scenarios. We setup a single IE of 2 bytes in the periodic Beacons.

The graph in figure 7.2, depicts the variation of the covert data transmitted using the IE in Bytes per second. As expected, the covert channel capacity is greatly dependant of the beacon periodicity, which is set by the SO. Parameters like number of nodes, traffic rate and packet size won't affect the number of bytes sent. As expected, with a lower SO it is possible to send more covert traffic due to more frequent beacons. One of the greatest advantages of embedding IEs in the beacons instead of data frames, is that these are broadcasted, which amplifies the reach of the covert information.

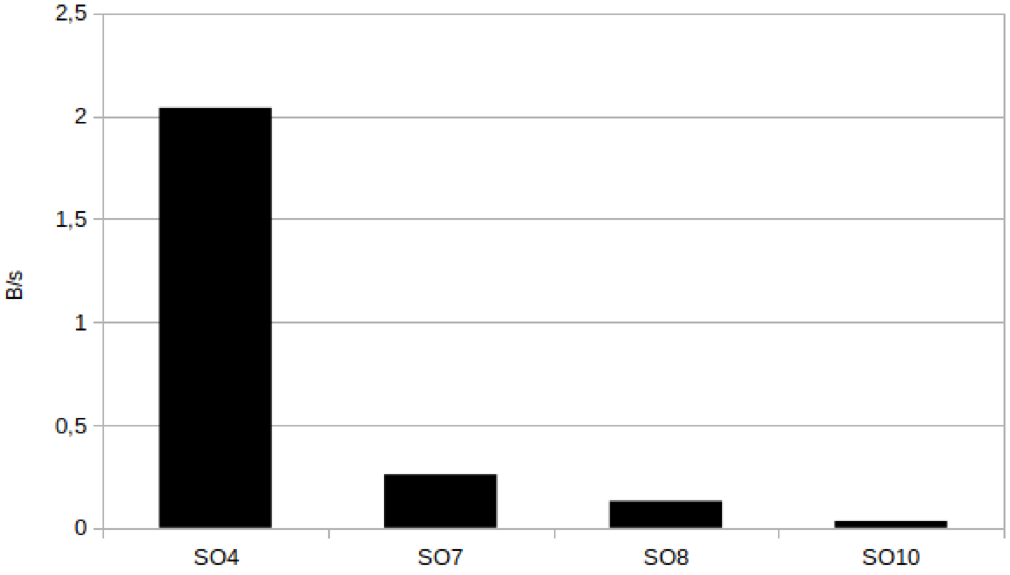


Figure 7.2: Difference between SO

In figure 7.3 we can be seen the cumulative covert traffic sent through the network as simulation time progresses.

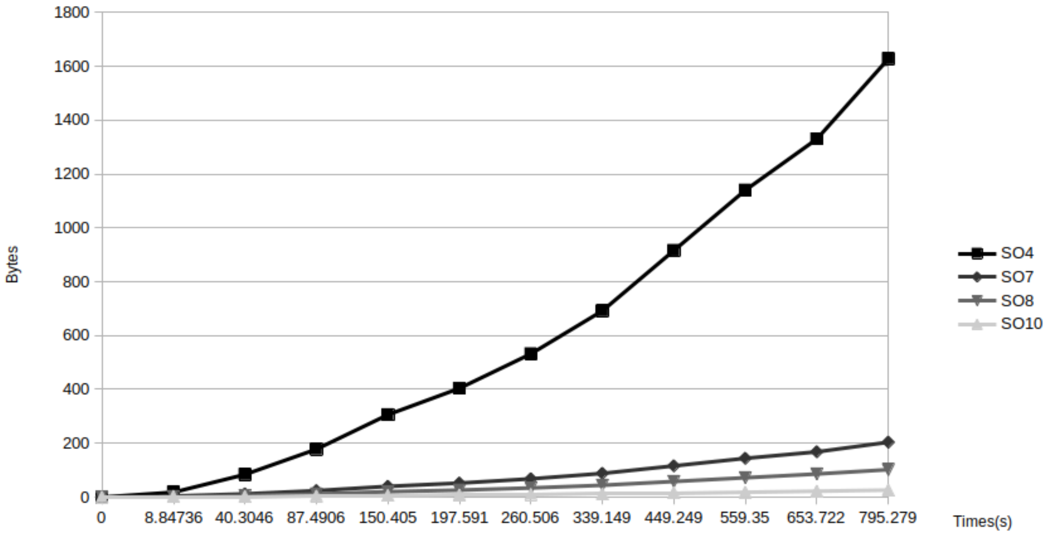


Figure 7.3: Difference between SO

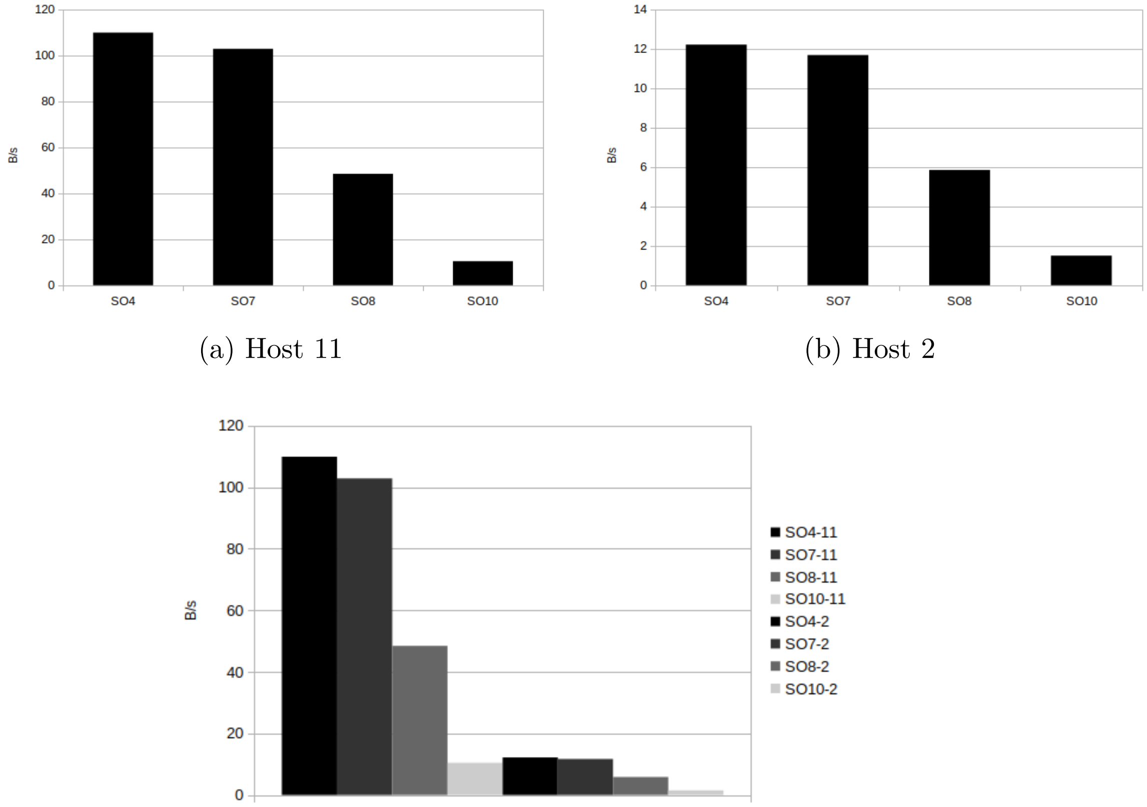
The other advantage of sending IEs in the beacon message is the MAC header doesn't increase visibly in size, so the message can be sent in a less conspicuous fashion.

7.3 Analysis of IE Exfiltration Capacity in Data Frames

When the IE is set to 1 in the MAC header the total packet length will increase, because a IE has at least 16 bytes of size. Despite this disadvantage, with IEs in data frames we can send more than one element ID so it will support more traffic. As we turn on IE support, the MAC header now has at least 19 extra bytes, which makes it quite perceptible that there is more information being pass, even though only 2 bytes are important, since that's where the desired information goes. If IEs are already under use, then this will not be so noticeable.

7.3.1 Impact of SO

In order to analyse the impact of the SO in the covert capacity, simulations were carried out at different SO settings while maintaining the same traffic generation rates. We further carried out analysis for a network with 2 and 11 nodes. Figure 7.4 presents the results. Clearly, lower SO generates more frequent superframes, and in consequence more frequent slots, which increments the number of transmission opportunities. Interestingly, with only two nodes, we can exfiltrate 12 bytes of covert information per second by relying in only one IE. Naturaly, as we expand to 11 nodes, the amount of information available in the network grows significantly.

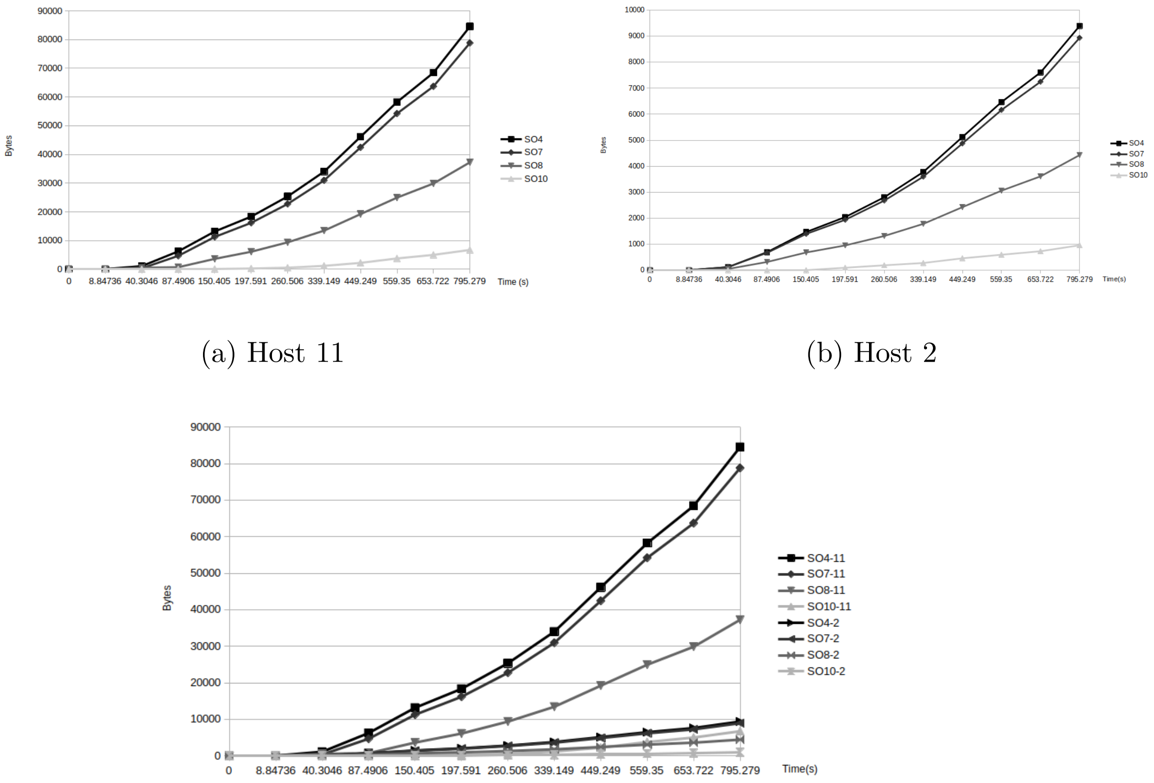


(a) Host 11

(b) Host 2

Figure 7.4: Comparison between nodes

This factor can be noticed in figure 7.5. As expected, the greater the number of nodes in a network, the greater the amount of information exfiltrated over time.



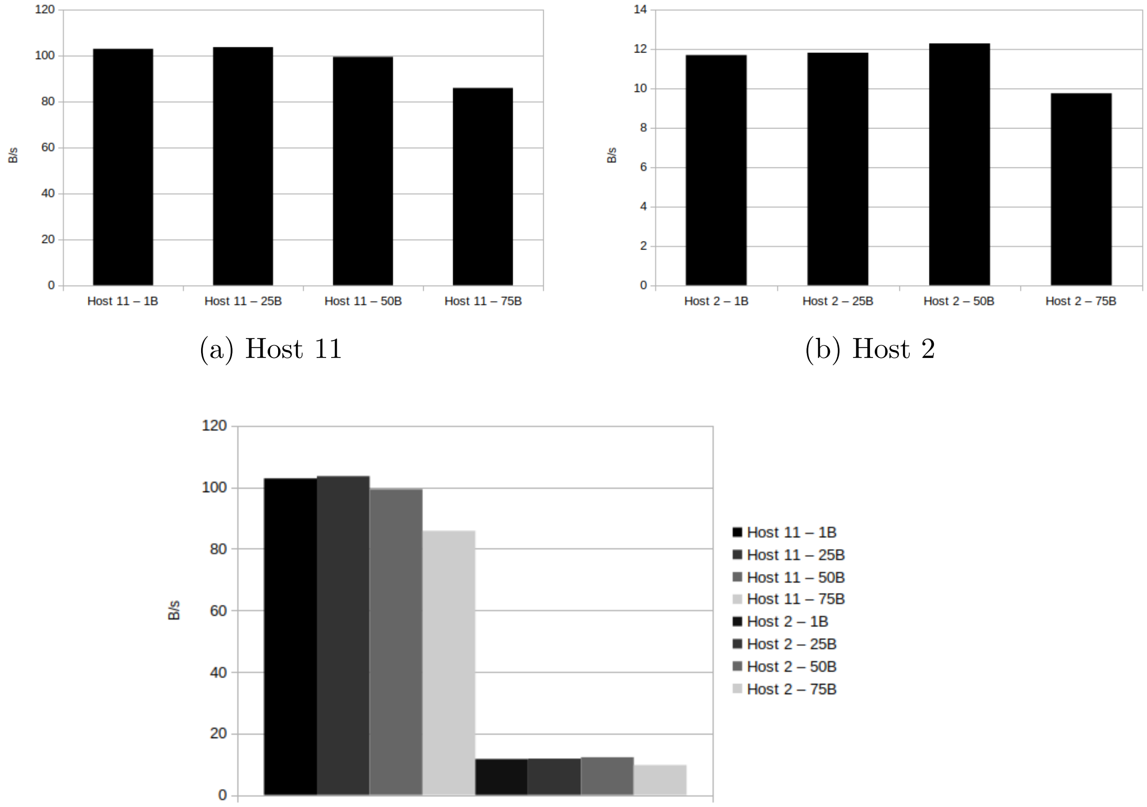
(a) Host 11

(b) Host 2

Figure 7.5: Comparison between nodes

7.3.2 Impact of packet size

After analyzing the impact of the SO, it is interesting to analyse the impact of different packet size. We change the payload of each packet and repeat the experiments. The graphs in figure 7.6 are referent to simulations with 11 nodes and with 2 nodes. The SO is set to 7, a more balanced beacon periodicity. Clearly, even with that SO value, we can see that maximizing the packet length, by increasing the payload to 75 bytes, effectively reduces the amount of exfiltrated data. The reason is that less packet will fit the available slot size. This reduces the amount of covert transmission opportunities.

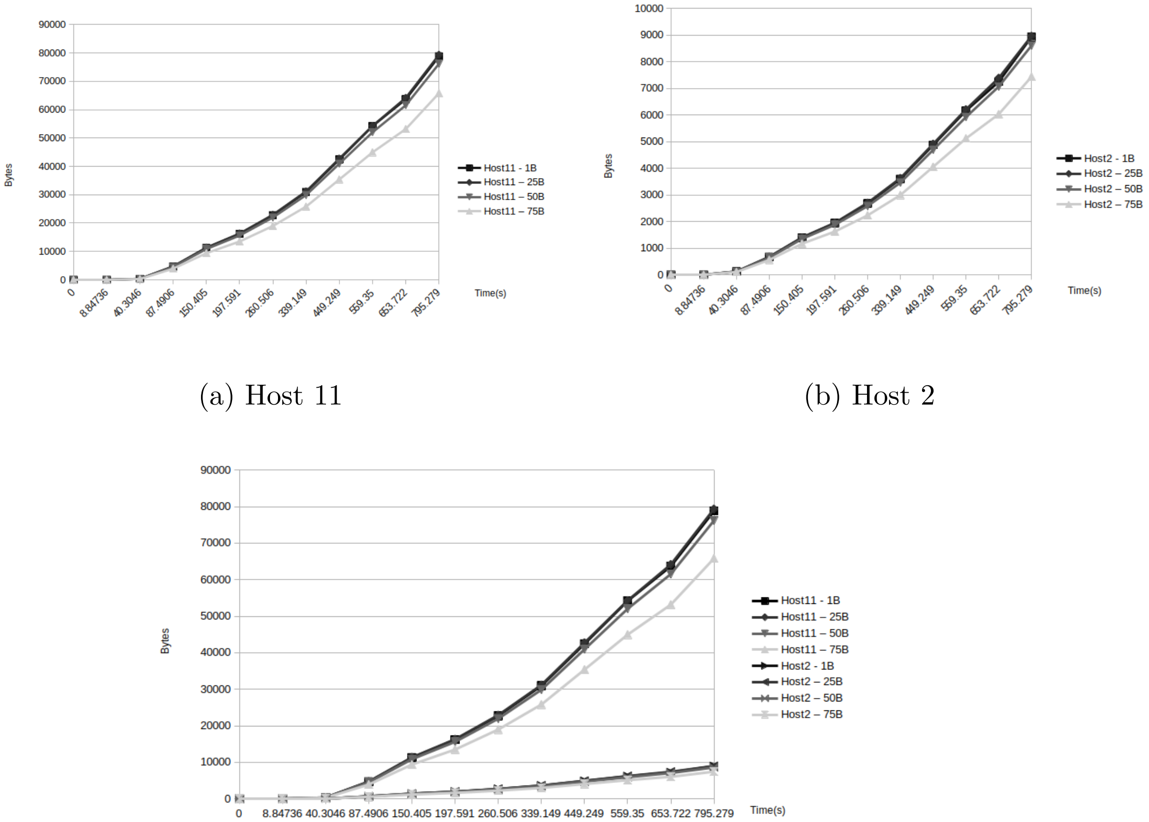


(a) Host 11

(b) Host 2

Figure 7.6: Comparison between nodes

The growing of bytes sent can be analysed in the graphs in figure 7.7



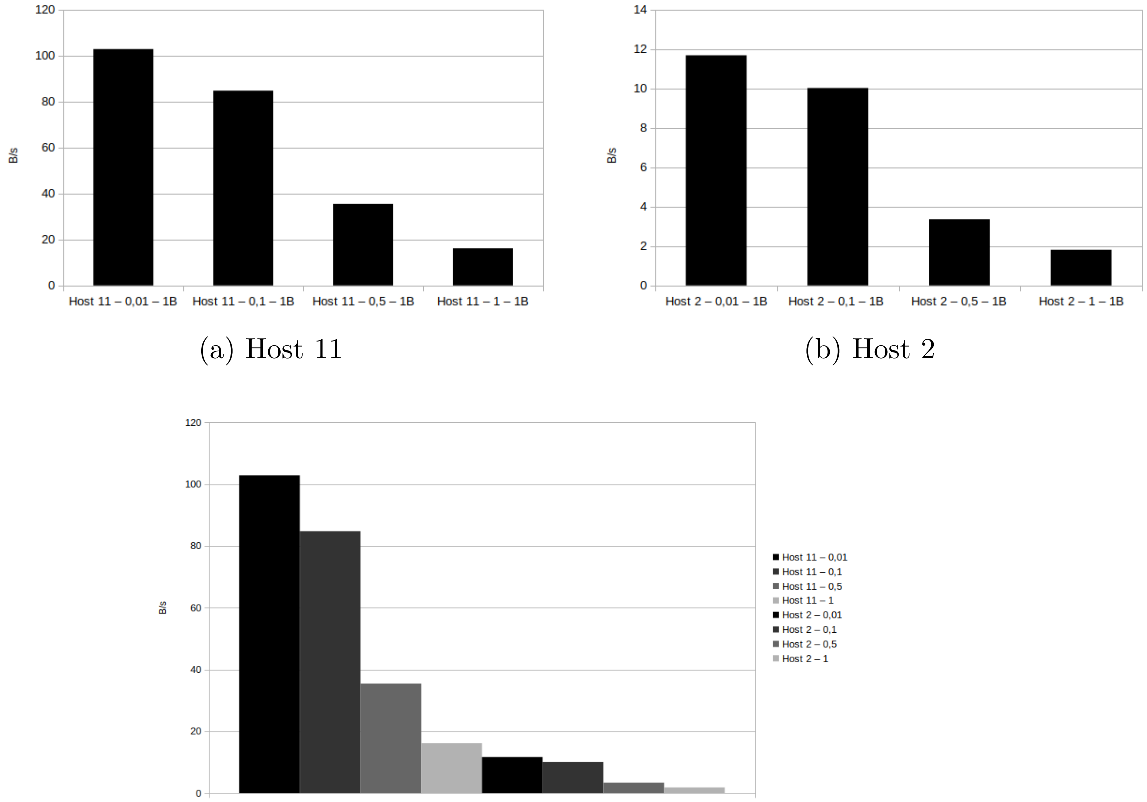
(a) Host 11

(b) Host 2

Figure 7.7: Comparison between nodes

7.3.3 Impact of traffic rate

Another interesting factor to be analysed is traffic rate generation, because the higher the traffic rate, the lower the chances of sending bytes in the messages. This can be seen in figure 7.8

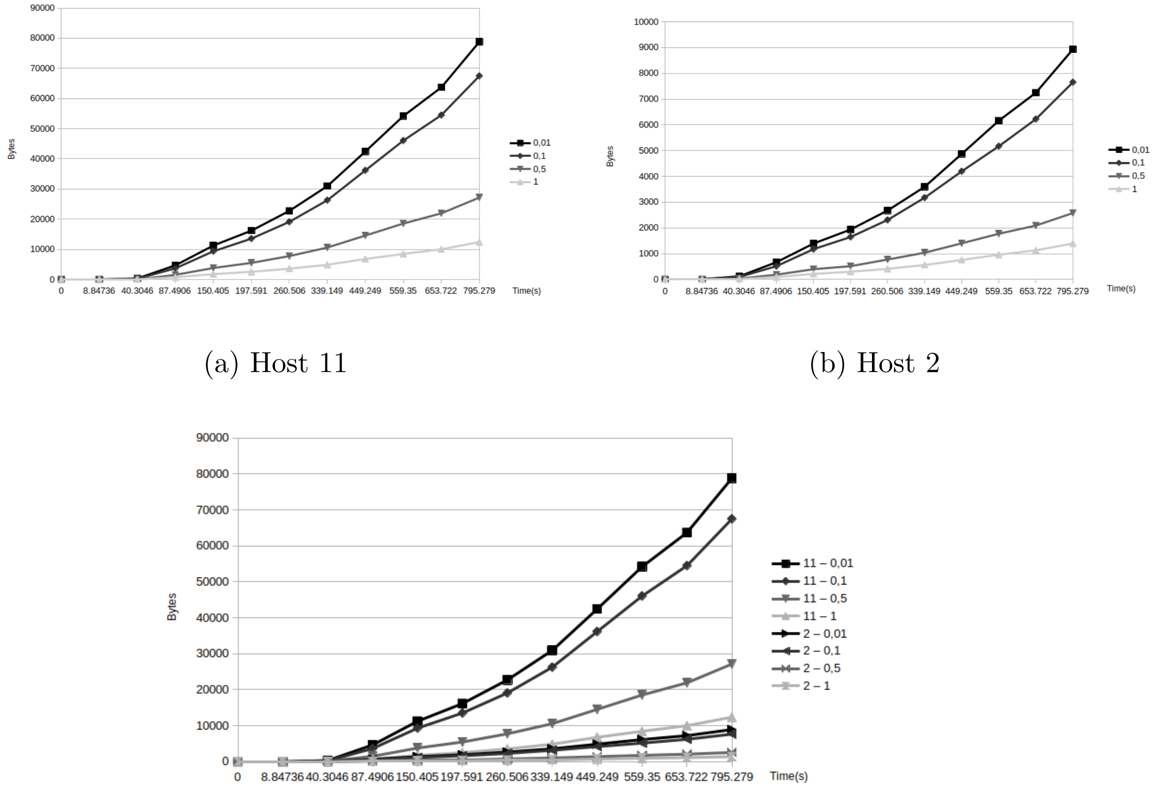


(a) Host 11

(b) Host 2

Figure 7.8: Comparison between nodes

It is also possible to observe the growth of bytes sent over the past time in the graphs of figure 7.9.



(a) Host 11

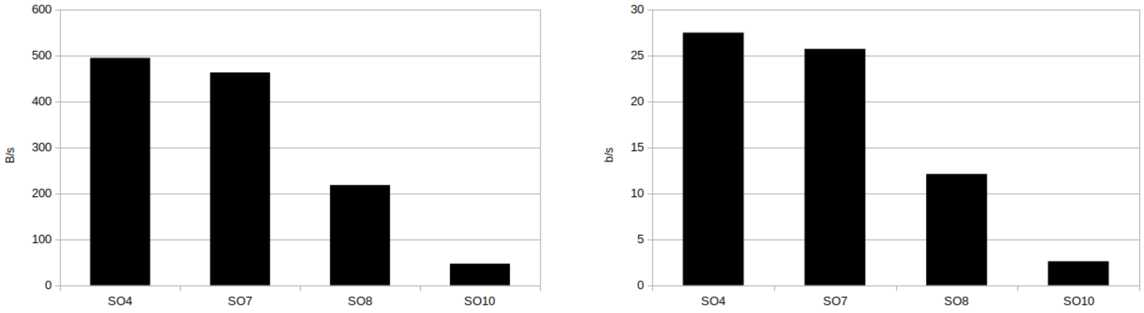
(b) Host 2

Figure 7.9: Comparison between nodes

As shown, as simulation time progresses, for higher amounts of traffic the higher covert traffic volumes can be reached. Interestingly, with only two nodes and one IE, generating 1 packet of data per second is enough to reach over 1KB after 13 minutes.

7.4 Other Alternatives

In the table 4.2 it is possible to see other alternatives is sending data with steganography. In order to have an in-depth study it was studied the opportunity if sending data in the MAC command frame and in the Multipurpose frame (Figure 7.10. The major difference between them is the fact that the first one can sent 73 bits (about 9 bytes) and the second one can sent 4 bits only.

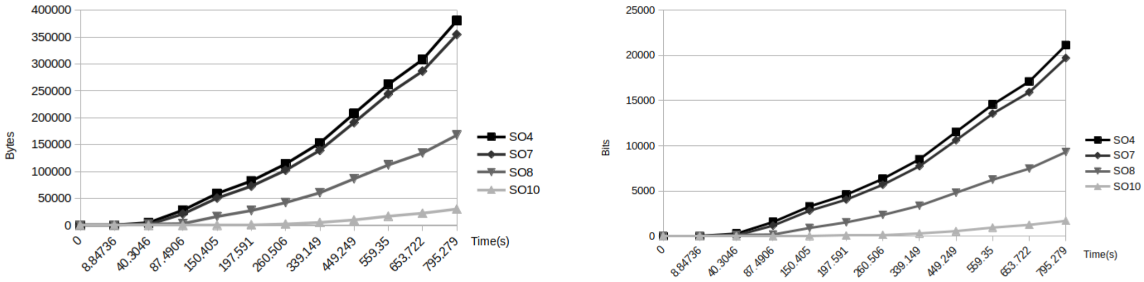


(a) MAC Command frame

(b) Multipurpose frame

Figure 7.10: Other Alternatives comparison

In the next graphs (figure 7.11 can be seen the cumulative exfiltration capacity during simulation time.



(a) MAC Command frame

(b) Multipurpose frame

Figure 7.11: Other Alternatives comparison

Chapter 8

Conclusions

8.1 Overview

Regarding the main Thesis' objectives described in the initial chapter, all were successfully accomplished. The standard was studied and over-viewed, in order to explore the possibility of covert channel implementations. A set of techniques was chosen for simulation assessment with particular emphasis on IE usage.

It is clear that with just two bytes per IE one can exfiltrate a significant amount of information. We have shown that some network parameters such as SO, packet size and most importantly generated traffic can affect the capacity of such covert techniques.

8.2 Future Work

The possibility of developing an authentication methodology using covert channels has already been studied and analysed in the literature, with the help of multiple techniques to authenticate a network node. We expect such steganography opportunities can be used in benfic ways to support authentication and authorization mechanisms for this specific protocol. The contents of this Thesis are fundamental to enable such mechanisms, by leveraging already available storage covert channel opportunities.

Bibliography

"A Novel Timing-based Network Covert Channel Detection Method" (Oct. 2019). In: *Journal of Physics: Conference Series* 1325 (1), p. 012050. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1325/1/012050. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1325/1/012050.

"A Timing Channel Spyware for the CSMA/CA Protocol" (Mar. 2013). In: *IEEE Transactions on Information Forensics and Security* 8 (3), pp. 477–487. ISSN: 1556-6013. DOI: 10.1109/TIFS.2013.2238930. URL: http://ieeexplore.ieee.org/document/6410028/.

Anderson, Ross and Fabien Petitcolas (Dec. 1998). "On The Limits of Steganography". In: *IEEE Journal on Selected Areas in Communications* 16, pp. 474–481. DOI: 10.1109/49.668971.

Artz, D. (2001). "Digital steganography: hiding data within data". In: *IEEE Internet Computing* 5.3, pp. 75–80. DOI: 10.1109/4236.935180.

Caviglione, Luca (Jan. 2021a). "Trends and Challenges in Network Covert Channels Countermeasures". en. In: *Applied Sciences* 11.4. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 1641. ISSN: 2076-3417. DOI: 10.3390/app11041641. URL: https://www.mdpi.com/2076-3417/11/4/1641 (visited on 02/18/2022).

— (2021b). "Trends and Challenges in Network Covert Channels Countermeasures". In: *Applied Sciences* 11.4. ISSN: 2076-3417. DOI: 10.3390/app11041641. URL: https://www.mdpi.com/2076-3417/11/4/1641.

Caviglione, Luca, Alessio Merlo, and Mauro Migliardi (May 2018). "Covert Channels in IoT Deployments Through Data Hiding Techniques". In: *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 559–563. DOI: 10.1109/WAINA.2018.00144.

"Covert timing channel detection method based on time interval and payload length analysis" (2020). In: *Computers and Security* 97, p. 101952. ISSN: 01674048. DOI: 10.1016/j.cose.2020.101952. URL: https://doi.org/10.1016/j.cose.2020.101952.

"Covert Timing Channels for IoT over Mobile Networks" (Dec. 2018). In: *IEEE Wireless Communications* 25 (6), pp. 38–44. ISSN: 1536-1284. DOI: 10.1109/MWC.2017.1800062. URL: https://ieeexplore.ieee.org/document/8600755/.

Cunha, André (2007). “On the use of IEEE 802.15.4/ZigBee as federating communication protocols for Wireless Sensor Networks”. In: accessed 24 august 2022.

“Department of Defense Trusted Computer System Evaluation Criteria” (1985). In: *The ‘Orange Book’ Series*. accessed 11 september 2022. London: Palgrave Macmillan UK, pp. 1–129. ISBN: 978-1-349-12020-8. DOI: `10.1007/978-1-349-12020-8_1`. URL: `https://doi.org/10.1007/978-1-349-12020-8_1`.

Elsadig, Muawia A. and Yahia A. Fadlalla (Dec. 2018). “Packet Length Covert Channels Crashed”. In: *Journal of Computer Science and Computational Mathematics*, pp. 59–66. ISSN: 22318879. DOI: `10.20967/jcscm.2018.04.001`. URL: `https://www.jcscm.net/cms/?action=showpaper&id=2050637`.

Elshoush, Huwaida, Mahmoud Mahmoud, and Abdelrahman Altigani (Feb. 2022). “A new high capacity and secure image realization steganography based on ASCII code matching”. In: *Multimedia Tools and Applications* 81. DOI: `10.1007/s11042-021-11741-y`.

Frustaci, Mario et al. (Aug. 2018). “Evaluating Critical Security Issues of the IoT World: Present and Future Challenges”. en. In: *IEEE Internet of Things Journal* 5.4, pp. 2483–2495. ISSN: 2327-4662, 2372-2541. DOI: `10.1109/JIOT.2017.2767291`. URL: `https://ieeexplore.ieee.org/document/8086136/` (visited on 02/18/2022).

Gamundani, Attlee, Amelia Phillips, and Hippolyte Muyingi (Nov. 2018). “An Overview of Potential Authentication Threats and Attacks on Internet of Things(IoT): A Focus on Smart Home Applications”. In: DOI: `10.1109/Cybermatics_2018.2018.00043`.

Hwang, Kwang-il and Sung-wook Nam (May 2014). “Analysis and Enhancement of IEEE 802.15.4e DSME Beacon Scheduling Model”. en. In: *Journal of Applied Mathematics* 2014. Publisher: Hindawi, e934610. ISSN: 1110-757X. DOI: `10.1155/2014/934610`. URL: `https://www.hindawi.com/journals/jam/2014/934610/` (visited on 02/13/2022).

“IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4” (2006). In: *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*. accessed 24 august 2022, pp. 1–320. DOI: `10.1109/IEEESTD.2006.232110`.

“IEEE Standard for Low-Rate Wireless Networks” (2020). In: *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*. accessed 28 august 2022, pp. 1–800. DOI: `10.1109/IEEESTD.2020.9144691`.

Johnson, Daryl et al. (2010). “Covert channels in the HTTP network protocol: Channel characterization and detecting man-in-the-middle attacks”. In: p. 11.

Kaur, Sumeet, Savina Bansal, and R. K. Bansal (2014). “Steganography and classification of image steganography techniques”. In: *2014 International Conference* *on Computing for Sustainable Global Development (INDIACom)*, pp. 870–875. DOI: 10.1109/IndiaCom.2014.6828087.

Köstler, Maximilian et al. (Sept. 2016). "Towards an Open Source Implementation of the IEEE 802.15.4 DSME Link Layer". In: *Proceedings of the 15. GI/ITG KuVS Fachgespräch Sensornetze*. Ed. by Juergen Scholz and Alexander von Bodisco. accessed 12 september 2022. Augsburg, Germany: University of Applied Sciences Augsburg, Dept. of Computer Science, p. 4.

kourzanov (2022). *EIT-ICT-RICH/ns-3-dev-TSCH*. accessed 28 august 2022. URL: https://github.com/EIT-ICT-RICH/ns-3-dev-TSCH (visited on 02/20/2022).

Kurunathan, John Harrison (2021). "Improving QoS for IEEE 802.15.4e DSME Networks". In: accessed 28 august 2022, pp. 1–225.

Lin, Jie et al. (Oct. 2017). "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications". en. In: *IEEE Internet of Things Journal* 4.5, pp. 1125–1142. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2017.2683200. URL: https://ieeexplore.ieee.org/document/7879243/ (visited on 02/18/2022).

Liu, Zhihong et al. (Apr. 2020). "Covert Wireless Communication in IoT Network: From AWGN Channel to THz Band". In: *IEEE Internet of Things Journal* 7.4. Conference Name: IEEE Internet of Things Journal, pp. 3378–3388. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.2968153.

Martins, David and Herve Guyennet (2010). "Attacks with Steganography in PHY and MAC Layers of 802.15.4 Protocol". In: *2010 Fifth International Conference on Systems and Networks Communications*. accessed 24 august 2022, pp. 31–36. DOI: 10.1109/ICSNC.2010.11.

Masood, Tariq and Paul Sonntag (Oct. 2020). "Industry 4.0: Adoption challenges and benefits for SMEs". en. In: *Computers in Industry* 121, p. 103261. ISSN: 0166-3615. DOI: 10.1016/j.compind.2020.103261. URL: https://www.sciencedirect.com/science/article/pii/S0166361520304954 (visited on 02/15/2022).

Meyer, Florian, Ivonne Mantilla, and Volker Turau (Apr. 2020). "Sending Multiple Packets per Guaranteed Time Slot in IEEE 802.15.4 DSME: Analysis and Evaluation". In: accessed 30 august 2022. DOI: 10.1002/itl2.167.

Millen, Jonathan (Feb. 1999). "20 years of covert channel modeling and analysis". In: accessed 11 september 2022, pp. 113–114. ISBN: 0-7695-0176-1. DOI: 10.1109/SECPRI.1999.766906.

Nain, Ajay Kumar and P. Rajalakshmi (2017). "A reliable covert channel over IEEE 802.15.4 using steganography". In: *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, pp. 711–716. DOI: 10.1109/WF-IoT.2016.7845486.

notes, Eletronics (n.d.). *IEEE 802.15.4 Standard: a tutorial / primer.* accessed 24 august 2022. URL: https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/basics-tutorial-primer.php.

Omoniwa, Babatunji et al. (2019). “Fog/Edge Computing-Based IoT (FECIoT): Architecture, Applications, and Research Issues”. In: *IEEE Internet of Things Journal* 6.3, pp. 4118–4149. DOI: 10.1109/JIOT.2018.2875544.

Osterlind, Fredrik et al. (Nov. 2006). “Cross-Level Sensor Network Simulation with COOJA”. en. In: *Proceedings. 2006 31st IEEE Conference on Local Computer Networks.* ISSN: 0742-1303. Embassy Suites Hotel, Tampa, FL, USA: IEEE, pp. 641–648. ISBN: 978-1-4244-0418-6. DOI: 10.1109/LCN.2006.322172. URL: http://ieeexplore.ieee.org/document/4116633/ (visited on 02/13/2022).

otosection (n.d.). *Ns3 Projects Network Simulation Projects Ns3 Code.* accessed 14 setptember 2022. URL: http://dubaikhalifas.com/.

Piscitello, Dave (2016). *What Is an Internet Covert Channel?* accessed 11 september 2022. URL: https://www.icann.org/en/blogs/details/what-is-an-internet-covert-channel-29-8-2016-en.

Ramonet, Alberto Gallegos and Taku Noguchi (2019). “IEEE 802.15.4 Historical Evolution and Trends”. In: *2019 21st International Conference on Advanced Communication Technology (ICACT).* accessed 24 august 2022, pp. 351–359. DOI: 10.23919/ICACT.2019.8702040.

— (2020). “IEEE 802.15.4 Now and Then: Evolution of the LR-WPAN Standard”. In: *2020 22nd International Conference on Advanced Communication Technology (ICACT).* accessed 24 august 2022, pp. 1198–1210. DOI: 10.23919/ICACT48636.2020.9061514.

Rowland, Craig H. (May 1997). “Covert channels in the TCP/IP protocol suite”. In: *First Monday* 2.5. accessed 11 september 2022. DOI: 10.5210/fm.v2i5.528. URL: https://firstmonday.org/ojs/index.php/fm/article/view/528.

Screenshots (n.d.). urlhttps://omnetpp.org/intro/screenshots. accessed 24 august 2022.

Shih, Chao-Fang, Ariton E. Xhafa, and Jianwei Zhou (June 2015). “Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks”. In: *2015 IEEE International Conference on Communications (ICC).* ISSN: 1938-1883, pp. 6494–6499. DOI: 10.1109/ICC.2015.7249359.

Simmons, Gustavus J. (1984). “The Prisoners’ Problem and the Subliminal Channel”. en. In: *Advances in Cryptology: Proceedings of Crypto 83.* Ed. by David Chaum. accessed 11 september 2022. Boston, MA: Springer US, pp. 51–67. ISBN: 978-1-4684-4730-9. DOI: 10.1007/978-1-4684-4730-9_5. URL: https://doi.org/10.1007/978-1-4684-4730-9_5 (visited on 02/15/2022).

Statista (2022). *Number of IoT devices 2015-2025*. en. URL: `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/` (visited on 02/15/2022).

"Steganography in MAC layers of 802.15.4 protocol for securing wireless sensor networks" (2010). In: *Proceedings - 2010 2nd International Conference on Multimedia Information Networking and Security, MINES 2010*, pp. 824–828. DOI: `10.1109/MINES.2010.175`.

"TACAN: Transmitter Authentication through Covert Channels in Controller Area Networks" (Mar. 2019). In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 23–34. URL: `http://arxiv.org/abs/1903.05231`.

Tahmasbi, Fatemeh, Neda Moghim, and Mojtaba Mahdavi (Nov. 2016). "Adaptive ternary timing covert channel in IEEE 802.11". In: *Security and Communication Networks* 9 (16), pp. 3388–3400. ISSN: 19390114. DOI: `10.1002/sec.1545`. URL: `https://onlinelibrary.wiley.com/doi/10.1002/sec.1545`.

Tian, Jing et al. (Aug. 2020). "A Survey of Key Technologies for Constructing Network Covert Channel". In: *Security and Communication Networks* 2020. Publisher: Hindawi. ISSN: 1939-0114. DOI: `10.1155/2020/8892896`. URL: `https://www.hindawi.com/journals/scn/2020/8892896/` (visited on 02/13/2022).

Tuptuk, Nilufer and Stephen Hailes (Mar. 2015). "Covert channel attacks in pervasive computing". In: *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 236–242. DOI: `10.1109/PERCOM.2015.7146534`.

Valero, M., A. Bourgeois, and R. Beyah (2010). "DEEP: A Deployable Energy Efficient 802.15.4 MAC Protocol for Sensor Networks". In: *2010 IEEE International Conference on Communications*. accessed 24 august 2022, pp. 1–6. DOI: `10.1109/ICC.2010.5501955`.

Varga, András (2003). *OMNeT++ Manual*. 2.3. accessed 12 september 2022. URL: `http://src.gnu-darwin.org/ports/science/omnetpp/work/omnetpp-2.3p1/doc/usman.pdf` (visited on 02/13/2022).

VOTIRO (n.d.). *Image Steganography Example: How I Created an Attack*. accessed 14 setptember 2022. URL: `https://votiro.com/%5Cblog/image-steganography-example-%5Chow-i-created-an-attack/`.

Wendzel, Steffen et al. (Apr. 2015). "A Pattern-based Survey and Categorization of Network Covert Channel Techniques". In: *ACM Computing Surveys* 47.3. arXiv: 1406.2901, pp. 1–26. ISSN: 0360-0300, 1557-7341. DOI: `10.1145/2684195`. URL: `http://arxiv.org/abs/1406.2901` (visited on 02/18/2022).

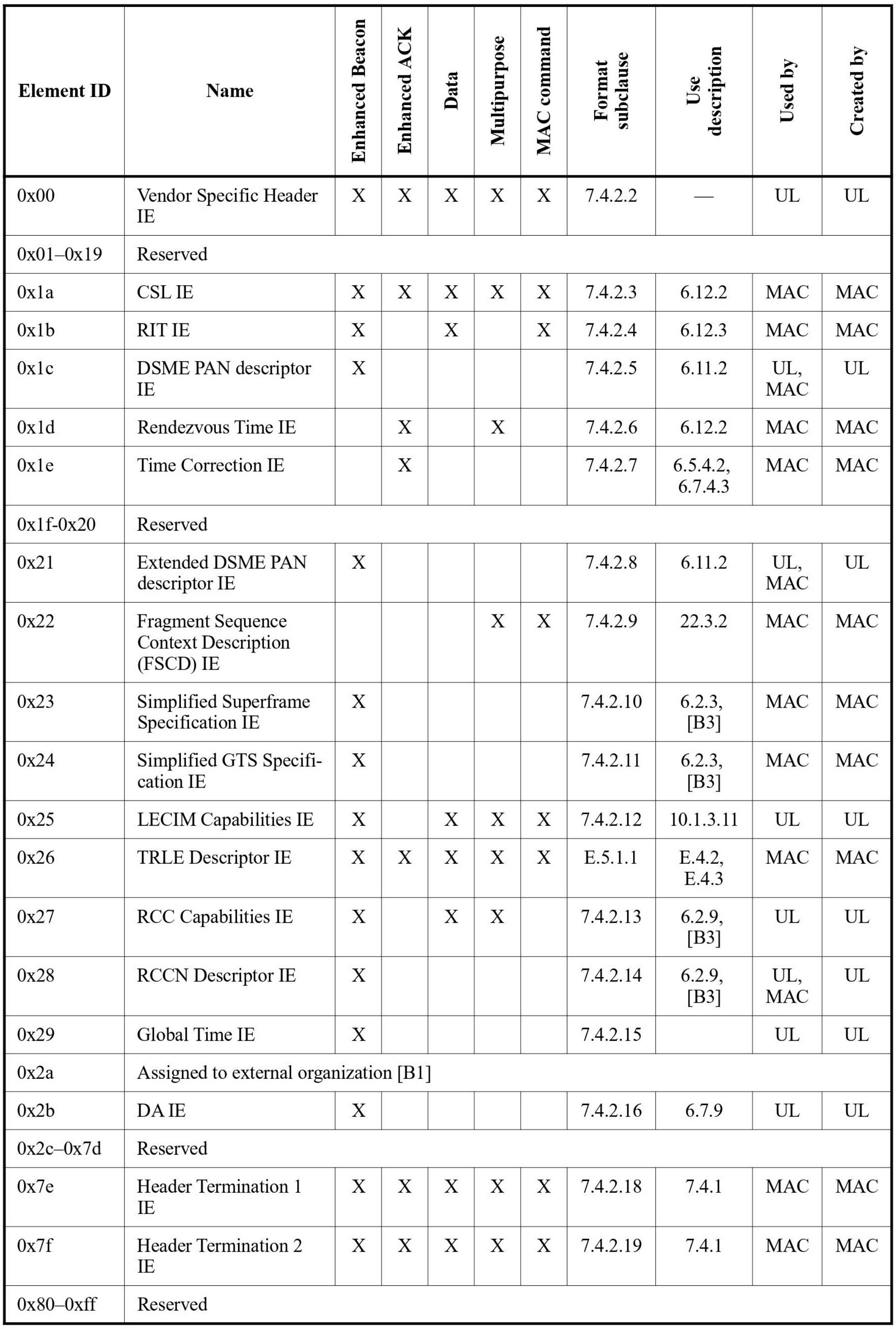
What Is INET Framework?r (n.d.). urlhttps://inet.omnetpp.org/Introduction. accessed 24 august 2022.

Zander, Sebastian, Grenville Armitage, and Philip Branch (Jan. 2007). “Covert channels in the IP time to live field”. In: accessed 11 september 2022.

Zhang, Xiaosong et al. (Jan. 2019). “A packet-reordering covert channel over VoLTE voice and video traffics”. In: *Journal of Network and Computer Applications* 126, pp. 29–38. ISSN: 10848045. DOI: 10.1016/j.jnca.2018.11.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S1084804518303527.

Appendix A

Element IDs for Header IEs



Element ID	Name	Enhanced Beacon	Enhanced ACK	Data	Multipurpose	MAC command	Format subclause	Use description	Used by	Created by
0x00	Vendor Specific Header IE	X	X	X	X	X	7.4.2.2	—	UL	UL
0x01–0x19	Reserved									
0x1a	CSL IE	X	X	X	X	X	7.4.2.3	6.12.2	MAC	MAC
0x1b	RIT IE	X		X		X	7.4.2.4	6.12.3	MAC	MAC
0x1c	DSME PAN descriptor IE	X					7.4.2.5	6.11.2	UL, MAC	UL
0x1d	Rendezvous Time IE		X		X		7.4.2.6	6.12.2	MAC	MAC
0x1e	Time Correction IE		X				7.4.2.7	6.5.4.2, 6.7.4.3	MAC	MAC
0x1f-0x20	Reserved									
0x21	Extended DSME PAN descriptor IE	X					7.4.2.8	6.11.2	UL, MAC	UL
0x22	Fragment Sequence Context Description (FSCD) IE				X	X	7.4.2.9	22.3.2	MAC	MAC
0x23	Simplified Superframe Specification IE	X					7.4.2.10	6.2.3, [B3]	MAC	MAC
0x24	Simplified GTS Specification IE	X					7.4.2.11	6.2.3, [B3]	MAC	MAC
0x25	LECIM Capabilities IE	X		X	X	X	7.4.2.12	10.1.3.11	UL	UL
0x26	TRLE Descriptor IE	X	X	X	X	X	E.5.1.1	E.4.2, E.4.3	MAC	MAC
0x27	RCC Capabilities IE	X		X	X		7.4.2.13	6.2.9, [B3]	UL	UL
0x28	RCCN Descriptor IE	X					7.4.2.14	6.2.9, [B3]	UL, MAC	UL
0x29	Global Time IE	X					7.4.2.15		UL	UL
0x2a	Assigned to external organization [B1]									
0x2b	DA IE	X					7.4.2.16	6.7.9	UL	UL
0x2c–0x7d	Reserved									
0x7e	Header Termination 1 IE	X	X	X	X	X	7.4.2.18	7.4.1	MAC	MAC
0x7f	Header Termination 2 IE	X	X	X	X	X	7.4.2.19	7.4.1	MAC	MAC
0x80–0xff	Reserved									

Figure A.1: Element IDs for Header IEs [“IEEE Standard for Low-Rate Wireless Networks” 2020]