Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione

Curriculum: Telecomunicazioni

Security in IoT systems - Issues and Solutions

Candidate
Francesca Nizzi

Supervisors
Prof. Romano Fantacci

Dr. Tommaso Pecorella

Dr. Francesco Chiti

Dr. Laura Pierucci

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXII, 2016-2019

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

People talking without speaking
People hearing without listening
People writing songs that voices never share
No one dared
Disturb the sound of silence

The Sound Of Silence – Simon & Garfunkel

Acknowledgments

Dunque eccomi quì, alla fine di questi 3 anni, per certi versi lunghissimi, per altri volati in un minuto.

Se sono arrivata fino a questo punto lo devo principalmente al mio tutor Tommaso che (a modo suo, ma gli si vuole bene lo stesso) mi ha guidato (e sopportato) in questo viaggio. Non smetterò mai di ringraziarti per tutto quello che hai fatto per me.

Ovviamente non possono mancare i ringraziamenti al Prof. Romano Fantacci, e ai Dott.ri Francesco Chiti e Laura Pierucci per il loro aiuto in questa avventura.

Un ringraziamento particolare va' a Benedetta, compagna di avventura e persona a cui è impossibile non voler bene; ad Alessio e Giulio e più in generale a tutti i ragazzi del DaCoNetS; alla mia famiglia, a David e ai miei carissimi amici.

A tutte le awesome persone di Estra, OF e ZTE conosciute in questo ultimo anno e mezzo di sperimentazione 5G, è soprattutto grazie a Voi se ho capito che cosa voglio fare da "grande" !

So I'm willing to change,
I'm going to try,
To show I am strong enough to trust in you
Trust In You – The Offspring

Contents

Contents v

1 Introduction 1

2 Network Models 5

2.1 The Internet of Things (IoT) Panorama . . . 5

2.1.1 A Brief Story about IoT . . . 5

2.1.2 Scenarios . . . 7

2.2 Multi-hop Technologies . . . 8

2.2.1 IEEE 802.15.4 . . . 8

2.2.2 Internet Protocol version 6 (IPv6) . . . 10

2.2.3 IPv6 over Low power Wireless Personal Area Network (6LoWPAN) . . . 12

2.2.4 Routing Protocol for Low-Power and Lossy Networks (RPL) . . . 13

2.2.5 Interactions between RPL and 6LoWPAN-ND protocols: incorrect loop detection and unnecessary recovery 15

2.3 Cellular Networks . . . 21

2.3.1 Previous Cellular Networks Technologies . . . 21

2.3.2 5G . . . 23

3 National 5G Trial 31

3.1 Project Objective . . . 31

3.1.1 Area 2 - Prato and L'Aquila trial . . . 32

3.2 The first Demo . . . 34

3.2.1 Result . . . 37

3.3 The Second Demo . . . 39

3.4 Role of the networking simulators . . . 40

3.4.1 Traffic Models 41
3.4.2 Transport Protocol Models 41
3.4.3 Network Slicing Models 41
3.4.4 Core Network Performance Models 42
3.4.5 Scalability and Extended Time Simulations 42

4 Network Protocols for IoT 45
4.1 A Distributed Token Passing Protocol for Time Constrained Data Gathering in Vehicular Ad Hoc Networks 45
4.1.1 Related work 48
4.1.2 System Model 52
4.1.3 Proposed Protocol 55
4.1.4 Numerical Results 61
4.2 A cooperative spectrum sensing protocol for IEEE 802.15.4m wide-area WSNs 64
4.2.1 Introduction 66
4.2.2 System Model 67
4.2.3 Proposed Method 69
4.2.4 Numerical Results 73

5 System Security 79
5.1 Security in Classical and IoT Networks 80
5.1.1 Defense in Depth and Whack-A-Mole Approaches . . 82
5.1.2 Privacy Concerns 84
5.2 "Network Sentiment" Framework to Improve Security and Privacy for Smart Home 85
5.2.1 Introduction 85
5.2.2 Related works 88
5.2.3 SHIELD system 90
5.2.4 Network Sentiment 94
5.2.5 Smart Home Testbed 98
5.3 IoT Security via Address Shuffling: the Easy Way 103
5.3.1 Threat Model and Motivating Scenarios 104
5.3.2 Related Work 106
5.3.3 Proposed Method 107
5.3.4 Simulation Results 118
5.4 False Data Detection for Fog and Internet of Things Networks 122
5.4.1 Introduction 122

5.4.2 Related Works 125
5.4.3 Attack Tree and Attack Models 127
5.4.4 Data IDS 133
5.4.5 Experimental results 140

6 Conclusion and Future Works 149

A GAUChO Project 153
A.1 The GAUChO Project: Overview and Challenges 153
A.2 Contribution 155
A.3 GAUChO Platform 156

B Publications 159

Bibliography 163

Chapter 1

Introduction

Omne initium difficile est,
sed forti et patienti animo obtines omnia
Any beginning is difficult,
but you will get everything with a strong and patient spirit –
Anonymous

With the term Internet of Things (IoT) we define an eco-system formed by interconnected "smart" objects (things) that: are *connected*, able to *interact*, and *exchange and elaborate data*. To communicate its data and to have access to aggregate information of neighbors make a thing recognizable and it is the basis to fuel it with 'intelligence', as is the ability to autonomously react to changes in the environment. As an example, an alarm clock could ring earlier than expected according to the expected travel time needed to reach the destination, an heating system could adjust its settings if the outside temperature drastically decreases, etc. In order to perform its operations, IoT devices are able to map the real word in the "cyber" one and vice-versa. It easy to understand that the possible applications of IoT systems are endless: manufacturing processes, autonomous driving, health, environmental protection, etc.

It is extremely difficult to design a protocol that is able to embrace every single IoT use-case due to the diverse scenario requirements and the heterogeneous devices types. Recently, new protocols able to simplify the communications have been standardized. As an example, Institute of Electrical and

Electronic Engineers (IEEE) 802.15.4 defines the physical and Medium Access Control (MAC) layers for Wireless Sensor Network (WSN). Moreover, in order to fully interconnect the IoT devices through Internet, a number of adaptation layers have been defined by the Internet Engineering Task Force (IETF) community, such as 6LoWPAN and RPL. Application-level protocols have been standardized as well, even though they suffer from the usual trend between the need for standardization and the tendency of manufacturers to create a 'closed' ecosystem to lock the customers to a specific vendor. Nevertheless, the market seems to be oriented toward open protocols, like Constrained Application Protocol (CoAP) (developed by IETF), Message Queue Telemetry Transport (MQTT) (ISO/IEC PRF 20922), and Lightweight M2M (LwM2M) (developed by the Open Mobile Alliance for M2M).

Obviously proprietary solutions still exist, and paying a royalty offers the advantage of "certified" devices that (hopefully) should inter-operate in a seamless way. The principal industrial solutions are ZigBee, LoRa/LoRaWAN, and Sigfox.

One of the most serious issues in IoT are in the privacy and security areas. Firstly, the devices to communicate use wireless technologies and, for definition, are easy to eavesdrop, secondly the devices interact directly with the real world, raising concerns in the privacy and safety of users. These are aspects must not be underestimated because a small "incident" can lead to safety risks. In general, IoT security issues are quite similar to the "traditional" devices ones, e.g., use un-encrypted messages, weak authentication methods, guessable password and user names, default login information left unchanged upon deployment, etc. In addition to these problems, IoT devices have computational and memory limitations, they are often battery-powered, and placed in inaccessible areas. These problems make it hard, if not impossible, to use 'traditional' security policies and attack countermeasures. As a consequence, novel security approaches must be developed.

Following these considerations, during My PhD I focused my research on two main topics: develop communication protocols aimed to fulfill the scenario constrains of a typical IoT applications, and enhance the security for IoT systems.
In addition, in the last year I contributed to the Italian 5G trial. One of the selected area are the cities of Prato and L'Aquila and in this area the trial is led by OpenFiber and Wind3, in addition ZTE is providing radio access, core

and transport networks. Moreover, the Use Case where I'm contributing the most is supervising by Estra SpA.

For what concerns the first topic, I studied two cases, as is multi-hop networks, and cellular systems. In particular, in the first sub-topics I develop a novel approach for time constrained information gathering in Vehicular Ad-Hoc Network (VANET) scenario, based on a token passing scheme, adapted to wireless communications by creating a virtual ring where nodes are connected to a predecessor and a successor. In the second sub-topic a novel channel allocation scheme for Low-Rate Wireless Personal Area Network (LR-WPAN) based on a cooperative centralized spectrum sensing approach is proposed.

The second topic is focused on IoT security and privacy. A first result obtained is the development of an easy to implement algorithm able to change every kind of address (MAC and IP) with a minimal impact on the network overhead. This result aims at decreasing the attack surface, according to the principles of Moving Target Defense security.

The key novelty behind this shuffling algorithm is the hash function that enables a controlled and collision-free address shuffling. Only the legitimate nodes and the network coordinator are able to predict the address renew outcomes, and from the point of view of an attacker, the addresses follow a random pattern over time. The second relevant work is focused on security in a Smart Home environment. The proposed idea is to dynamically adapt the security level according to a novel analysis. In this scenario, Internet Service Provider and end-user can cooperate with each other to adapt the security level. Moreover an user security level is propagated to nearby home networks to increase/decrease their relative security. The last work is based on a novel Intrusion Detection System based in real-time data analysis. This new approach is based on the real-world data collected by the sensors: data measured by the IoT devices are checked against the "neighbor" measurements, and if an anomaly is detected, countermeasures are applied. This approach is useful in preventing attacks towards the sensed data.

This dissertation is organized as follows: Chapter 2 deals with the scenarios and network models studied during my Ph.D.; in Chapter 3 the national 5G experimentation which is held in Prato is presented and where I contributed to; in Chapter 4 the two developed protocols are illustrated; in Chapter 5 the system and the security works are explained. Finally, in Chapter 6 the conclusions and future research trends are described. Moreover, in Appendix A the GAUChO - A Green Adaptive Fog Computing and Networking Architecture project is explained; Appendix B contains the list of my publications.

Let the rock off begin !
BEELZEBOSS (THE FINAL SHOWDOWN) – TENACIOUS D

Chapter 2

Network Models

> *The nice thing about standards is that there are so many to choose from. And if you really don't like all the standards you just have to wait another year until the one arises you are looking for.* – Andrew S. Tanenbaum

Starting with a short introduction about IoT, in this chapter will be presented the enabling technologies for multi-hop communications and the cellular technology.

2.1 The IoT Panorama

2.1.1 A Brief Story about IoT

Since the '80s, researchers around the world have been looking for ways to connect everyday objects (such as toasters, refrigerators, coffee machines) to Internet and control them remotely. The 1999 can be considered as the "zero-year". Internet was the hot topic and in his presentation at Procter & Gamble, Kevin Ashton coined the term *Internet Of Things* to attract investments. His idea was summarized in a phrase: *In the real world, things matter more than ideas*[1].

[1] Later, in 2009, Kevin Ashton wrote a paper [49] where he described this idea more formally.

In June 2000 LG announced its first “IoT” device: LG Internet Digital DIOS, an Internet refrigerator [24]. Despite this, due to its high cost (more than $ 20,000) it was a commercial flop.

Later, in December 2002 the NY Times Magazine published an article [28] about *Ambient Orb*: a device created by MIT Media Lab spin-off. Ambient Orb monitored the Dow Jones, personal portfolios, weather and other data sources and changes its color based on the dynamic parameters. It was considered one of the first IoT devices.

In November 2005 the International Telecommunications Union (ITU) published a report stating: *A new dimension has been added to the world of information and communication technologies (ICTs): from anytime, any place connectivity for anyone, we will now have connectivity for anything. Connections will multiply and create an entirely new dynamic network of networks – an Internet of Things* [20]. It was the first technical report about IoT world.

Following this trend, in 2008 a group of companies launched the Internet Protocol for Smart Objects Alliance (IPSO Alliance)[2] to promote and enable the “smart object” communications. Its primary task was offered webinars, interoperability events, the publication of white papers, and work alongside standards organizations such as IETF, IEEE, and European Telecommunications Standards Institute (ETSI). Thanks to these missions, TIME Magazine listed the IPSO Alliance (and as consequence IoT) as the 30th most important innovation of 2008 [8]. The best invention in 2008 was the *Retail DNA Test*.

In 2009 Google started preliminary studies and test on self-driving car, and St. Jude Medical Center releases the Internet-connected pacemakers[3].

In 2010 the number of connected devices per person was 1.84: 12.5 billion of devices connected to Internet against 6.8 billion of world population. Besides this, China named IoT as key technology and announced big investment. In this context (more connected devices than world population) first predictions about connected devices in the next 10-15 years were supplied. The well-known prediction given by Hans Vestberg (Ericsson former CEO) [11] stated that in 2020 the number of connected devices will be 50 billion: approximately 6.58 connected devices per person [21]!

In 2011 the American firm Gartner added IoT technology to its “hype

[2] On March 2018 the IPSO Alliance merged with the Open Mobile Alliance (OMA) to form OMA SpecWorks

[3] This pacemaker was also the first IoT medical device to be hacked in 2016.

cycle[4]". IoT was introduced in the graph as a *technology trigger* and the experts hypothesized that the scientific and industry communities will research in this field for at least 5 years. Due to an explosion of "smart" devices, in 2016 IoT was dropped from the hype cycle, meaning that IoT became a "mainstream" technology and not only a research field.

Nowadays IoT developments gets devices cheaper, and easier-to-use. Self-driving cars continue to improve, blockchains start to be used in public administration (for example during elections), and the perspective to control home-devices from a smartphone app continue to make IoT an attractive offer for future. However, it must be also noted that the connected devices predictions have been revisited: important experts assert that the connected devices in the 2020 will be approximately 28-30 billion [25].

2.1.2 Scenarios

During My PhD period, I focused my research on three particular IoT scenarios.

Vehicular IoT. This scenario is mainly related to *connected cars*, that is cars able to communicate with the surrounding environment. Car communications can be classified into five types: Vehicle to Infrastructure (V2I), Vehicle to Vehicle (V2V), Vehicle to Cloud (V2C), and Vehicle to Pedestrian (V2P), Vehicle to Everything (V2X) [12]. Moreover, the applications could be divided into two categories: Single vehicle, and Cooperative safety and efficiency applications.

Smart Home. A Smart Home is a building with Internet-connected devices used in order to enable its remote monitoring and management. A Smart Home is characterized by the presence of a huge number of small, low power devices, along with more classical ones. According to the IoT paradigm, all of them are expected to be always connected to the Internet in order to provide enhanced services. Examples of devices can be Smart TVs, lightbulbs, thermostats, Kitchen appliances, Pet feeders, etc. We can consider the Smart Home as the smallest entity of the so-called Smart City environment. Even if the information produced by the single device is very small (i.e., few bytes per hour, typically), the massive number of devices leads to a large information rate. The biggest advantage of the Smart Home concept is to allow a remote control over the house and its residents, enabling smart

[4] Is a graphical and conceptual presentation to represent the maturity, adoption, and social application of emerging technologies.

services, grater energy efficiency, etc. However, it is clear that the data collected by the Smart Home can lead to serious privacy problems, making it necessary to ensure a proper security level in the system.
Industrial IoT. We can define the Industrial IoT (IIoT) as the use of interconnected intelligent devices to enhance manufacturing and industrial processes. Each network (i.e., a single IIoT ecosystem) is able to monitor, collect, exchange, and analyze data. IIoT presents a defined architecture divided in four layers: the lowest layer is the *device* layer that represents the physical components such as Cyber-physical systems (CPS), above there is the *network* layer that represents the communication protocols that collect and transport data to the *service* layer that consist in applications able to analyze data. The upper layer is the *content* layer that represent the user interface. Nowadays IIoT is also called Industry 4.0 because we are spectators of the fourth industrial revolution.

2.2 Multi-hop Technologies

As explained previously, IoT concept is growing quickly, and in parallel the major international standardization bodies are developing IoT standards. In this section the IoT main reference standards are presented.

A Low power and Lossy Network (LLN) is a type of network characterized by low energy consumption, low transmission bit-rate, highly unreliable communication link (due to its wireless nature), and all the interconnected devices are constrained (in terms of power supply (i.e. battery), computing power, memory, etc..). A LR-WPAN (or LoWPAN) is a subset of LLN. Frequently, the LR-WPAN network is called WSN. The reference protocol suite is shown in Fig. 2.1.

The biggest ideas behind these technologies are that *the IP protocol could and should be applied even to the smallest devices* [135] and that *low-power devices with limited processing capabilities should be able to participate in the Internet of Things* [162].

2.2.1 IEEE 802.15.4

IEEE 802.15.4 specify physical and MAC layers for LR-WPAN. It is the first (non-proprietary) low-power radio standard and its first version was released in May 2003. As written in [34]: *the main objectives of an LR-WPAN are*

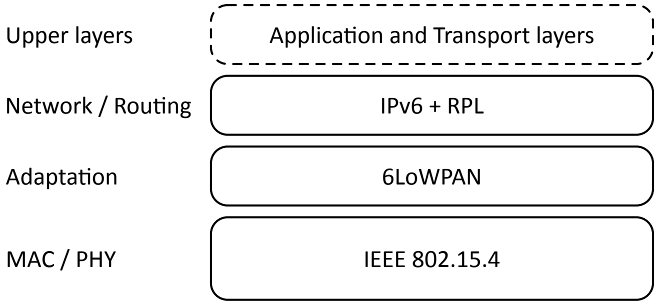


Figure 2.1: LR-WPAN reference layers.

ease of installation, reliable data transfer, extremely low cost, and a reasonable battery life, while maintaining a simple and flexible protocol. To reduce the device cost, IEEE 802.15.4 devices can be Full-Function Device (FFD) or Reduced-Function Device (RFD). A FFD is able to forward packets and perform coordination functions while a RFD has very limited capabilities and can only join to the network as a leaf.

The standard specifies the allowed network topologies. In its first version only **star** and **peer-to-peer** topologies were mentioned but soon also the **cluster-tree** was added. In all of them, the Personal Area Network (PAN) coordinator is responsible for starting and maintaining the network, manage the node associations, and forward the PAN traffic to Internet.

The operating bands are almost all in the Industrial, Scientific and Medical (ISM) band (a spectrum portion that is license-free) and the bit-rate depends on the adopted physical layer[5].

Furthermore, the standard states that a device can use two different address types: **long** (64 bits) and **short** (16 bits, henceforth called MAC-16). The long address is globally unique, and it is assigned to the device by the manufacturer[6]. MAC-16 is unique in the PAN, and it is assigned to devices by the PAN Coordinator during the association phase. A device can use either its long or short address within the PAN, but long addresses are discouraged due to their excessive length.

[5] In the latest release there are 18 different physical layers.

[6] In the standard is called extended address and corresponds to the Extended Unique Identifier - 64 bits (EUI-64) address device.

Theoretically, the available MAC-16 address space is 2^{16}. However, the following addresses are reserved [34, 131]:

- `FF:FE` - Device with no short address allocated, it will use its long address in all messages;
- `FF:FF` - Device not yet associated to PAN / Broadcast MAC-16 address;
- `80:00` - `9F:FF` - Multicast MAC-16 addresses.

As consequence, only $2^{16} - (2 + 2^{13}) = 57342$ addresses are available.

Moreover, the MAC-16 address assignment represent a privacy problem that will be discussed in Sec. 5.3.

An important feature of IEEE 802.15.4 is the MAC frame size: the maximum Protocol Data Unit (PDU) is 127 Bytes.

2.2.2 IPv6

IPv6 is the "new" Internet Protocol (IP) version. It was announced in 1995 with the Request For Comment (RFC) 1883 [79] but was launched on 8 June, 2011. The IPv6 objective was to fix the previous IP version (the Internet Protocol version 4 (IPv4)) weaknesses and enhance its strengths. IPv6 has introduced new features that make it incompatible with the predecessor. These features are:

- Address space (i.e. number of bit);
- Simplified Header;
- Interface Auto-configuration;

One of the biggest improvement is the *huge* addressing space. As Steve Leibson wrote in [22]: *we could assign an IPv6 address to every atom on the surface of the earth, and still have enough addresses left to do another 100+ earths.* This could be a pretty accurate estimate because now there are available $2^{128} \approx 3.4 \times 10^{38}$ addressed[7]! An IPv6 address is expressed as 8 groups of 4 hexadecimal digits, each group representing two octets, where the groups are separated by colons (:), e.g., `2001:db8:f00d:cafe::1`.

[7] IPv4 allowed $2^{32} \approx 4.3 \times 10^{9}$ addressed. On 3 February, 2011, Internet Assigned Numbers Authority (IANA) held a ceremony when it allocated the lasts unreserved address blocks thus exhausting all possible addresses [16]

Regarding the packet header, IPv6 uses an header 40 Bytes long divided in 8 fields, where the IPv4 header is 20 Bytes long divided in 13 fields. This means that routers can process packets with fewer checks, increasing their speed.

In IPv6, Address Resolution Protocol (ARP) has been redesigned in order to enable a seamless address auto configuration based on the MAC address. Temporary addresses (needed for privacy) and the non-uniqueness of MAC addresses requires that every host, for every address, must test the address uniqueness through a Duplicate Address Detection (DAD) procedure. This involves the use of Neighbor Solicitation (NS) - Neighbor Discovery (ND) (Neighbor Discovery Protocol (NDP)) for every address. Moreover, since addresses are usually short-lived the procedure is repeated frequently.

IPv6 establishes these *types* of addresses:

- *unicast*: one-to-one. Each destination address uniquely identifies a single receiver endpoint;

- *multicast*: one-to-many. Packets are routed simultaneously in a single transmission to many recipients. Multicast addresses have the first octet equal to `FF`;

- *anycast*: one-to-nearest. Packets are routed to any single member of a group of potential receivers that are all identified by the same destination address. The routing algorithm selects the single receiver from the group based on which is the nearest according to some distance measure;

Furthermore, IPv6 has a Maximum Transmission Unit of at least 1280 Bytes: this means that *all* the IPv6 packets are bigger than 1280 Bytes. It is worth noticing that all the Internet links have an Maximum Transmission Unit (MTU) size (path MTU), and this could be smaller that the minimum MTU required by IPv6 (it is the case of an IEEE 802.15.4 link). IPv6 *disallows* the use of fragmentation performed by intermediate nodes, only the originator sender is allowed to fragment a packet (if a traveled path MTU is too small to contain an IPv6 packet). A solution to adopt IPv6 over WSN protocols is represented by the 6LoWPAN set of protocols.

2.2.3 6LoWPAN

After the first release of IEEE 802.15.4 in 2003 and due to its reached popularity a question arised: *how can we connect these networks to the already existing Internet ?* Connect LoWPANs to the existing infrastructure have many benefits: as explained in [114] *IP-based technologies already exist, are well-known, and proven to be working.* In order to connect the LLN networks to the existing IP network some problems must be considered. In particular:

- *Small packet size*: IEEE 802.15.4 has 102 Bytes of maximum payload size, without considering the security header, which would reduce the payload size even further. Due to this, IPv6 and upper layers headers must be compressed. The short packet size poses problems also for the routing protocol, that must achieve low data packets overhead.
- *Device types*: due to the main characteristic of these devices (mainly computational and memory constraints), the routing protocol must be extremely efficient and consume very low bandwidth for route discovery and maintenance.
- *Configuration and Management*: it can happen that some devices are deployed in impervious areas. As a consequence, minimal configuration and management is a requirement.
- *Service Discovery*: LLNs require easy to implement protocols to discover, control and manage devices.

The 6LoWPAN group of specifications [100, 131, 160] were designed to solve the problems cited above. They can be seen as a middleware between IP layer and IEEE 802.15.4, or in some cases a *substitute* protocol. In particular they provide:

- Efficient IPv6 header compression [100];
- Fragmentation and reassembly [131];
- Neighbor discovery optimization [160]. This function is based on the IPv6-ND [138] and is optimized for the LLNs. Its goal is to minimize the number of Internet Control Message Protocol (ICMP) packets needed in a Low-power Wireless Personal Area Networks (LoWPAN), primarily the ones dealing with ND. This protocol defines a method

to distribute Contexts used by [100] and redesigns the NDP in order to leverage the LLNs short addressing assignment.

IPv6 over Low power Wireless Personal Area Network - Neighbor Discovery (6LoWPAN-ND) defines a new entity: the 6LoWPAN Border Router (6LBR). It is the “bridge” between 6LoWPAN and IP-based networks. Moreover, the address collision detection task is assigned to the 6LBR. All nodes register to the 6LBR, and all ND and DAD checks (usually via multicast messages) are performed through unicast queries to the 6LBR. Even though not explicitly required by the standard, the 6LBR and the PAN coordinator functions are usually implemented within the same node.

Despite the 6LoWPAN-ND benefits, it is not a mandatory protocol and it may not be implemented.

It is worth noticing that the fragmentation and compression is completely transparent to IPv6 and also the User Datagram Protocol (UDP)[8] header can be compressed.

2.2.4 RPL

In a WSN, and in LLN in general, there are four routing approaches:

- No routing: used when there is no multi-hop. A simple table is enough;
- Proactive routing: Low mobility. It is possible to pre-calculate the routes;
- Reactive routing: High mobility. Routes must be calculated on-demand;
- Flooding: No time to calculate a route. KISS approach;

Moreover, routing in LLN might be performed in two different layers:

- Mesh-under: the routing is performed at MAC level: the MAC layer is responsible for the whole routing using L2 techniques;
- Route-over: the routing is done at IP layer. It’s almost the normal IPv6 routing.

Routing becomes a **key function**.
RPL [183] is a new routing protocol designed by IETF ROLL Working

[8] Most of the IoT applications work with UDP as Transport protocol.

Group[9]. RPL is a proactive routing solution built to meet LLN requirements. It is worth noticing that RPL is not limited to WSN environment, but it is becoming the standard routing protocol for WSN operating systems.

In general, RPL is based on the construction and maintenance of a *Destination-Oriented Directed Acyclic Graph (DODAG)*: a Directed Acyclic Graph (DAG) originating from a root node. DODAG construction is based on *Objective Function (OF)* and *Goal*. A network might have multiple DODAGs each with a different Goal. Its construction is perform from the root to the nodes. A node can join more than one DODAGs at the same time and can either participate to the DODAG as a Router (FFD) or as a Leaf (RFD or FFD).

RPL defines three values to build and maintain a topology:

- *RPL Instance ID*: identifies a set of one or more DODAGs. At most, a node can be part of one DODAG in a RPL Instance but can belong to multiple RPL Instances;

- *DODAG ID*: is the identifier of a DODAG root and corresponds to a routable IPv6 address. The set of (RPL Instance ID, DODAG ID) identifies a single DODAG in the network;

- *DODAG Version Number*: is a value that, when changed by the root, triggers a complete DODAG refresh.

Thus, a DODAG is uniquely identified by the tuple {*RPL Instance ID, DODAG ID, DODAG Version Number*}.

RPL defines two types of routes: *Upward* (from the nodes to the root), and *Downward* (from the root to the nodes), as shown in Fig. 2.2. DODAG construction, maintenance, and Upward routes management are performed through DODAG Information Object (DIO) messages. This type of message is periodically sent by the DODAG root and by all the FFD nodes in the network. DIO message is *usually multicast*, i.e. from a node to all its children. To limit the network overhead, unsolicited DIO are sent according to a Trickle Timer [119]: a particular timer that increases and decreases the DIO frequency according to specific rules. As a rule of thumb, one can think that the DIO transmission frequency is low when the network is "stable", i.e., when there are no changes in the topology. Although DIO messages are

[9] ROLL is an acronym for Routing Over Low power and Lossy networks and, as explained in the name, the Working Group goal is to design routing protocol for LLNs.

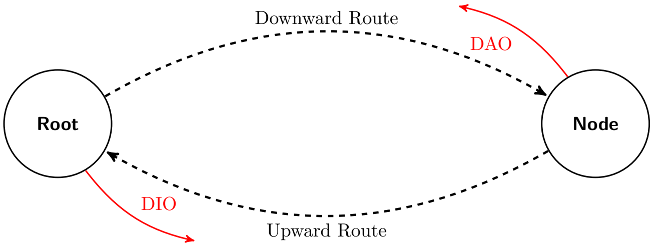


Figure 2.2: Upward and Downward routes construction process. DIO messages are multicast while DAO messages are unicast.

sent using multicast, internal RPL mechanisms ensure that all nodes in the network will receive and process them correctly.
Downward routes are propagated through DODAG Advertisement Object (DAO) messages, sent by each node to the root, via any intermediate node. DAO messages are *unicast*, i.e. from a node to its preferred parent.
Upward route is mandatory while downward route is optional, therefore DIO messages are always sent.
RPL standard provides for a third message: DODAG Information Solicitation (DIS). It is used to ask for DODAG information, and may be multicast or unicast.

RPL Control Messages (i.e. DIO, DAO, and DIS messages) are based on Internet Control Message Protocol version 6 (ICMPv6) [75] packets with *type* field sets to 155.

2.2.5 Interactions between RPL and 6LoWPAN-ND protocols: incorrect loop detection and unnecessary recovery

Theoretically, RPL and 6LoWPAN-ND protocols should be independent. Indeed, in RPL standard it is not stated that 6LoWPAN-ND is necessary for RPL operations, and is not mentioned that any ND operation may negatively affect RPL function. However, without a proper 6LoWPAN-ND implementation, RPL exhibits transients and, ultimately, can lead to a severe energy waste in the network.

Erroneous loop detection problem

In order to ensure that a DODAG is loop-free, RPL implementations must adopt various error recovery mechanisms, i.e.:

1. Data packets are marked as "upward" or "downward". A packet flowing in the "wrong" direction triggers a loop recovery;
2. RPL messages must be checked for inconsistencies, e.g., a DAO sent from a parent to a child is considered as a loop evidence.

When a DODAG loop is detected, a node should start a "local repair" procedure. This procedure involves detaching the node from the DODAG, along with all its children. All the nodes detached from the DODAG must start a new attach procedure, which means: increase DIO sending frequency, further DAO exchanges, etc.

It is important to notice that the overall effect is not limited to the nodes involved in the repair. DIS messages (used to join a DODAG) can reset Trickle Timers of neighbor nodes, increasing the DIO message frequency, and causing further energy waste. Generally, a local repair can propagate its effects to a large part of the network. As a consequence, unnecessary local repairs must be avoided at all costs.

It is worth noticing that a loop recovery should not be triggered unless necessary. However, there is a case where a condition can lead to a loop detection, even when no real loop exists. Figure 2.3 shows the scenario of this erroneous loop detection. In this situation, Node 4 is the parent of Node 5. At a certain point, Node 5 will receive a DIO message from Node 1, triggering the problem. Node 5 will receive the message either because the link between the nodes is unreliable,or because Node 5 is moving toward Node 1.

The problem arises because DIO and DAO have different destination addresses. DIO messages are sent to a link-local multicast address (`ff02::1a`), and does not required a link-layer (L2) address resolution. As a consequence, DIO messages are sent immediately. On the contrary, DAO messages are unicast. Therefore, it is necessary to have a valid L2 address in the Neighbor Discovery Cache (NDIS Cache) in order to send the packets. If the NDIS Cache Entry has expired, the packet is held in the NDIS Cache Queue until the L2 entry is refreshed by the ND protocol. This involves the transmission of two ICMP messages introducing a non-negligible delay.

The event scheduling (also shown in Figure 2.4) is as follows:

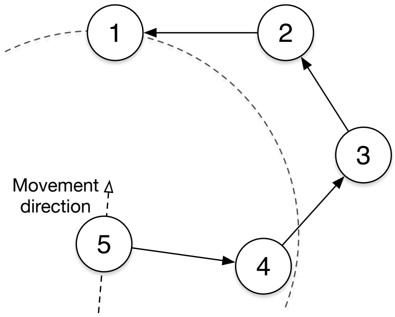


Figure 2.3: Erroneous loop event topology

T_0 Node 4 sends a multicast DIO, hence triggering a DAO from Node 5;

T_1 Node 5 replies to the DIO with a DAO;

T_2 Node 1 sends a multicast DIO;

T_3 Node 5 receives the Node 1 DIO and performs a route optimization:

- Sends a DAO to the new parent (Node 1),
- Sends a NO-PATH DAO to the old parent (Node 4), and
- Resets its Trickle Timer and starts broadcasting its new rank;

T_5 Node 4 receives the new DIO from Node 5 and records its rank;

T_6 The DAO created at T_1 is finally sent by Node 5 to Node 4;

T_7 **Node 4 receives a DAO from a node with a lower rank, and detects a loop.**

Note that the NO-PATH DAOs generated at T_3 are queued and will be sent *after* the one generated at T_1.

Summarizing, the loop is detected because a node (Node 4 in this case) receives messages out of their generation order. Even if the DIO has been generated *after* the DAO, it is transmitted *before* it, because the DAO transmission is delayed by:

$$\Delta_{\text{DAO}} = \Delta_{\text{DelayDAO}} + \Delta_{\text{NDIS cache}} \tag{2.1}$$

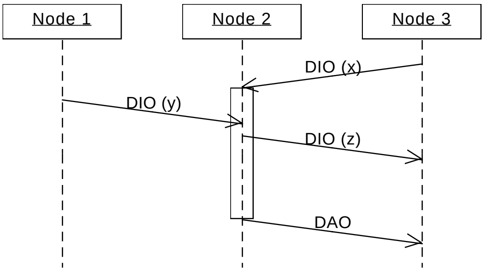


Figure 2.4: Erroneous loop event

where, as stated previously, Δ_{DAO} is a system parameter, and $\Delta_{\mathrm{NDIS\ cache}}$ is a delay due to the (eventual) NDIS cache update.

It is worth noticing, that this effect is quite uncommon as it happens only if the nodes are at a certain distance (or one is moving towards the other), and a particular NDIS Cache entry is expired.

Simulation Results

In order to evaluate the effect of the erroneous loop detection, an RPL simulator was used [51]. The simulator is based on ns-3 and implements RPL, 6LoWPAN, IEEE 802.15.4, and has a partial support for 6LoWPAN-ND.

The simulation setup is composed of 200 nodes in a 50 × 250 $[m^2]$ area, with random positions. Only one RPL root is present, and is placed at the center.

During the simulation, an erroneous loop detection was triggered at about second 600. As Figure 2.5 shown, the network has a sudden load increase, which is recovered quickly thanks (also) to the node spatial density, allowing multiple paths.

Proposed Solution

The described problem can be avoided with some simple methods (non described in the standard):

1. Do not send DAO message to a node that is no more among the parent list;

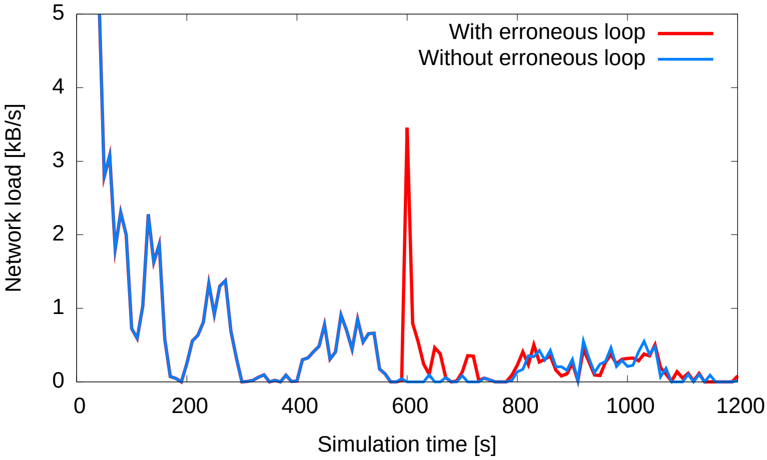


Figure 2.5: Erroneous loop network load

2. Invalidate a DAO message from an old child node;
3. Avoid DAO messages delay.

The first solution involves removing pending DAO messages from the DAO transmission queue and the NDIS Cache. When a parent is removed from the parent list, the pending DAO messages should be removed. The complexity of this solution depends on how the two message queues are organized, and it might not be trivial. In particular, the NDISC Cache is managed by the IP protocol, and the operating system might not provide a way to remove queued messages.

Invalidating incoming DAO messages may be a good alternative. However, RPL does not have a way to pair a DAO with its triggering DIO. As a consequence, a temporary state should be associated with each neighbor, in order to keep track of children nodes' eventual rank changes. However, this raises also a problem of the temporary status lifetime, as $\Delta_{\text{NDIS cache}}$ depends on the node's position in the DODAG and the network load. The complexity of this solution discourages its implementation.

The last option can be achieved by using 6LoWPAN-ND *and* setting the DelayDAO (i.e., Δ_{DAO}) to zero. When 6LoWPAN-ND is used, ND is not performed for link-local addresses as a node must assume that the link-local address has been generated from the MAC address by using IPv6 Stateless

Address Autoconfiguration (SLAAC) [175]. Therefore, the MAC address can be inferred directly from the IP address (RFC 6775, section 5.6). This mechanism is possible because MAC addresses in IEEE 802.15.4 are assigned by the PAN coordinator. As a consequence, they are unique. Moreover, 6LoWPAN-ND ensures IP address uniqueness through address registration. Since DAOs are sent to link-local addresses, by using 6LoWPAN-ND, there is no caching in the NDIS Cache.

Setting DelayDAO to zero has the negative effect to be unable to optimize the DAO transmission, i.e., each DAO will be sent and forwarded autonomously, while using a DelayDAO different DAOs can be joined in a single message. Thus, also removing the DAO delay is not a good option.

However, if 6LoWPAN-ND is used, $\Delta_{\text{NDIS cache}}$ is always zero, meaning that ignoring DAOs becomes a viable (albeit complex) option. In particular upon receiving a DIO, we must apply Algorithm 1. Upon receiving a DAO, it is just necessary to check if the sender is in the DaoBlackList and its entry is not expired. If the entry is found, then the DAO must be discarded.

Algorithm 1 Process incoming DIO

```
Define: $\text{Rank}_{\text{sender}}$: the Rank of sender node
Define: $\text{Rank}_{\text{receiver}}$: the Rank of receiving node
  if ($\text{Rank}_{\text{sender}} < \text{Rank}_{\text{own}}$) AND (sender is a child) then
      Insert {sender, $T_{\text{now}} + \Delta_{\text{DelayDAO}}$} into DaoBlackList
  Delete from DaoBlackList all entries with $T_{\text{expiration}} < T_{\text{now}}$
```

Algorithm 1 is easy to implement, prevents the false loop detection, and it disables the loop detection for specific nodes and small periods. As a consequence, it should not affect negatively the RPL loop detection and recovery mechanisms.

In our opinion, an implementer should resort to the first option. However, in networks where it is not possible to ensure that DAOs are not sent by nodes changing their rank, it is important to also invalidate them on reception, thus making it important to adopt 6LoWPAN-ND.

It is worth stressing, that 6LoWPAN-ND is *not* mandatory for RPL networks. This is especially important because RPL may be used in network types other than LLNs.

In this paper, we described an erroneous loop detection arising in RPL networks due to DAO transmission delays when nodes change their ranks.

We outlined some possible mitigation techniques to the problem. In our opinion these mitigation systems should be included in the standard, because in order to mitigate completely the problem, all of them should be applied. Nevertheless, one of them requires to use 6LoWPAN-ND, which is not mandatory for RPL.

Without the proposed mitigation, RPL can gracefully recover from the false loop detection. However, the loop recovery phase can severely degrade the network's performance and drain node's batteries.

The outlined solutions may be also useful when RPL is used in different scenarios, where 6LoWPAN-ND is not applicable. However, it will be necessary to define an upper bound to the NDP or (possibly) to use a NDP substitute algorithm, at least for the neighbor nodes.

2.3 Cellular Networks

Mobile communication has greatly evolved in the last decades, and to study the new cellular network generation it is important to "understand" the oldest generations.

2.3.1 Previous Cellular Networks Technologies

The **first** generation of wireless cellular technology (1G) was an analog mobile network and was introduced in the 80s. Its only objective was to make voice calls between mobile terminals. To achieve this, omnidirectional antennas were used in order to cover as much space as possible. The channel access was based on Frequency-Division Multiple Access (FDMA) and the voice channels were modulated by a Frequency Modulation (FM). Moreover there was no form of cryptography, and there was no a unique world-wide standard[10], but each state (or federations of states) had developed its national one.

In order to improve transmission quality, system capacity, signal coverage, and add cryptography, the **second** generation of 2G mobile networks was studied. It was the first wireless cellular technology using a digital modulation. 2G technologies enabled the networks to provide the services such as text messages (SMS - Short Message Service), and MMS (Multimedia

[10] Part of this generation are: Nordic Mobile Telephone (NMT), used in North Europe, Switzerland, the Netherlands, Eastern Europe and Russia; Advanced Mobile Phone System (AMPS) used in North America and Australia; Total Access Communications System (TACS) in the United Kingdom (UK); Radio Telephone Mobile (RTM) in Italy.

Messages Service). All text and phone conversations are digitally encrypted. Even the 2G system did not have a single worldwide standard, but unlike the previous generation, the European Union, through the European Telecommunications Standards Institute (ETSI), developed a single standard for all member countries: The Global System for Mobile Communications (GSM). The channel access was based on Time-Division Multiple Access (TDMA). Other worldwide standard were: cdmaOne (IS-95) used in United State of America (USA) and Asia, Personal Digital Cellular (PDC) used in Japan[11].

In order to allow web navigation (i.e., data transmissions) 2.5 and 2.75 generations were standardized. The biggest difference with the previous mobile generation was that 2G and 1G used circuit-switching, while 2.5G and the next generations used packet-switching. General Packet Radio Service (GPRS) and Enhanced Data rates for GSM Evolution (EDGE) were the GSM evolution.

The **third** generation (3G) was developed when the 2G technologies were no longer able to support the new applications that were emerging. The new international 3G mobile telephony standards follow the International Mobile Telecommunications (IMT)-2000 [23] technical specifications defined by the International Telecommunication Union - Radio communication (ITU-R) and were released in the early 2000s. For the first time, the international community is looking for a worldwide mobile communication system for global roaming. In Europe and Japan Universal Mobile Telecommunications Service (UMTS) was created and revised by the 3rd Generation Partnership Project (3GPP): it was a GSM evolution. The access channel was based on the novel Wideband Code Division Multiple Access (WCDMA). In North America, and in some Asian Countries the 3rd Generation Partnership Project 2 (3GPP2) developed its own 3G standard: the CDMA2000 a backwards-compatible evolution of the cdmaOne.

As it happened for the GSM, also for UMTS some “small” (backward-compatible) upgrades have been developed: High Speed (Downlink/Uplink) Packet Access (HSPA), know as 3.5G and in 2010 the Evolved HSPA (HSPA+, known as 3.75G).

Some (big) problems, as high battery consumption, high costs of infras-

[11] It worth noticing that IS-95 and PDC were not compatible with the GSM. To overcome this problem, “GSM-style” standards were developed in USA (PCS-1900), UK and Asia (DCS-1800). These standards were compatible with the GSM, they worked at ‘difference frequencies: so the dual, tri and quad-band (850 - 900 - 1800 - 1900) mobile phones were arise.

tructure construction, and a standard technologically not mature, brought the international community to develop the **fourth** generation (4G) cellular systems. The main differences with respect to the previous generations are the all-IP packet-switched network and the capability to send and receive data at speeds comparable with DSL connections. It is important to note that the so-called Long Term Evolution (LTE) is not a full 4G standard, because it does not meet the technical criteria (as specified in the IMT-Advanced Standard [14]) of a 4G wireless cellular technology, and it should be more properly called a 3.95G mobile standard. The LTE Advanced (LTE-A) standard formally satisfies the ITU-R requirements to be considered a 4G standard. LTE is a cellular standard developed by ETSI and it is incompatible with 2G and 3G networks.

While in Oceania and North America the 2G cellular network have already been switched off, in Europe most States will turn off the 3G network diverting the voice service to the GSM network (or using Voice Over LTE (VoLTE) where possible) and the Internet traffic to the 4G-5G networks.

2.3.2 5G

Telco operators and industry experts are already engaged in research and development of the next generation of cellular network, the infrastructure that can further improve the data transmission rate to support the growing number of users and services accessible from mobile network. The **fifth** generation (5G) mobile network is an emerging technology that will satisfy the ITU-R recommendation M.2083-0 “IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond” (hereafter referred simply as IMT-2020) [18]. This recommendation defines the basis that the next generation cellular telecommunication standard must have, describing application trends (therefore potential users, traffic types and technological trends), and spectrum implications.

ITU-R hypothesizes three usage scenarios in IMT-2020:

- **Enhanced Mobile Broadband (eMBB)**. This is the reference scenario for the human-centric use cases, i.e., access to multimedia contents, services, and data. The goal is to reach a data rate of gigabytes per second;

- **Ultra-reliable and low-latency communications (URLLC)**. This is the reference scenario for the critical machine-type communications such as industrial applications, autonomous vehicles, and remote medical surgery. In this use case, stringent requirements (in terms of throughput, latency and availability) are present.

- **Massive Machine-Type Communications (mMTC)**. This is the Internet of Things (IoT) reference scenario. In this case billions of sensors and machines will be interconnected, each transmitting a low volume of (sensitive) data. Low cost and long battery life are fundamental requirements that the next mobile networks generation products will have to meet.

In Europe the studies on the next generation telecommunication started in December 2013 when the European Union (EU) funded, in collaboration with the European Information and Communications Technology (ICT) industries and research centers, the 5G Public Private Partnership (5GPPP). In early-2015 5GPPP released its first work which gives *an overview of the 5G vision of the European ICT sector* [6]. In this contest a phase 1 (finished in June 2017) was aimed at the pre-standardization of the entire 5G system (physical layer, network architecture and management). The standardization and experimentation are now in phase 2 [5].

5GPPP studies indicate that 5G will be the native network for many industries such as automotive, healthcare, energy, media & entertainment, and factories of the future, and will lead to new applications with constraints in low latency (few milliseconds), high throughput, mobility, connection density and reliability. 5G will provide new services and better communication technologies allowing the best development of the so-called IoT: millions of devices with an electronic identity, will be able to communicate over the cellular network and be controlled remotely.

It is wort noticing that the greatest challenge of the 5G network will be to provide support for a wide range of services (with very different requirements). To achieve this, 5G architecture network must be extremely flexible: toward this end, network slicing is the most promising technology. According to 5GPPP, *the network slice is a composition of adequately configured network functions, network applications, and the underlying cloud infrastructure (physical, virtual or even emulated resources, RAN resources etc.), that are bundled together to meet the requirements of a specific use case, e.g., bandwidth, latency, processing, and resiliency, coupled with a business purpose* [35]. A slice instance requires the assignment of resources that will be

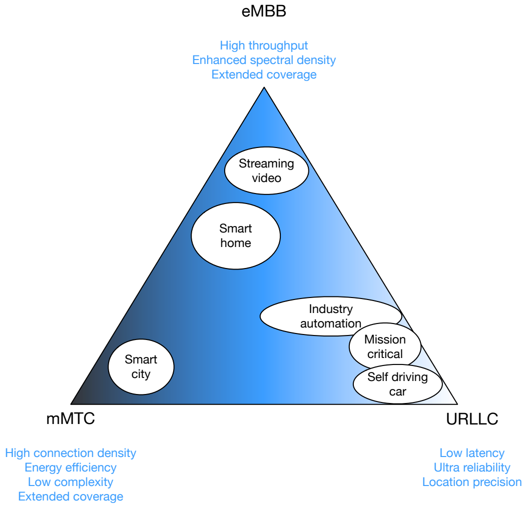


Figure 2.6: IMT-2020 scenarios.

used in isolated and disjunctive (or non-disjunctive) mode.

Physical network is sliced in many virtual subnets that are sharing infrastructures and radio resources, being logically separated. Resources, security features and Quality of Service (QoS) are assigned to any network slice.Numerous network slices can be developed with the same characteristics and be usable by different groups of users according to specific needs. Different network slices can be used for the eMBB, mMTC and URLLC service scenarios, each characterized by specific requirements and managed by the orchestration level. The solution offered is able to support both static and "on demand" slices. The introduction of slices allows the creation of dedicated networks for dedicated services, these networks are conceptually separate from each other. It will be possible to design the slices using management and design tools that use open API interfaces, and instantiating predefined templates.

Network slicing abstracts, isolates and separates the logical network components from the physical resources. To support this abstraction a management plane is required. The fundamental element enabling evolution of the management plane is Software-Defined Networking (SDN) architecture: SDN decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infras-

Table 2.1: IMT-2020 reference targets.

Key capability	IMT-2020 target	Relevant for
Peak data rate	20 Gbps	eMBB
User experienced data rate	100 Mbps	eMBB
Latency	1 ms	URLLC
Mobility	500 km/h	eMBB - URLLC
Connection density	10^6/km^2	mMTC
Energy efficiency	100×4G	eMBB
Spectrum efficiency	3×4G	eMBB
Area traffic capacity	10 Mbps/m^2	eMBB

tructure to be abstracted for applications and network services. With this new emerging technology, network operators will allocate and release network resources according to the specific context required by the application. The new 5G network will support both machine-to-machine and machine-to-human communications that have different characteristics (in terms of generated traffic): so this new network will be conceives as flexible infrastructure that is application, service, and context aware [35].

5GPPP defines six use case family groups [4]: dense urban, broadband everywhere, connected vehicles, future smart offices, low bandwidth IoT, tactile Internet/automation.

Each of these use cases has a Key Performance Indicator (KPI) that correspond to a performance requirement (in terms of consumers service experience) that the future 5G network will have to support. Eight *key capabilities* are suggested [18]: peak data rate, user experienced data rate, latency, mobility, connection density, energy efficiency, spectrum efficiency, area traffic capacity. Additionally other five capabilities (spectrum and bandwidth flexibility, reliability, resilience, security and privacy, operational lifetime) are required in order to make the future network more secure, flexible and, reliable. Peak data rate, user experience data rate, mobility, energy efficiency, spectrum efficiency and area traffic capacity are the parameters that must be considered in eMBB scenario, latency and mobility are the distinctive properties of the URLLC scenario, and connection density is the characteristic feature of the mMTC scenario. The IMT-2020 reference targets are summarized in table 2.1.

Contrary to IMT-2020, 5GPPP classified the key capabilities in *nine* parameters: device density, mobility, infrastructure topology, traffic type, user data rate, latency, reliability, availability, 5G service type communications. Moreover, localization and security requirements are taken into account in the development of industrial applications. The 5G Italian experimentation uses the parameters suggested by 5GPPP as target KPIs.

Security problems have been studied by the 5G-ENSURE project and a white paper was released [3]. 5G security will not only be an evolution of existing 3G & 4G security architectures but will have to consider new important aspect. To allow a flexible and dynamic infrastructure, 5G will make extensive use of SDN and Network Function Virtualization (NFV), that require different secure management: *to identify and model attack vectors in this dynamic environment and to be able to offer strong network protection, security control points have to be defined based one establish boundaries between different actors' network functions and slices and their interfaces.*

The 5G physical layer is a crucial point in the standardization process: there are some on-going studies to identify a (possible) global usable spectrum.

In November 2015, in Geneva, the 15th World Radio-communication Conference (WRC-15, [52]) was held, and 7 new bands in the millimeter waves (24 GHz - 86 GHz) have been identified. Therefore the 5G spectrum will be divided into two part: LTE frequency range (600 MHz - 6 GHz) and millimeter waves, with the latter allowing a higher data-rate. Preliminary studies on these bands were completed around mid-2017. EU with the member states worked to find some “pioneer” bands, common across the entire Europe, to allow a preliminary experimental of 5G from 2018. These “pioneer” bands will be classified in 3 groups [2]:

- low-range (up to 1 GHz, focusing on the 700 MHz band). This frequencies will be using to obtain extensive radio coverage and will be mainly used by IoT devices. IoT communication specifications will not be available before 2021 (3GPP Release 17);

- mid-range (1 - 6 GHz, in particular the 3.6 - 3.8 GHz band). This frequencies are a good trade-off in terms of coverage and capacity (i.e. link speed) and currently are used in the 5G Trials that are occurring in the entire Europe;

- high-range (more than 6 GHz). This frequencies will be used for exceptional events (such as concerts, festivals, sport events, etc.) in order

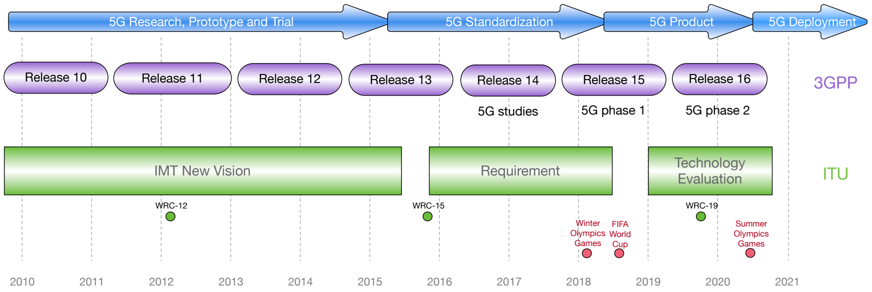


Figure 2.7: Ongoing 3GPP - ITU 5G standardization.

to serve a large number of users in a limited area and not incur in congestion of the existing network.

In WRC-19, that took place in Sharm el-Sheikh (Egypt) in November 2019, the 'pioneer" bands in the millimeter waves was fixed, taking into account the preliminary studies about the 7 candidate bands. The "winner bands" are: 24.25 - 27.5 GHz, 37 - 43.5 GHz, 45.5 - 47 GHz, 47.2 - 48.2 GHz, and 66 - 71 GHz. Using these frequencies, allow to have the 85% of global spectrum harmonization [29].

To adequately support all the scenarios, a flexible and scalable physical layer must be proposed: this new physical layer is called New Radio (NR). NR includes both an improvement of the 4G network and the addition of new standardized functionalities. The physical layer study is out of scope of this thesis, and if the reader is interested in the topic can read [189], and [123].

5G standardization is still in progress and is being carried out by the major standardization bodies (3GPP, ITU-T, IETF, ETSI, IEEE). 3GPP foresees the completion of the first version of 5G by mid-2018 (June - September) with the release 15 where the non-standalone operation[12] will be standardized, and will complete the phase 2 in end-2019 (with the release 16). In figure 2.7 a graphical version of the standardization process is shown.

The networks deployments that will satisfy the requirements of IMT-2020 could be start in 2020. Currently a variety of telecommunication operators around the world have announced the installation of a 5G network to carry out preliminary tests. In the entire world, the trials were started at the end of 2018.

[12] Non-standalone means that LTE control plane is utilized as anchor for the NR, while standalone means that all the entire control and data plane are implemented in the NR.

The new 5G technologies will revolutionize existing business models by providing a new network communication paradigm, able to support communications between humans and objects and to satisfy the constant growth of data traffic. 5G will be a fundamental tool to enable communication between the consumer, distributors and sales companies, allowing customers (not just prosumers) to be proactive towards service providers.

Infrastructures, such as smart meters and the network itself, can be managed more efficiently thanks to better monitoring and more information that can travel with very low latency. 5G will certainly be present in the digital life of citizens and companies, even more pervasively and effectively than it is today through the current telecommunications network.

Chapter 3

National 5G Trial

Some things in life can never be fully appreciated nor understood unless experienced firsthand. – The Twelve Networking Truths - RFC 1925

This Chapter is related to the Italian 5G trial, funded by the Italian Economical Development Minister (MiSE: Ministero dello Sviluppo Economico).

3.1 Project Objective

In March 16, 2017 the MiSE issued a public call in order to collect project proposals in the framework of the European 5G trial (the so-called 5G Action Plan). Open Fiber, Estra, University of Florence, along with other partners, participated to the MiSE initiative for pre-commercial 5G experimentation.

The trial has been divided into three geographical areas and the project proposal winners have been [27]:

- Area 1 - Milan: Vodafone Italia S.p.A.
- Area 2 - Prato and L'Aquila: Wind3 S.p.A., and Open Fiber S.p.A.
- Area 3 - Bari and Matera: Telecom Italia S.p.A., Fastweb S.p.A., and Huawei Technologies Italia S.r.L.

As written in [1]: *Milan, Prato and Bari were selected on the basis of criteria relating to geographical distribution, ultra-fast connectivity, availability of frequencies in the band* 3.7 - 3.8 *GHz, and membership of European corridors. In addition to these selected cities, L'Aquila and Matera have also been identified: the first as part of the post-earthquake reconstruction phase, the latter as* 2019 *European Capital of Culture.*

The trial is scheduled to end on June 21, 2020.

The goal of the trial is twofold: on one hand it will give the opportunity to telecom operators to develop and test the 5G technology using realistic deployments (including the core network). On the other hand, it will enable service operators to acquire useful insights on the types of services enabled by the 5G network, and to plan ahead the kind of backend systems needed to successfully launch innovative services over the 5G network.

As a matter of fact the 5G network will not be "only" a faster 4G, and considering it simply as an evolution is reductive. The 5G network will, of course, allow faster communication to traditional users, but it will also carry completely new kind of services, especially in the area of massive machine to machine and ultra-reliable communications. In our opinion this will represent a major change in the market, as it will allow to deploy new kinds of services, ultimately enabling, for example, the Smart City and Smart Industry concepts.
In this Chapter a Use Case (UC) that are being performed in the Area 2, and in particular in the city of Prato, it will presented.

In the following, works [145], [143], and [142] are explained.

3.1.1 Area 2 - Prato and L'Aquila trial

The Area 2 trial leaders are Open Fiber (OF)[1] and Wind3[2] and the main objective is to realize innovative services for citizens benefit and a new business models between network operators, industry, public administrations and, research centers [26]. The entire trial is called *5G City*.

In Area 2, 7 UC are developed in Prato and 5 UCs are created in L'Aquila. Practically all the UCs are focused on e-Health, Video-surveillance, Virtual and Augmented Reality (VR and AR), IoT, Smart cities, and Industry 4.0.

[1] Italian telecommunications company established with the aim of build an independent fiber-based access network.

[2] Italian telco operator. Wind3 is the leading mobile operator in Italy in terms of customers number.

In the following I'll present the one I was most involved with.

Use Case 5 - Sensors and IoT

The UC 5 is supervised by Estra S.p.A.. Estra Group is one of the leading operators in the energy sector in Italy in natural gas distribution and sale (in terms of distributed and sold volumes), electricity sale, Liquefied Petroleum Gas (LGP) distribution and sale, natural gas trading, in the generation of electricity from renewable sources, and in the energy services and telecommunications sectors.

This use case is aimed at heterogeneous sensors information gathering applications. The goal is to develop an IoT platform for smart city, smart meters, and industrial automation, as shown in Fig. 3.1. The applications involve both mMTC and URLLC type communications, both indoor and outdoor. In this context, three scenarios are considered:

- *Smart city management.* In a smart city, information flow from different area: environment, public and private mobility, social events, etc. The collected data coming from heterogeneous sensors provides an entire view of the city conditions, allowing fast alerts solutions thus limiting the negative effects on citizens coming from unexpected events (e.g., traffic jams, improvise storm, etc.). In this scenario, the main role is played by environmental and structural sensors/applications.

- *Public and private metering management.* In this context, the objective is integrate smart metering services within a single transport network. In this way, the system development will be economically viable. Moreover, the citizen/company can remotely monitor the status of the metering, enabling the system to respond quickly. In this scenario a back-draw is the user privacy and data security: it is mandatory to guarantee that the sensed data will not be used to violate the user's privacy or be used in a malicious way.

- *Industrial processes integrated management.* In these years we are witnessing the fourth industrial (Industry 4.0) revolution and the need for integrated industry management is a fundamental pillar for this innovation processes. The 5G network will provide a capillary infrastructure that will meet the severe requirements needed by industrial Cyber-Physical System (CPS), and is financially suitable.

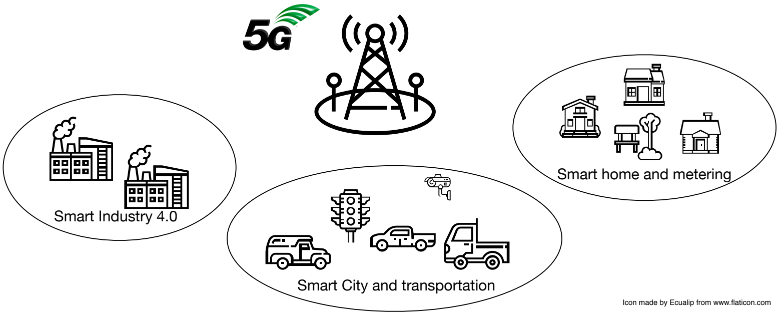

5G
Smart Industry 4.0
Smart City and transportation
Smart home and metering
Icon made by Ecualip from www.flaticon.com

Figure 3.1: Use Case 5 - Sensors and IoT.

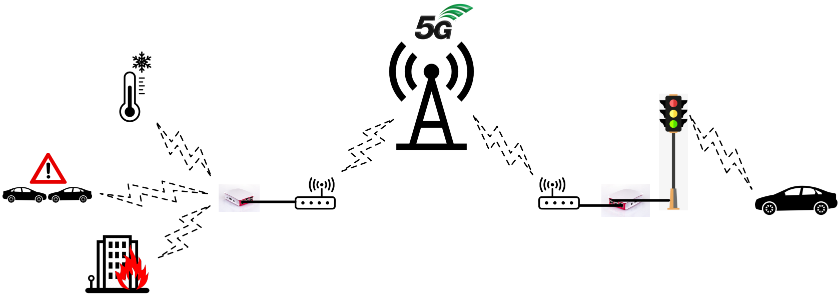


Figure 3.2: The Use Case 5 smart city management.

Sensor networks will be connected together in 5G, allowing companies to implement low-cost custom and centralized management systems using easy and web-accessible tools. 5G technology will reduce the latency time allowing faster reaction in the system to early identify and solve critical conditions.

These scenario will also consider smart devices sub-networks (i.e. sensors, Ad-Hoc network), where the main challenge is related to devices security.

3.2 The first Demo

On January, 31 2019 in Prato there was the first UC 5 demo. In this contest a Smart City management scenario was tested.

Visible Light Communication (VLC) is nowadays seen as a promising technology for supporting ITS. VLC is based on the modulation of the LED-based light source with an information. LED lights allow a fast on-off or high-low intensity switching, realizing a digital modulation of the light signal [133]. Every light source can be thus permuted into a source of information. A photo-diode can act as receiver of this information. In the context of smart cities, the communication between infrastructure and vehicles is important. Traffic-lights are light sources always on, and spread around the city. VLC technology is beneficial since it can support a very low latency together with an energy saving. IEEE 802.15.7 is the standard for local and metropolitan area networks using optical wireless communications, including VLC [37].

5G network and VLC can be seen as a promising coupled technologies to provide an ultra-low latency information to the vehicles all around the urban territory. A VLC system for Infrastructure-to-Vehicle (I2V) communications allows to send broadcast messages to all vehicles that are in front of a single traffic light, slightly increasing the end-to-end latency. Furthermore the VLC system creates a discriminant based on the position of the vehicle, exploiting only the characteristics of the transport layer of the communication.

The problem of continous interaction between Smart Cars and Smart Cities is of particular interest for future urban traffic management. Several ongoing research efforts are devoted to study efficient VANET solutions. Moreover, cars could be equipped with 5G communication modules, providing a full interconnected system. The problem with these approaches lies with the difficulty of precise geo-localization of the cars, and the fact that the car owners should pay data plans for their vehicles (in the direct 5G connection case) or rely on cooperative communication system (in the VANET case), which in turn requires to have a critical mass to enable an efficient communication system.

In our scenario, we propose to use VLC to enable a communication link between cars and the City-managed infrastructure. This has the benefit of not requiring any always on communication system to enable the dissemination of important information between Smart Cars and the Smart City. This approach is particularly compelling, as the traffic light used as a communication link can be interconnected to the Smart City management platform simply by using a 5G device, making this solution cost-effective and easy to deploy.

In Fig. 3.2 the use case is shown. In this scenario both VLC and sensors are used to inform citizen about city status. A VLC system for Infrastructure-to-Vehicle (I2V) communications allows to send broadcast messages to all vehicles that are in front of a single traffic light. Moreover, the VLC system allows to gather precise data about the position of the vehicle, exploiting only the characteristics of the transport layer of the communication.

The benefits of this solution, of course, are not limited to the information broadcasting, and could be used also to collect data about the actual number of vehicles queued at the traffic lights, their planned routes (if the Smart Car system has an active route planning system), or even track a car movement across the city, enabling further improvements to the Smart City management like interactive traffic suggestion and real-time road status updates.

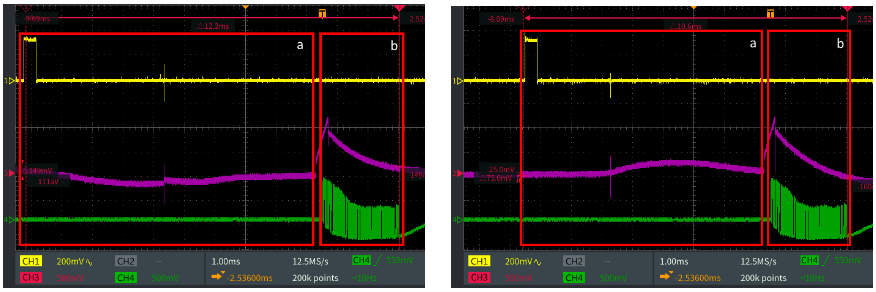
The testbed was realized at the Polo Universitario Città di Prato (PIN) and its scheme is shown in Fig 3.2. The 5G communication link is part of the 5G experimental testbed built in Prato. The used hardware is: 2 experimental 5G Customer Premises Equipment (CPE)s, 2 Raspberry Pi 3 Model B+, 3 WeMos D1 mini, a traffic light compliant with the Italian regulation, a VLC transmitter, a VLC receiver, and some sensors:

- *flame*: used to simulate a fire;
- *gyroscope/accelerometer*: used to detect an incident between two (scale model) cars;
- *temperature/humidity*.

One Raspberry is used as a data collector and forwarder from the sensors to the 5G network, while the second emulates the unit that should be positioned inside the Traffic Light. Both are relatively simple devices, the first having mainly the goal to collect and process the sensors data, the second to convert them into a format suitable for the VLC transmission. The three WeMos D1 mini are used to pilot and control the sensors. The transmitter and receiver are experimental devices built by the authors.

The data collected by the temperature/humidity sensor are continuously collected and broadcasted on a periodic basis, while the other sensors data are processed by the control unit and forwarded only when an alarm condition is detected (e.g., fire, ice, car collisions, etc.).

This setup allowed to both test the system for continuous data dissemination, and for alarm-like event types, where the first kind has a higher tolerance on losses and delays. It is worth noticing that, due to the testbed hardware current limitations (to be removed in the future), the devices are connected in the following way:



(a) Maximum end-to-end latency time. (b) Minimum end-to-end latency time.

Figure 3.3: Measured latency time.

- Sensors (WeMos) to Raspberry: Wi-Fi connection managed by the Raspberry in Access Point mode;
- Raspberry to 5G CPE: Wired GBit Ethernet connection;
- Raspberry to VLC transmitter in the traffic light: USB connection.

All the code in the Raspberries is written in Python 3 and all the WeMos code used to drive the sensors is written in C/C++.

3.2.1 Result

The goal of the testbed was twofold: we wanted to demonstrate the feasibility of the system, and to collect a set of preliminary data on the performances of the testbed. The measures are aimed at performing a preliminary evaluation of the delays introduced by the various devices and communication links, and to analyze what kind of optimizations should be made in order to achieve better performances in terms of latency and packet loss.

The result have been collected using an oscilloscope attached at the two Raspberry devices network adapters and the VLC receiver. The results are shown in Fig 3.3, where the yellow line represents a signal transmitted by the sensors-side Raspberry on the Ethernet cable, the pink line a signal transmitted by the traffic light Raspberry on the USB cable, and the green line the arrived packet at the VLC receiver.

The yellow line, when high, represents an alarm trigger, in the pink line the rising edge represent the time necessary to forward the packet on the USB cable, the green line represent the data at the VLC receiver. The packet is considered fully received when the data are completely received, i.e., when the green line stops oscillating.

We performed several measures, which we will omit for brevity. Without loss of generality, we will present only Figures 4(a) and 4(b), which represent respectively the maximum and minimum measured end-to-end latency time.

In the Figures we did highlight the latency introduced by 5G and VLC connections. The two rectangles named *a* and *b* represent the latency introduced by 5G and VLC, respectively. The portion between the two rectangles represent the processing time introduced by the second Raspberry.

From these measurements, it is evident how the larger latency contribution is due to the 5G network (7.5 to 10 ms), while the VLC latency is practically constant and equal to 2.5 ms. The raspberry processing time is negligible (some μs).

The VLC performance is in line with the data transmission time required to send the packet through the optical channel at the rate of 100 Kbps, as defined in the IEEE 802.15.7 standard [37] for outdoor applications. As a consequence, it can be lowered only by decreasing the packet size or increasing the VLC channel rate.

The 5G latency is acceptable for this kind of applications, and could be partially optimized by decreasing the packet size, or by using data prioritization in the 5G core network. Considering that the 5G network used in the testbed is an experimental system, it lacks some optimization features being foreseen in the 5G deployed networks. Moreover, we used two 5G CPEs, equivalent to the use of two 5G links (CPE - BS + Core Network and back). In a Smart City deployment, we could expect to use only one link, with the sensors sending their data to a Smart City control center, connected directly to the 5G Core Network, leading to shorter latency on the 5G part.

As far as the packet loss, we were unable to detect a statistically relevant number of losses due to communication link failures (we had a few due to hardware failures or human errors). Further automated measurement campaigns will be necessary to estimate the packet loss probability.

In this demo the 5G experimental activities in the city of Prato is outlined. In particular, the use case of 5G network supporting VLC communications between traffic lights and vehicles.

The results show that 5G and VLC can be successfully coupled to obtain a very low latency and low cost communication system for data dissemination

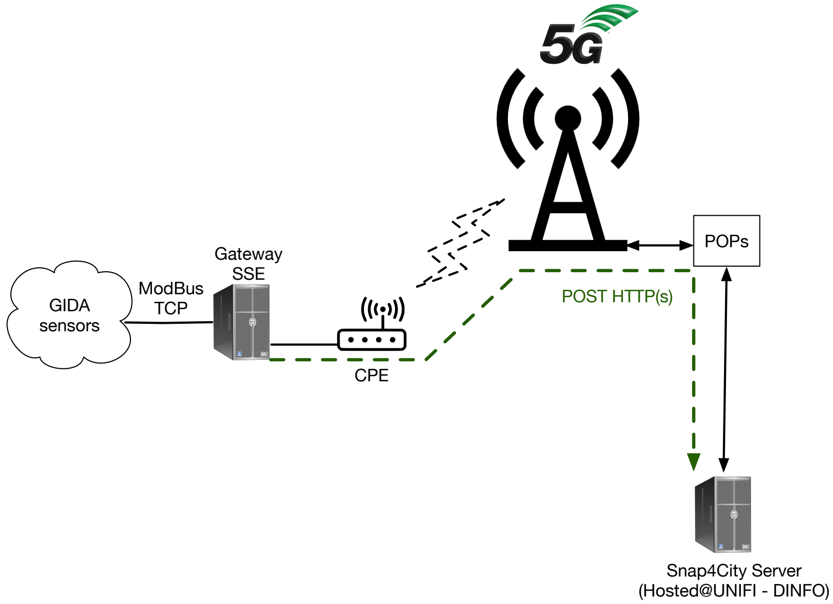


Figure 3.4: The Use Case 5 industrial processes integrated management.

to Smart Cars, enabling a number of Smart City mobility applications. The only caveat highlighted is represented by the need to negotiate a proper message delivery Quality of Service between the Smart City application and the 5G network provider, in order to ensure a prompt delivery of the alarm-type messages.

3.3 The Second Demo

On October 17, 2019 there was the second UC 5 demo. One of the great novelties compared to the previous demo was the first 5G Stand Alone site deployed by ZTE.

The demo was performed in collaboration with GIDA - Gestione Impianti Depurazione Acque and SSE - Sirio Sistemi Elettronici, so an Industry 4.0 scenario was presented. In Fig. 3.4 the demo architecture is shown.

The objective of this demo is to integrate 5G and GIDA sensors in optics URLLC. In this scenario, the sensors were polling through ModBus-TCP protocol [3] in a continuous loop and send to a cloud platform through 5G network.

To poll the sensors (for security concerns, we are authorized to interrogate only pumps status and their absorption in kW) a gateway loaned by SSE was used. The code was written in python3 with the framework PyModbus (a pure Python Modbus protocol stack) and a custom class to manage the HTTPS POST/GET to the cloud platform.

3.4 Role of the networking simulators

Network simulators play a central role in academia and industry alike. They can be useful to evaluate new network protocols, verify mathematical models, explore network dynamics, get insights about the relationships between algorithms parameters, etc.

Of course network simulators can not be the only tools available for research and implementation, and testbed can be equally important.

Nevertheless, the testbed and network simulators can and should be used to complement each other shortcomings.

Thanks to our experience in the 5G testbed in Prato, we can affirm that the main shortcomings are the following:

1. Limitation in the deployment - there are practical issues in the setup of a large-scale testbed, like the number of available sites, the number of devices (particularly critical in case of a new technology), etc.

2. Limited or null possibility to tweak the core network setup and algorithms, as they are both proprietary, and experimental.

3. Poor experiment reproducibility, both due to channel fluctuations, and to different software releases on the terminals and core network.

Due to the above reasons, we argue that the use of a network simulator can be extremely useful, not to substitute the testbed, but to complement its results.

However, given the complexity of the 5G standard, a complete stack development seems to be not foreseeable in the very near future. Moreover, any

[3] ModBus is a serial communications protocol developed in 1979 for use with Programmable Logic Controllers (PLC). ModBus-TCP is a variant of the serial ModBus used for communications over TCP/IP network.

development would need to be coordinated, in order to allow the researchers to focus on the relevant ‘implementation dependent’ parts.

In particular, we think that the following topics should be considered as a priority in the 5G simulation development.

3.4.1 Traffic Models

At the moment the ns-3 traffic models are highly debatable. Even though they can be used for research purposes, they are mostly based on ‘traditional’ traffic types.

Since 5G aims at new market types, it is necessary to develop new traffic models for the mMTC and URLLC cases.

Moreover, as pointed out by the ns-3 developers multiple times, it is necessary to abstract the traffic generators from the actual transport protocols, in order to be able to simulate traffic aggregation and proprietary transport protocols.

3.4.2 Transport Protocol Models

Even though TCP/IP is the protocol of reference for 5G networks, it is also true that other transport protocols exists, in particular in industrial automation cases.

In order to ease the development of new protocols, it would be necessary to provide ‘example’ implementations of bare-minimum L3 and L4 protocols, to be used as a guideline for the developers. As a matter of fact, most ns-3 models are not meant to be used as an example because they are either too simple or too complex.

The addition of example protocols and models could be beneficial both to the academia and the industry.

3.4.3 Network Slicing Models

One of the major features in 5G networks will be the Network Slicing.

In order to effectively simulate the effect of slices, it is important to have models for the MAC Scheduler/Resource Broker, and the MUX (see Figure 3.5). These are the two most important element, and they are also the one responsible for maintaining the QoS of the slices.

Equally important will be the capability to “load” the slices with aggregate, realistic, traffic patterns.

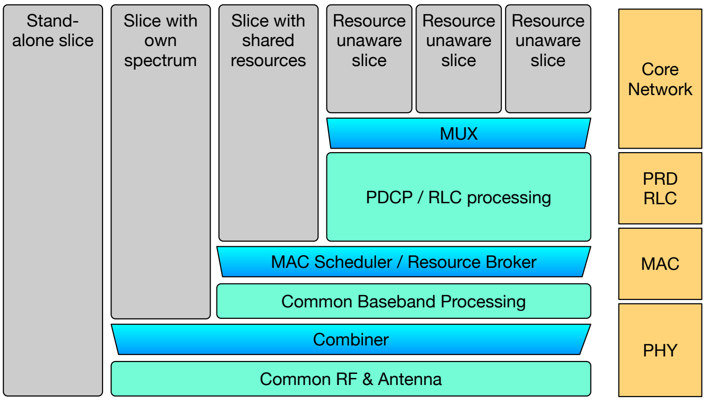


Figure 3.5: Layer of Network Slicing in 5G.

3.4.4 Core Network Performance Models

Thanks to the testbed experiments, we noticed how the Core Network performances are important in the 5G network.

Since one of the main goals of 5G is the *low latency*, and due to the heavy use of SDN technology in the Core Network, the Core Network (CN) performances should not be simulated with the appropriate (statistical) delays.

Simulating the whole Core Network is probably not a good idea. Moreover, vendors will not disclose publicly the exact implementation, leaving little or no chances for a realistic simulation. Nevertheless, it should be possible to add statistical delays to the packet processing, possibly following an empirical distribution. Moreover, the statistical delays should be time-dependent, as we noticed how the SDN implementations tend to add different delays according to specific patterns, e.g., upon 'forgetting' a flow, a node must interact with the SDN controller, adding a further delay to some packets of a flow.

3.4.5 Scalability and Extended Time Simulations

Some of the 5G application scenarios (in particular mMTC) involve very low bandwidth traffic flows. This poses a problem, as a single device might have a data-rate almost negligible, but the device density in the order of $10^6/\text{km}^2$.

It is evident that, in these conditions, to have statistically relevant results, a simulations should encompass days or weeks of simulated time. If the 5G stack is simulated at packet level, this means that the simulation would be too slow for any computer.

In order to overcome these limitations, we argue that it should be possible to implement the following:

1. A *statistical* model for the 5G stack, possibly self-generated by ns-3 simulations at different time scales,

2. An aggregate traffic model for low-priority traffic types.

The first would be useful to simulate the performance of mMTC traffic in presence of other kind of traffic types, while the second would be useful to do the opposite.

At the moment it is possible, and easy, to define aggregate traffic types. The other feature, as is to statistically simulate some layers of a standard, it is not. In order to do so, a 5G model should be designed with the specific goal of having some parts substituted by equivalent blocks implementing only a statistical behavior, and to be able to cross-validate them with the full stack implementation.

Moreover, it would be beneficial to evaluate new methods to speedup the simulations. Two candidates are:

1. A GPU-enabled spectrum model;

2. A stronger support for multi-core, multi-trad simulation system.

Using Message Passing Interface (MPI) in ns-3 is actually possible, but it is strongly limited. As an example, it is not possible to use MPI along with wireless channels. This issue is not easy to fix, however, it should be considered as a priority. About GPU-enabled spectrum model some computationally intensive tasks, and in particular the ones involving the channel interference models, could be effectively offloaded to GPU using CUDA or similar programming techniques.

Chapter 4

Network Protocols for IoT

All that we are is the result of what we have thought. – BUDDHA

In this chapter the two protocols developed during my Ph.D. studies are presented.

4.1 A Distributed Token Passing Protocol for Time Constrained Data Gathering in Vehicular Ad Hoc Networks

In this section a novel approach for time constrained information gathering in a typical VANET is presented. This novel approach is based on a token passing scheme, adapted to wireless communications by creating a virtual ring where nodes are connected to a predecessor and a successor node. To address the typical fast topology changes of VANETs, we proposed a specific approach, called *Tom Thumb*, that is a distributed protocol that node-by-node circulates a special packet, called *token*, which collects the information stored in each vehicle until returning to the *first* unit within a specified time constraint. The protocol has been properly designed in terms of (i) the more effective hop-by-hop and distributed heuristic implementing the objective function ii) the token packet format, i.e., the syntax and semantics of its fields. Finally, the performance of the proposed approach is validated for different time constraints and numbers of vehicles, always pointing out a remarkable gain, especially in the presence of severe constraints, i.e., in terms of time deadline, collected information amount and success probability.

This work [72] was published on MDPI Electronics, Special Issue: Vehicular Networks and Communications.

VANET are a Mobile Ad Hoc Networks (MANETs) special case, typically allowing both V2V or V2I wireless communications [104, 177], to support complex Intelligent Transportation System (ITS) applications [105,172]. The main feature of VANETs is represented by the constrained and correlated node mobility *patterns*, which in turns implies an extremely time-varying network topology, with network *partitioning* and *merging*. Despite this drawback, a VANET is expected to give rise to an intelligent and cooperative eco-system in order to improve the driving experience, with special regard to safety applications [71].

A commonly referenced implementation of the VANETs paradigm is represented by the IEEE 802.11p protocol [30], which is complemented by the IEEE 1609 protocol suite [107,179]. The set of IEEE 802.11p and IEEE 1609 is usually referred as Wireless Access in Vehicular Environments (WAVE) in the U.S., while in Europe the corresponding is the cooperative-ITS (C-ITS) based on ITS-G5.
In 2017, 3GPP introduced the Vehicular-to-everything (V2X) communications in the Long Term Evolution (LTE) Release 12 standard, considering new direct device-to-device (D2D) communication mode as a way to effectively support both public safety services and for Proximity Services (ProSe). D2D, indeed, introduces *direct* single-hop communications between two devices, with limited or even without any Base Station (BS) support [48, 165]. These direct (or proximity) communications can achieve high data rates with low end-to-end delay, whilst they are time- and energy-consuming, due to the use of beaconing signals and scanning for direct user discovering. Moreover, as D2D communications usually use *unlicensed* band like Wi-Fi, they have the drawback that coverage and number of discovered devices are limited due the stochastic nature of interference over these bands. On the other hand, if a cellular network supports D2D communications (the so-called *in band* D2D communications), it is also in charge of discovering D2D candidates, managing time-frequency resources, which could overload the cellular infrastructure [67], [69].

This work considers a VANET-aided D2D discovery scenario for data

gathering, that offloads part of the signaling network traffic to the VANET for allowing low-rate D2D communications as in [74]. In the literature, broadcast approaches are commonly addressed mainly to deliver messages to the desired destinations, additionally via intermediate nodes. However, this approach can increase the traffic volume leading to the well-known *data storm* effect.
As a consequence, an open issue is represented by the design of a protocol, in alternative to the classic store-and-carry routing paradigm, that manages the data gathering and dissemination, together with the on-the-way information processing. To this purpose, in [83] the authors aimed at introducing the Token Ring (TR) approach in an IEEE 802.11 wireless network, which is referred as Wireless Token Ring Protocol (WTRP). WTRP is a Media Access Control (MAC) designed for Ad-Hoc network with dynamic topology and stringent requirements (i.e. band, latency and failure recovery). In particular, it arranges the network topology as a unique *ring*, in which every node has a *predecessor* and a *successor*, which are a priori known before any transmission occurs. Despite WTRP has been soon patented [64], the principle behind it has been investigated with reference to MANET domain [169] and, in particular, it has been adopted in VANET for information collection and dissemination [56, 97].

In this work, we propose a protocol inspired by WTRPs seminal work [83], where significant differences have been introduced: (i) token circulates in a network that is dynamically and *step-by-step* formed, i.e, the initial unit (for example the traffic light) starts the process of token packet sending it to only *one* neighbor node (i.e., the one with higher information), which in turn selects in the next round only one vehicle and so on until the time deadline is reached, (ii) there is no a priori or proactive neighbors discovery, while it is *in-path* performed, (iii) in every step, after possible locally storing and processing, data are passed to the *best* next hop in order to perform a distributed information gathering, thus this approach is close to a token passing, (iv) the number of visited devices can change dynamically according to the traffic conditions and mobility patterns, (v) the token is delivered to the initial unit creating a *virtual ring* within a time deadline (vi) the token packet format, i .e, syntax and semantics of its fields, and the protocol phases are tailored to this specific use case.

Specifically, we characterize a novel protocol, called *Tom Thumb* (TT), suitable for information gathering and sharing within a VANET, by adopting D2D communications among vehicles in an urban environment. The proposed approach can be adopted to support typical *smart cities* missions such as an improvement of traditional infrastructure through Information and Communications Technology (ICT), enhancing the quality of life for citizenship, companies and institutions [190]. In this context, each vehicle, which can be conveniently assumed to be paired with a smartphone used by drivers or passengers, shares its local context awareness with its neighbors belonging to a VANET group. In particular, our proposal resorts to a token passing protocol scheme initiated and terminated in the *same* device (e.g., the traffic light) creating , which iteratively discover the nearby vehicles and collect information for traffic monitoring and control applications, as expected in a typical smart city scenario. The selection of the best *closed* path is a typical NP-hard optimization problem, as it could analyze all the possible paths among vehicles, to select the one maximizing data gathering while matching a predefined time constraint; in particular this problem is unaffordable at the increase of the number of vehicles [97]. In addition, in the presence of vehicles mobility, a centralized approach requires the instantaneous and ideal knowledge of vehicles positions, which can be achieved at the expensive of a prohibitive signaling overhead. To this purpose, the proposed TT adopts a sub-optimal approach by hop-by-hop selecting the best path and then concatenating local optima, instead of evaluating the global one. In particular, it does not visit all the nodes but driven by in path processing it selects the path collecting an information quantity near to the optimum. However, TT is close to the optimum approach in terms of data gathering, as it has been validated by numerical results with different time constraints.

4.1.1 Related work

Intelligent Transport System (ITS) covers a wide range of applications for vehicles, such as driving safety and warning, up to automated driving, information exchanges among vehicles and Internet applications. VANETs [192], a special class of MANETs [176], is a key component for developing current ITS. As in MANETs, the VANET network paradigm has no fixed infrastructure and instead, the members of the network themselves, i.e., the vehicles, provide network services. VANETs devices are classified into two categories: Road Side Unit (RSU) (e.g., traffic lights) and On-Board Unit (OBU) (e.g., wireless device embedded on vehicle) [95]. In general, communications in VANETs typically allow vehicle to vehicle (V2V) or vehicle to

infrastructure (V2I) wireless interfaces. VANET is a consolidated research topic because it has shown to be an excellent element to improve vehicle and road safety and traffic congestion avoidance. In order to implement this kind of services, cooperative awareness plays a crucial role. WiFi- based Vehicle -to-everything (V2X) communications technology represents a mature candidate, whose capabilities have already been tested. Presently, IEEE 802.11p and the corresponding ITS-G5 in Europe are the current standards. As an alternative, 3GPP considers the support to the V2X feature in cellular networks (i.e., LTE and the next generation 5G). Recently, D2D communications (or Proximity services) are introduced to avoid the control signaling overhead to register and synchronize all the vehicles with the cellular infrastructure. Indeed, D2D communications allow autonomous devices discovering and exchange of data without the presence (or with limited presence) of the cellular infrastructure (i.e. the eNodeB). An overview of standardization directions for vehicular communications can be found in [87]. Furthermore, in the literature, several works presented the joint use of IEEE 802.11p and infrastructured LTE, as in [59, 94].
Several medium access control (MAC) layer protocols are designed for V2X for an efficient coordination among the nearby vehicles and for reliable transmission mechanism of safety message. Among these different proposals, the Token Ring (TR) protocol [167] has been also considered to arrange nodes topology in a ring. However, our proposal avoids the network formation, while a token passing scheme is instead adopted.

Token-based Medium Access Control protocols for VANET

One of the most challenging research topics in VANET is to efficiently design a scheme for accessing the network medium. There has been some effort in the research community to adapt the classical wired-based medium access TR protocols to wireless communications. The reliable neighbor-cast protocol (RNP) [128] uses token as acknowledgment message to deliver reliable multicast among nearby vehicles selected by a voting scheme as a neighboring group. Wireless TR protocol (WTRP) has been presented in [83] as a novel protocol for wireless environment designed for Ad-Hoc network with dynamic topology and stringent requirements (i.e. band, latency and failure recovery). The aim of WTRP was to guarantee the quality of service (QoS) in terms of bounded latency and reserved bandwidth. Additionally, this protocol improved the access efficiency by reducing the number of re-transmissions due to collisions. This was done by creating a *virtual* ring where nodes, upon joining a network, are required to be connected to a predecessor and a successor node. However, WRT is applied to ITS for guaranteeing QoS and recovering from multiple failures, but this protocol does not react to the typical fast topology changes of VANETs.

Since the WTRP protocol was proposed, some other improved versions of this protocol have been published in literature. In [80], an enhanced version of the WTRP token passing-based MAC protocol is presented (EWTRP). The improvements in comparison to WTRP include a preemption mechanism, a hibernation mechanism and a contention mechanism. A comparison between EWTRP and WTRP was carried out through analyses. Results show that EWTRP produces higher throughput while it consumes less power and is more suitable to operate in small-scale wireless ad hoc networks.

Additionally, the Ripple protocol is presented as a wireless token passing-based protocol specially designed for wireless mesh networks in [65]. This protocol improves existing random-access approaches by proposing a decentralized controlled-access protocol to protect nodes from unintentional packet collisions and to enhance the throughput of the network.

In [124], the authors of this study propose a token ring-based protocol for multi-channel routing to improve the performance of mesh networks. This performance is obtained by defining delay-guaranteed rules for the actions of joining a ring and creating new rings. The authors present a state machine to define their protocol. Analytical results on bounded delay are presented and show a good performance when compared to state of the art approaches.

Furthermore, Sun et al. propose an automatic adjustment of the token path w.r.t. to the subnet's dynamic topology which is not always arranged in a ring shape [168]. Furthermore, the operation of token path maintenance is simplified and the channel efficiency is increased also in WDTP.

In [53], Bi et al. propose a multi-channel TR media access control protocol (MCTRP) for inter-vehicle communications. By means of an adaptive ring coordination and a channel scheduling, vehicles are autonomously managed into different rings, operating on diverse service channels. The authors show that this topology management allows special messages, such as emergency messages, to be disseminated with a limited delay. Additionally, they present a token based data exchange protocol that improves the network throughput for non-safety multimedia applications.

In the Overlay Token Ring Protocol (OTRP) [193] vehicles are arranged into *multiple* overlapped virtual rings, where a unique token performs the control functions. The ring architecture dynamically follows the traffic condition to provide data transmission with QoS and reliable safety messages exchange.

In this paper, differently from the OTRP scheme, the ring is not a priori formed, but the neighbors discovery is carried out hop-by-hop *without* a complete list of nearby devices, such that the resulting ring has not a specific size, i.e., it is just a *virtual* ring originated in the initial unit (e.g.,the traffic light), connecting the visited vehicles, and reaching again the initial unit in order to gather the collected information within a time deadline. Besides, in the OTPR scheme, only few devices (about 6) can be grouped together to form overlapped rings; as a consequence, this is more suited for dynamic traffic condition, rather than for an emergency scenario, where it is needed to exchange messages among a high number of vehicles.

In the literature, several research papers are mainly focused on *dissemination protocol* as shown in recent surveys [39, 62]. Blind data flooding is the simplest dissemination way but has limited performance due to a high percentage of redundant data and packet collisions which lead to the broadcast storm problem. Indeed, Geocast data dissemination protocols consist of sending data only to vehicles inside a specific geographical area [45]. Recently, there are in particular novel and efficient proposals identifying a small group of vehicle for re-broadcasting the messages [180].
Data dissemination is more stringent for Vehicle Delay Tolerant Networks (VDTN) with intermittent and opportunistic connections among vehicles than VANET. For example, in [120], the D^{2}NFCE algorithm determines the node forwarding capability for data dissemination first by evaluating the effective connection time period then building a predictive traffic model based on wavelet neural network and finally, fitting the historical and predictive throughput of vehicles.
On the opposite, *data gathering* schemes are recently proposed for VANET. Data collection protocol can be organized with a centralized node , e.g., a cluster head (CH) which collects the data of its vehicles and delivers it to the initial unit (e.g. the RSU) using flooding tecnhique , as in TrafficGather protocol [61]. The centralized node can be the eNodeBs of LTE network in urban enviroment which creates several clusters of vehicles. The cluster topology is broadcasted to the vehicles by eNodeB. Each cluster head delivers the aggregate data to the eNodeB which deletes redundant and undesired

data, as shown in LTE4V2X [157].
A centralized medium access technique based on space division multiple access (SDMA) is also used in the Clustered Data Gathering Protocol (CDGP) organizing the data collection with the election of a cluster head among the various segments in which the road is divided. The CH assigns a slot to the vehicle and if data are not available the entire slot will be lost. The use of a token is considered by the same authors to reduce the loss of slots in the new versions i) the token based cluster data gathering protocol (TCDGP) and ii) its enhanced version Distributed Data Gathering Protocol (DDGP) [56]. Both proposed protocols consider the election of a vehicle as a CH within a specific area of an highways to collect data, while Baiocchi et al. [178] extend this protocol towards an urban environments. Specifically, a sub-set of vehicles is selected to act as relay nodes (RNs), thus creating a temporary backbone network that can be used for data dissemination and collection. In the *discovery* phase, the selection of vehicles traveling in the region of interest is achieved by broadcasting a request message from the RSU, which is in turn forwarded across the formed network in a multi-hop way. Differently, according to our proposal, each selected vehicle is responsible to iteratively select the best neighbor node (i.e., the one with more data stored on board), thus forming a truly self-organized VANET, without HELLO messages broadcasting.
Other protocols are based on random medium access scheme, such as the Clustered Gathering Protocol (CGP) [66]. In this case the road is organized in virtual segments of the same length. In each segment a geographical cluster is formed and a CH is elected to collect and aggregate data from vehicles present in that segment, before sending these data to the next segment or to the base station (BS). However, CGP is specifically designed only for one-way road and for larger region a greater overhead is expected due to the flooding approach adopted to forward the collected data towards the initiating unit, i.e, the BS. In addition CGP suffers for the high number of collisions.

4.1.2 System Model

In this work, a distributed message passing-and-processing protocol to support intelligent traffic management systems is presented. In particular, according to the 5G vision, the V2X communication are adopted mode properly mapped to the existing IEEE 802.11p air interface. In this way, vehicles are allowed to directly communicate with each other, without involving the eNodeB scheduling. In addition, the authors refer to a particular communication pattern, i.e., one-to-many-to-one scheme, according to which mobile devices (i.e., OBUs) process and collect information in a *cooperative* way, once they have been triggered by one RSU. It can be noticed that the RSU can act as a gateway sending the collected information to its own eNodeB, which in turn can transfer it to a Big Data Cloud for data analytic and decision making process.

Following the approach proposed in [83], a message passing scheme among VANET devices is relying on, namely TT, where the message is denoted as *token*. The typical use case is depicted in Fig. 4.1, consists in a *group* of vehicles in the proximity of a traffic light. As soon as they stop, for instance when the *red* light turns *on*, the procedure of data collecting is started, involving the cars temporarily queued, until red light turns *off*. The first step is carried out by the RSU associated with the traffic light, that discovers the surrounding cars selects the best one and passes the token to it, waiting for its return within a time *deadline* (typically the red light period). Then, the token is iteratively passed among devices, where each one stores and possibly refines the carried information in order to improve their *collective* context awareness. Finally, the token has to be sent to the initial RSU, before the timeout (i.e., closing the *virtual* ring), otherwise, it is considered lost and no information is eventually gathered. The proposed scenario can be generalized in order to take into account a limited mobility pattern which models a road segment affected by traffic congestion or a large and saturated roundabout, where information gathering is even more important.

Denoting with $\mathbf{P}_{i,j}$ the set of all the possible communications paths connecting the i-th and j-th generic couple of devices in the VANET, and with with e_k the local information available at the k-th vehicles, the objective of collecting the largest information in a limited amount of time through the following constrained optimization problem is formulated as:

$$\text{(OPT)} \qquad \max_{\pi \in \mathbf{P}_{\alpha,\alpha}} \mathcal{F}\left(\mathbf{e}_{\pi}\right), \tag{4.1}$$

$$\text{subject to: } \left|\tau\left(\pi\right) - \tau^{*}\right| = 0, \tag{4.2}$$

where α conventionally represents the first RSU initiating the protocol, $\mathbf{e}_{\mathcal{J}}$ is the set of information associated to the vehicles belonging to the path π, $\mathcal{F}()$ is function modeling the join information processing and distribution, $\tau\left(\pi\right)$

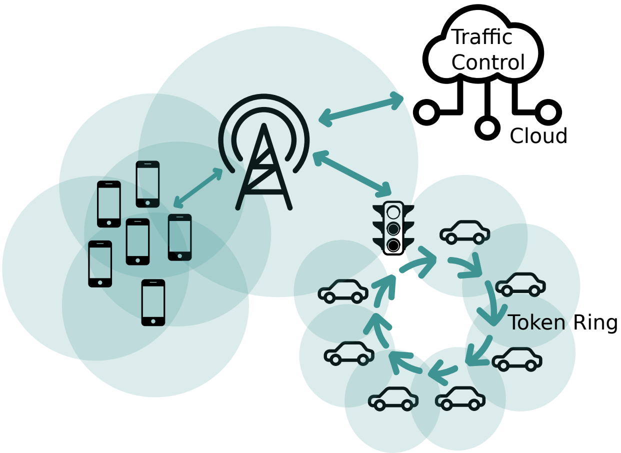


Figure 4.1: Reference scenario representing the typical use case for the proposed protocol.

is the cumulative latency associated to the path $\mathcal{J}$ and τ^* the considered time deadline. The constraint means that the delay $\tau(\mathcal{J})$ associated to the path $\mathcal{J}$ needs *exactly* to match the time deadline $(\tau^*)^1$.

For the sake of model simplicity, the distributed information processing consists in a data *gathering* is considered, i.e.,

$$\mathcal{F}\left(\mathbf{e}_\pi\right) \doteq \sum_{i \in \pi} e_i, \tag{4.3}$$

this meaning that every involved device (except the initial RSU) sums its local information to the received one, as explained in Section 4.1.3. In addition, it is worth noting that if the j-th device is visited more than one time (e.g., in the presence of a loop) the provided information is *null.* In some specific applications, known as consensus sensing, it is further required that the provided cumulative information $\sum_{i \in \pi} e_i$ was greater than a threshold value E^*.

It can be finally noticed that the optimization procedure solving eq. (4.1) with the additive information processing model of eq. (4.3) analyses all the possible multiple paths among vehicles in order to maximize the data gath-

[1] This constraint could be further relaxed, assuming that the difference is lower than a small time value $\epsilon > 0$.

ered presents a NP-hard complexity, which is particularly burdensome at the increasing of the number of vehicles. On the contrary, the proposed TT approach is a sub-optimal heuristic, that does not visit all the nodes, but only a small subset depending on the selected vehicles and the adopted in-path processing. However the amount of collected information is near to the optimum value, as shown by numerical results performed in Sect. 4.1.4.

4.1.3 Proposed Protocol

The proposed protocol consists into two main phases and an optional one, the latter executed only under specific conditions:

1. *token processing*
2. *neighbor discovery*
3. *next Token Owner (TO) selection & token forwarding*

In the following, each phase in terms of exchanged messages and sequence diagram is characterized.

Phase I: token processing

The protocol design is starting by illustrating the token packet structure, as shown in Fig. 4.2 along with its fields. Each field and its length (in bit [b] or Byte [B]) is defined. There are three possible packets that have a common field (*Type (Tp)*) formed by 2 bits which identifies the current packet. In particular, if this value is `00` the packet is the token; if it is `01` the packet is a Neighbor Discovery Broadcast (NDB) message; while it is `10` the packet is a Neighbor Discovery Response (NDR) message. The `11` value is not yet assigned. In addition, a *PADDING* field is inserted: it is either 30 bits for NDB and NDR packets or 8 bits for the token. In both cases, this field is set to 0. When nodes receive a packet, parsing the first two bit it identifies the message and can perform the correct operation.

Regarding the token packet the following fields are settled:

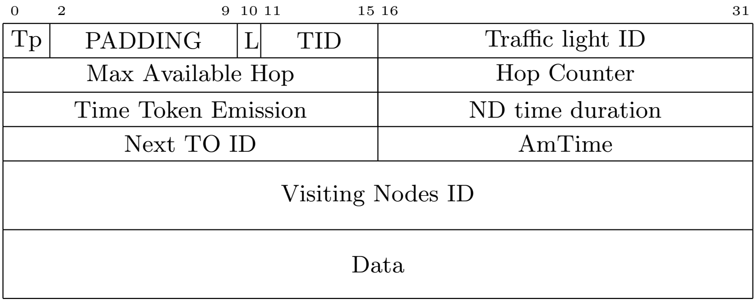
- *Link (L)* [1b]: this field contains the token route. If it is 0 the token is in Discovery Path (DP), if it is 1 the token is in Return Path (RP). In the DP mode the token discovers the route and searches the best next TO, while in the RP mode the token returns to the Traffic Light (TL) following an optimized route evaluated in the discovery mode.

- *Token ID (TID)* [5b]: this field contains the token *unique* identifier (ID).
- *Traffic light ID* [2B]: this field contains the RSU traffic light unique identifier.
- *Max Available Hop (MAH)* [2B]: this field contains the maximum number of hops in discovery mode. This value is given by Algorithm 2. The TL evaluates the maximum hop number divided the amount of time for the neighbor discovery (ND). If the evaluated time to perform the discovery path and the estimated return path (the RP hops number is less or equal to the DP hops number)[2] is bigger than the amount of time, it decreases by one the MAH and re-calculates the time.
- *Hop Counter (HC)* [2B]: this field contains the token hop number. When a node receives the token it increases this field by one if the token is in DP mode, while it decreases this field by one if the token is in RP mode.
- *Time Token Emission (TTE)* [2B]: this field contains the seconds that have elapsed since an epoch[3].
- *ND time duration* [2B]: this field contains the time (in *ms*) used for the ND.
- *Next TO ID* [2B]: this field contains the *next* TO identifier.
- *Amount Time (AmTime)* [1B]: this field contains the amount of time to perform token passing (in seconds). Within this time, the token must return to the TL. It is worth noticing that TTE could be different with respect to the red time start to allow the vehicles queue to be formed.
- *Visiting Nodes ID (VNsID)* [variable length]: this field contains all the nodes visited in DP mode, or all the nodes that will be visited in RP mode. The field length is extracted by the *Hop Counter* field. The queue is managed according to the last in first out (LIFO) discipline in DP mode and first in first out (FIFO) in RP mode. In other terms, in the discovery mode the first ID is the first node visited by the token, in return mode the first ID is the next TO.

[2] In the discovery path the ND time duration is much bigger than the transmission time.

[3] For example, Monday, 1 January 2019, 00:00:00 UTC.

- *Data* [variable length]: this field contains the data processed by the devices visited by the token, in according to EC principle.



0	2 9	10	11 15	16 31
Tp	PADDING	L	TID	Traffic light ID
Max Available Hop				Hop Counter
Time Token Emission				ND time duration
Next TO ID				AmTime
Visiting Nodes ID				
Data				

Figure 4.2: Token packet structure.

Within this phase, the device selected as the actual TO checks the *L* field and, according to Algorithm 3, it enters the correct phase.

Phase II: Neighbor discovery

This phase is performed only under specific conditions, i.e., if the token is in Discovering Mode. Accordingly, the TO discovers the nodes in its coverage radius. To this purpose, TO sends a NDB packet (shown in Fig. 4.3) with its unique ID (its role is equal to a *source address*) and the duration time of the neighbor discovery phase.

After sending NDB, the TO enters into a *waiting state*, waiting for neighbor responses within a predefined time. The presence of explicit Acknowledgment (ACK) messages is considered, as it usually happens for commonly adopted medium access schemes.

As explained in Section 4.1.3 the time duration is fixed by the red period of traffic light. Upon received the NDB, every node sets a timer, and if it does not send the NDR within this time, it aborts the transmission. Due to the limited coverage radius, the neighbors are very close: this assumption allows to identically evaluate the NDB received time.

All the nodes that received the NDB packet try to randomly access the medium in order to send their NDR within the ND duration time. Specifically, the NDR has the following fields:

- *Vehicle ID* [2B]: this field contains the unique ID of responding node.

<table>
<tr><th>0</th><th>15</th><th>16</th><th>31</th></tr>
<tr><td>Tp</td><td colspan="3">PADDING</td></tr>
<tr><td colspan="2">Token Owen ID</td><td colspan="2">ND time duration</td></tr>
</table>

Figure 4.3: Neighbor Discovery Broadcast packet structure.

<table>
<tr><th>0</th><th>15</th><th>16</th><th>31</th></tr>
<tr><td>Tp</td><td colspan="3">PADDING</td></tr>
<tr><td colspan="2">Vehicle ID</td><td colspan="2">TO ID</td></tr>
<tr><td colspan="2">Queue arrival time</td><td colspan="2">Data length</td></tr>
<tr><td colspan="4">Carried information (data)</td></tr>
</table>

Figure 4.4: Neighbor Discovery Response packet structure.

- *Token Owner ID (TO ID)* [2B]: this field contains the unique ID of the Token Owner.

- *Queue arrival time* [2B]: this field contains the time (again, seconds since the epoch) at which a vehicle stops.

- *Data length* [2B]: this field contains the length of the next field.

- *Carried information* [variable length]: this field contains the information carried by the vehicle.

As the waiting timer expires, TO selects the next TO basing on the received NDR packets, according to the first phase in Algorithm 3.

Whenever the TO does not receive any NDR, it re-triggers a new ND after some time. It is worth noticing that the time must be a multiple of the ND time duration and the TO must increase consequently the *Hop Counter* field in the token. If after two ND attempts, the TO does not receive any NDR, it triggers the token return to the TL.

Phase III: Next TO selection & token forwarding

The last phase is different if the token is in discovery or return phase. In general, the TO modifies some fields: it increases or decreases the *Hop Counter* field, it changes the *Next TO ID* field and adds or removes its unique ID to the *Visiting Nodes ID* field. In particular:

Algorithm 2 Max Available Hop selection

```
Max Hop = floor($AmTime$ / $ND\ time\ duration$)
repeat
    if $AmTime$ - ($Max\ Hop \times ND\ time\ duration$ + ($Max\ Hop$ - 1)$\times TX\ time\ duration$) then
        exit
    else
        $Max\ Hop \leftarrow Max\ Hop$ - 1
until $Max\ Hop$ is 0
Set $Max\ Hop$ in token packet
```

- If the L flag is set to 0 (i.e., discovery mode) the TO increases the *Hop Counter* field, appends its ID to the *Visiting Nodes* fields, puts the Next TO ID in the related field and, according to EC paradigm, inserts the Next TO data (carried information) by integrating the token context awareness, as explained in Sec. 4.1.2. It is worth noticing that the Next TO could be a previous visited node and, as explained before, the inserted data is 0.

- If the L flag is set to 1 (i.e., return mode) the TO decreases the *Hop Counter* field, removes its ID to the *Visiting Nodes* field, does not add any information in the *Data* field, and gets the first ID in *Visiting Nodes* field, while it modifies the previous value in *Next TO ID* field with this one.

After this operations, the TO sends the token, while setting an *AckTime*. Any node receiving the token analyses the *Next TO ID* field and if it is not the selected TO, discards the packet. The *Next TO*, instead, sends an *ACK* message: if it is not received before the timer expiring, the TO re-sends the token.

Next TO selection algorithm

The next TO is selected according to the TT Algorithm 3. There are two different Next TO selection Algorithms, depending on the path.

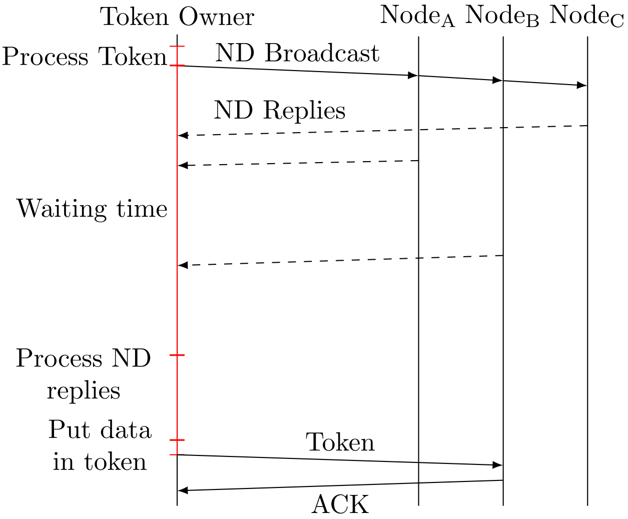


Figure 4.5: Token passing scheme for Discovering Phase.

If the token is in discovery mode (L field sets to `0`), after waiting timer expiration, the actual TO checks all the received NDR packets and, if it still has some time remaining (i.e., the *Hop Counter* is lower than *Max Available Hop*), it searches for the next TO.

It basically selects the neighbor that carries more information and, if it is not yet visited, it becomes the next TO. If the best neighbor has already been visited it is discarded. This process repeats until a Next TO is found or there are no available neighbors (i.e., all the neighbors have been discarded). In the latter case, the next TO becomes the vehicle last come in the queue, while in the former case, the next TO is a previously visited device and the added information is 0.

If a TO verifies that the *Hop Counter* becomes equal to the *Max Available Hop*, it triggers the Return Mode (changes to `1` the L field) and it becomes responsible to evaluate the return path. It analyses the *Visiting Nodes ID* field and performs an optimization: if loops are present (i.e., nodes are visited more than one time), they are deleted. Further it appends all the visiting node in a FIFO queue (the first ID becomes the Next TO) and it changes the *Hop Counter* (equal to the length of the modified *Visiting Nodes ID*). It can be noticed that henceforth the added information is 0. If a node receives the token in return mode, it removes its ID in the *Visiting Nodes ID* field, sets the following ID as the Next TO, decreases the *Hop Counter* field, and sends the token.

4.1.4 Numerical Results

The proposed TT protocol has been validated by performing numerical simulations on Python™ based framework integrating both `networkx` and `numpy` to set up and analyzing the network features. The performance in terms of the amount of collected information within a predefined time constraint, as expressed in eq. (4.3) are investigated; in addition, the probability of token recovering has been also derived. Specifically, the following algorithms are compared:

1. Drunkard Random Walk (*DrkRdnWlk*), i.e., a special case of the well known *random walk* with a finite time horizon, where TO randomly forwards the token packet to one of its neighbors. As a consequence, it is a lower bound for the performance.

2. Heuristic 1 (*Heu1*). In this approach, the next TO is selected as in Algorithm 3. The main difference with respect to the proposed TT scheme is that *Heu1* does not save the path and, when the return to the TL is triggered, the neighbors discovery procedure is still performed and the token packet is forwarded the best neighbor which becomes TO. The trigger is based on a time value τ which represents the amount of available time to explore the network normalized to the deadline. As soon as this threshold is exceeded, the token is triggered back. In performing our simulations τ is equal to 3/5, since it represents a good trade-off. This parameter is a priori set by the TL within the `PADDING` field of the token packet (Fig. 4.2). When a node receives the token at the current time t, it checks the elapsed time and if $t - TTE > AmTime \times \tau$, it changes the `L` field and sends the token to the next selected TO.

3. the proposed TT approach, previously characterized;

4. the CGP protocol already addressed;

5. the optimal solution, which is, however, affordable for a limited number of involved vehicles.

Algorithm 3 Tom Thumb Algorithm

```
if $L$ is 0 then
    if $Hop\ Counter$ is not $Max\ Hop$ then
        repeat
            finds(max($\epsilon_i$))
            if $i$ not in $VNsID$ then
                $NextTO \leftarrow i$
                $NextTO_{information} \leftarrow \epsilon_i$
                exit
            else
                remove $i$ from Neighbors
        until len(Neighbors) is not 0
        if len(Neighbor) is 0 then
            $NextTO \leftarrow$ last arrived
            $NextTO_{information} \leftarrow 0$
        adds its ID in $VNsID$
        $HC \leftarrow HC + 1$
    else
        $L \leftarrow 1$
        create $return\ path$
        changes $VNsID$ and $Hop\ Counter$
        $NextTO_{information} \leftarrow 0$
        $NextTO \leftarrow VNsID[0]$
else
    $NextTO_{information} \leftarrow 0$
    Remove ID from $VNsID$
    $NextTO \leftarrow VNsID[0]$
    $HC \leftarrow HC$ - 1
```

Parameter	*Scenario 1*	*Scenario 2*
number of cars (N_c)	15	120
number of lanes (N_l)	3	3
time deadline	3 [s]	10 [s]
token transmission time	0.1 [s]	0.1 [s]
neighbours discovery time (*ND*)	[0.3-0.7] [s]	[0.3-0.7] [s]
mean coverage radius	10 [m]	10 [m]

Table 4.1: Parameters adopted for Scenario 1 and 2.

Simulation campaign is performed according to a Monte Carlo approach, where 10^5 runs are repeated for each of the two investigated scenarios, whose parameters are listed in Table 4.1. Moreover, the information available at each vehicle is modeled as a normalized uniform random variable in the $(0-1)$ interval.

To address the behavior of the proposed approach, in Fig. 4.6 a preliminary snapshot of the token passing process is presented, by depicting the information gathered by the optimum, TT, *Heu1* and *DrkRdnWlk* approaches as a function of the round (i.e., hops) for $N_c = 24$ and $ND = 0.3$ [s]. The gathered information has been normalized to the maximum amount available in the scenario. It can be noticed that optimal values are closely approached by the proposed TT method which instead performs better than *DrkRdnWlk* and *Heu1* techniques.

In Fig.s 4.7a- 4.7b, the normalized collected information and the collection probabilities achieved by the different approaches are, respectively, sketched for the Scenario 1 and different ND timer duration, i.e., 0.3, 0.5 and 0.7 [s]. It can be preliminary pointed out that *DrkRdnWlk* protocol is the worst algorithm among the considered algorithms in terms of both collected information and collection probability. As a consequence, such that, whenever the token is returned back to the TL, the collected information is extremely low. Conversely, if it does not return to the TL within the time constraint, it is considered lost and the gathered information is null. Besides, *Heu1* and TT are able to return the token to TL, while respecting the time deadline. The percentage of gathered information depends on the ND time duration: the greater time spent for ND, the shorter path the token travel within the network and less information it is able to collect. Moreover, it can be pointed out that the proposed TT protocol is close enough to the optimum approach and always performing better than the other alternatives,

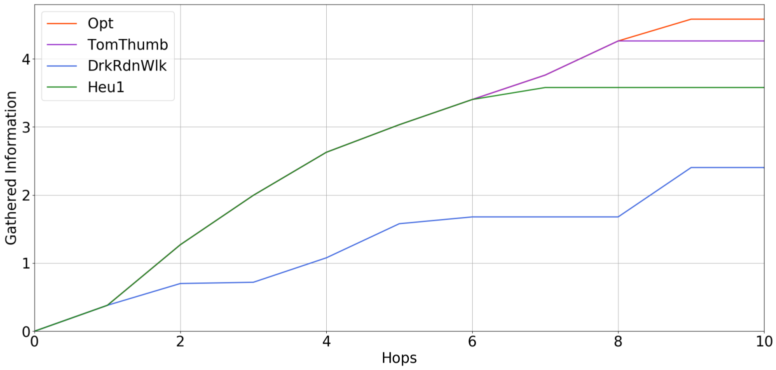


Figure 4.6: Normalized cumulative gathered information as a function of time round (i.e., hops) comparison among the proposed TT and alternative approaches for for $N_c = 24$ and $ND = 0.3$ [s].

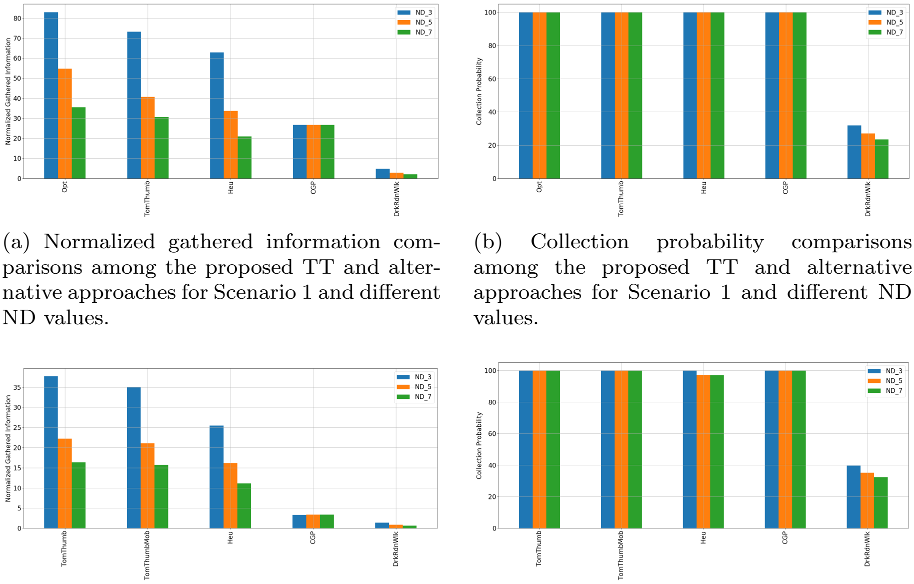
except for the case of high ND timer duration, where CGP achieve almost equivalent performance because of the limited number of vehicles .

Finally, the same investigation is repeated for Scenario 2, as reported in Fig.s 4.7c- 4.7d, where the higher number of vehicles makes the pursuit of the optimal path unaffordable. However, despite the performance degradation w.r.t. Scenario 1, TT performs better than all the other considered alternatives for any ND values. In addition, in the same in Figures, the presence of a low mobility pattern is considered. The road segment under investigation is affected by traffic congestion with an average vehicles speed equal to 50 [m/s]. It can be pointed out that the impact of mobility on TT is very limited with a relative performance decreasing of about 7%.

4.2 A cooperative spectrum sensing protocol for IEEE 802.15.4m wide-area WSNs

In this section, preliminary study about cognitive radio is disclosed. The following work [73] was presented in Paris at the IEEE International Conference on Communications (ICC) conference.

This work proposes a channel allocation scheme for LR-WPAN based on a cooperative centralized spectrum sensing approach. We refer here to



(a) Normalized gathered information comparisons among the proposed TT and alternative approaches for Scenario 1 and different ND values.

(b) Collection probability comparisons among the proposed TT and alternative approaches for Scenario 1 and different ND values.

(c) Normalized gathered information comparisons among the proposed TT and alternative approaches for Scenario 2 and different ND values in the presence also of a low mobility pattern.

(d) Collection probability comparisons among the proposed TT and alternative approaches for Scenario 2 and different ND values in the presence also of a low mobility pattern.

Figure 4.7: Overall results

a three-tiers LR-WPAN architecture, where we can distinguish the sensor devices level, the Child-coordinators level and, finally, the super LR-WPAN coordinator level. In the proposed scheme, channels allocation is performed by the LR-WPAN coordinator upon receiving the spectrum sensing reports from the Child-coordinators. The performance of the proposed scheme is evaluated in terms of resulting detection probability and throughput, in a scenario where Industrial, Scientific and Medical (ISM) bands are saturated and different LR-WPANs compete to the channel access. Performance comparisons with a classical scheme where the channel allocation is accomplished without performing spectrum sensing, are also presented in order to highlight the advantages of the proposed scheme.

4.2.1 Introduction

Nowadays, the paradigm of IoT [49] is evolving in order to allow connections, mainly wireless, of billions of devices and to support new advanced applications [58,68]. This trend boosted the introduction of a very dense and multi layered network architectures [50]. One of the more relevant issue focused here is represented by the channel access scheme, since a large portion of the electromagnetic spectrum is allocated to various wireless services (i.e., TV broadcast, military users, etc...). Therefore, it is straightforward to note that the widely adopted *static* frequency allocation schemes can not meet the need of an increasing number of end-user devices to be connected.

As a consequence, new methodologies have to be identified in order to allow a more efficient use of the available spectrum. Cognitive Radio (CR) [188] is widely considered as a possible solution to the spectral congestion problem. The main goal of the CR approach is to enable an opportunistic use of the frequency bands by unlicensed users, usually named Secondary Users (SUs), whenever they are not in use by licensed users, commonly referred as Primary Users (PUs). Hence, the adoption of CR technology by SUs entails the capability of performing the spectrum sensing in order to acquire awareness about the PUs spectrum usage.

Many factors (like multi-path fading and shadowing) can influence the reliability of the spectrum sensing outcome performed at the SU ends. Then, in order to guarantee a high spectrum sensing reliability, SUs could collaborate for sharing spectrum sensing information and taking more accurate decisions about the availability of spectrum resources according to a centralized decision principle [41]. Cooperation can be, also, a key to reduce devices costs: indeed, we can rely on low-cost and performance devices, since cooperative spectrum sensing achieves performances close to the best case. Another possible advantage of cooperative spectrum sensing is the sensing latency reduction, which in turn increases the useful transmission rate (and the system throughput) [129].

This preliminary work investigates the advantages of the cooperative sensing approach by focusing on a specific wireless network based on the sixth amendment of IEEE 802.15.4: IEEE 802.15.4m - TV White Space between 54 MHz and 862 MHz physical layer and proposes a solution tailored to this case study. In particular, on a centralized cooperative spectrum sensing scheme, in which SUs send their sensing information to a central unit, usually named Fusion Center (FC). The FC roles are to process sensing information and identify the spectrum holes through an hard decision. Since battery life of a wireless sensor device is a critical issue, in [186] the authors proposed a new MAC protocol to battery saving.

4.2.2 System Model

IEEE 802.15.4 is already discussed in Sec. 2.2.1. In the version under study (IEEE 802.15.4m [33]) the adopted band is extended toward the TV white space (TVWS), then additional PHY layers are added, but the channel access method remains unchanged.

In IEEE 802.15.4m a new network topology is added: TV White Space (TVWS) Multichannel Cluster Tree PAN (TMCTP). It is a cluster tree with a SuperPAN Coordinator (SPC) and subnets managed by coordinators called ChildPAN Coordinator (CPC). The communication among subnets is expected only by coordinators.

The superframe structure is unchanged (comparing to the base standard), only a new period is added. The new superframe is called *TVWS Multichannel Cluster Tree PAN superframe*. This new period, called *extended period*, is used to match the ChildPAN superframe structure to the SuperPAN Coordinator superframe (this is achieved by transmission beacon in the extended period). The superframe duration is described by MAC constant *macBeaconOrder*.

Also MAC frame is changed: the most significant difference is the increasing of Frame Check Sequence (FCS) field (from 2 to 4 byte).

We assume that our system is a TVWS Multichannel Cluster Tree PAN that consists of a SuperPAN Coordinator (SPC), N ChildPAN Coordinator

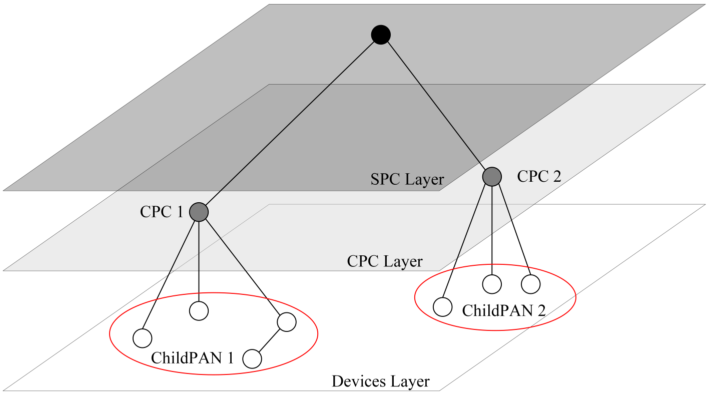


Figure 4.8: Example of TVWS Multichannel Cluster Tree PAN

(CPC), that can be seen as relay node as explained in [63], and K Reduced Function Device (RFD) for every ChildPAN. We also suppose that the unlicensed bands (e.g. ISM band at 2.4 GHz) are highly congested. The network uses a common channel, choosed by the SPC, to exchange messages. It is worth noting that this channel is also used by the SPC to communicate its PAN.

Since the network topology is a cluster tree, for our purpose the SPC represent the FC.

Each device has a single transceiver (half-duplex) which operates on the TV White Space band (54 MHz - 862 MHz) as well as the ISM band, but it can only use one channel at a time.

To know how are the TV White Space (i.e. the free frequencies), SPC could have a Internet connection to interrogate a geolocation database (GDB) to have a list of possible available channels in the geographic area in which is located. The SPC delegates CPC to coordinate devices in their PAN for spectrum sensing. This operation is done to figure out if there are external PANs that occupies the available channels and to verify the effective absence of PU (GDB could not be updated). We indicate the number of available channel in the TVWS band with L.

We also suppose that the PU activity follows an ON-OFF traffic: when it is ON, the PU transmits, when it is OFF the PU is idle and we can assume that the channel is free. We also suppose that the PU activity is slowly varying (e.g. likely within few days): so when PU is active, it occupies the channel for a long period (like TV transmission); when it is idle, the channel remains free for long period (occupation by external SU competitor excluded).

Energy Detection (ED) has been adopted as the detection method, due to its simplicity and interference detection rate, while it is not fair for the interfering nodes. If the output is greater than a predetermined threshold, the scanned channel is busy and it will be discarded.

4.2.3 Proposed Method

Analytical model

Spectrum sensing is formulated as statistic test between two hypothesis [118]:

$$\mathcal{H}_0 : \text{hypothesis no signal transmitted}$$
$$\mathcal{H}_1 : \text{hypothesis signal transmitted}$$

We introduce some hypotheses to simplify the analytic model: sensing time is less than channel coherence time, the noise is an Additive White Gaussian Noise (AWGN) process and the PU signal is a independent and identically distributed (iid) random process. Indicate with t_s the sensing time, we have that the sample are $N_s = \tau f_s$, with f_s the sampling frequency, we can write the statistic test for ED as [81]:

$$\mathit{ED(r)} = \frac{1}{N_s} \sum_{i=1}^{N_s} |r(i)|^2 \tag{4.4}$$

where $r(i)$ is the received signal at the SU transceiver.
Then, *ED(r)* is compared with a threshold λ: if it is greater than λ we choose the hypothesis $\mathcal{H}_1$, $\mathcal{H}_0$ otherwise.

The performance of a cognitive radio is derived basing on two parameters: the false alarm probability (p_{fa}) that is the probability that a CR evaluates the presence of a PU when it is absent, and the detection probability (p_d), i.e., the probability that a CR correctly evaluates the presence of a PU. Its opposite is the missed detection probability (p_{md}) that is the probability that a CR evaluate the absence of a PU when it is present. This probability has an important role in our system: from the PU side, highest p_d more protection it receive; from the CR side lower p_{fa} more possibility has to use spectrum holes [121].

If we have high number of samples ($N_s \to \infty$), by resorting to the central limit theorem, p_d and p_{fa} can be expressed as [121]:

$$p_d = Q\left(\left(\frac{\lambda}{\sigma_n^2(SNR+1)} - 1\right)\sqrt{\frac{N_s}{2}}\right) \tag{4.5}$$

$$p_{fa} = Q\left(\left(\frac{\lambda}{\sigma_n^2 - 1}\right)\sqrt{\frac{N_s}{2}}\right) \tag{4.6}$$

To limit detection uncertainty, more CR can cooperate, which is usually referred as cooperative sensing [118]. In our protocol we consider a *centralized* cooperation and a *hard* combining [188] in other words a combining in which devices transmit to the fusion center (FC) a bit: `1` if $ED(r) \geq \lambda$, `0` otherwise. We adopted a hard combining due to its simplicity. In particular, we employ the n-Out-Of-K fusion rule: if n of the K devices transmit bit `1` the FC assumes the presence of a PU, otherwise decides the absence of the PU. It is worth of noting that, if $n = 1$ n-Out-Of-K collapse in a OR rule, if $n = K$ n-Out-Of-K collapse in an AND rule [118].

If we assume that all the devices have the same p_d and p_{fa}, we can write the total (of the cooperative group) probability of false alarm, detection and missed detection, respectively, as:

$$\begin{aligned} Q_{fa} =& \ p\{\mathcal{H}_1|\mathcal{H}_0\} = \sum_{i=n}^{K} \binom{K}{i} p_{fa}^i (1-p_{fa})^{K-i} \\ Q_d =& \ p\{\mathcal{H}_1|\mathcal{H}_1\} = \sum_{i=n}^{K} \binom{K}{i} p_d^i (1-p_d)^{K-i} \\ Q_{md} =& \ p\{\mathcal{H}_0|\mathcal{H}_1\} = 1 - p\{\mathcal{H}_1|\mathcal{H}_1\} = 1 - Q_d \end{aligned} \tag{4.7}$$

The optimum value of n is related to the performance of the single device p_d and p_{fa}, that depends on λ, the threshold.

According to [195], if p_{md} is much greater than p_{fa}, $n \to 1$ and the optimal rule is an OR selection (i.e. it is necessary only few devices concordant), while if $p_{fa} \approx p_{md}$, n optimum is equal to $\frac{K}{2}$. Finally, if $p_{fa} \gg p_{md}$ the best policy is an AND choice (i.e., $n \to K$ and it is necessary have almost all devices concordant) because it is more likely to have a false alarm when the PU signal is absents than a missed detection when the PU is presents.

To find the n optimum we modified the objective function in [195] with the difference that the *total error rate* is expressed as a weighted summation:

$$\epsilon \doteq \beta \cdot Q_{fa} + (1-\beta) \cdot Q_{md} \tag{4.8}$$

The weight β $(0 < \beta < 1)$ for the total false alarm probability Q_{fa} missed detection probability (Q_{md}) allows to select which is more relevant for the reference application scenario. In addition, we characterized our optimization problem with an additional constraint on Q_{fa} (denoted Q_{fa^*}), as:

$$\begin{aligned} n_{opt} =& \arg\min_{n} \{\beta \cdot Q_{fa} + (1-\beta) Q_{md}\} \\ \text{s.t.}: &\quad Q_{fa} \leq Q^*_{fa} \quad \forall \; SNR \end{aligned} \tag{4.9}$$

The presence of the constraint Q_{fa^*} is motivated by the fact that the system throughput is more affected by Q_d.

Proposed protocol

Our proposed protocol introduces centralize cooperative spectrum sensing in the sixth amendment of IEEE 802.15.4. It helps the SuperPAN coordinator to choose a channel not interfered: in other words, to find a channel not occupied by PUs or additional competitor SUs.

Our proposed protocol consists in four phases: in the phase I, the CPCs compete against each other to get access to the channel; in the phase II, SPC waits for the CPCs report about spectrum sensing and, after all reports are received, processes the data for assignment the channels to the individual ChildPAN; in the phase III the SPC forwards its decision about the channel assignment to the CPCs and in the phase IV the CPCs communicate in their respective PAN the new channel (i.e. the initial frequency).

Phase I : The CPCs listen to the channel, waiting for an enhanced beacon (eBeacon). When eBeacon is received, each CPC attempts to get access to the channel with the CSMA/CA method. When a CPC succeeds, it sends a Dedicated Beacon Slot (DBS) to the SPC and it's been waiting for the following beacon frame that will have pending data. The CPC sends a data request to receive the DBS response (the access to the channel always occur with CSMA/CA method). When a CPC obtains the dedicated beacon slot, it waits for its DBS in the Beacon Only Period (BOP). When the DBS arrives, the CPC sends a Beacon in its PAN with fields request for sensing and GTS allocation enabled. As a device receives the beacon, it starts spectrum sensing. After serving each CPC, the SPC enters an inactive status.

Spectrum sensing lasts as long as a contention access period (CAP). After spectrum sensing is completed, the devices send a sensing response packet to the CPC with their individual results. Afterwards the device enters a waiting status. At the same time CPC create a matrix ($\mathcal{A}$) where the rows represent the devices in the ChildPAN and the columns are the channels where the devices implement spectrum sensing. The $a_{i,j}$ element corresponds to the decision of $i-$th device on $j-$th channel:

$$\mathcal{A} = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,L-1} & a_{1,L} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,L-1} & a_{2,L} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{K,1} & a_{K,2} & \cdots & a_{K,L-1} & a_{K,L} \end{vmatrix}$$

For every channel, the CPC choose $\mathcal{H}_1$ hypothesis if n device have forwarded bit `1`, $\mathcal{H}_0$ otherwise. At the end we have a vector where every $b_{1,j}$ element correspond to the j-th channel choice:

$$\mathcal{B} = |b_{1,1}\ b_{1,2}\ \cdots\ b_{1,L-1}\ b_{1,L}|$$

With the end of the second superframe, the first phase ending. The related time diagram is represented in Fig. 4.9a.

Phase II: In this phase every CPC in the cluster tree sends a sensing response to the SPC. This phase starts with a eBeacon whit GTS allocation transmitted by the SPC. During the CAP every CPC is in a wait state; when contention free period (CFP) begins, every CPC transmits, in its dedicated GTS, a sensing response packet (containing the vector $\mathcal{B}$) and waits an ACK from SPC. At the end of the active portion the SPC will have a matrix $\mathcal{C}$ where $c_{i,j}$ element represents the choice of the i-th Child PAN Coordinator over the j-th channel. This new matrix will have dimension $[N \times L]$:

$$\mathcal{C} = \begin{vmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,L-1} & c_{1,L} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,L-1} & c_{2,L} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,L-1} & c_{N,L} \end{vmatrix}$$

After all CPC have transmitted, SPC enters an inactive status. This time is used by the scheduler to process the matrix for assign a channel to every Child PAN. In [139] a scheduler technique used by SPC is explained. The overall time diagram is represented in Fig. 4.9b.

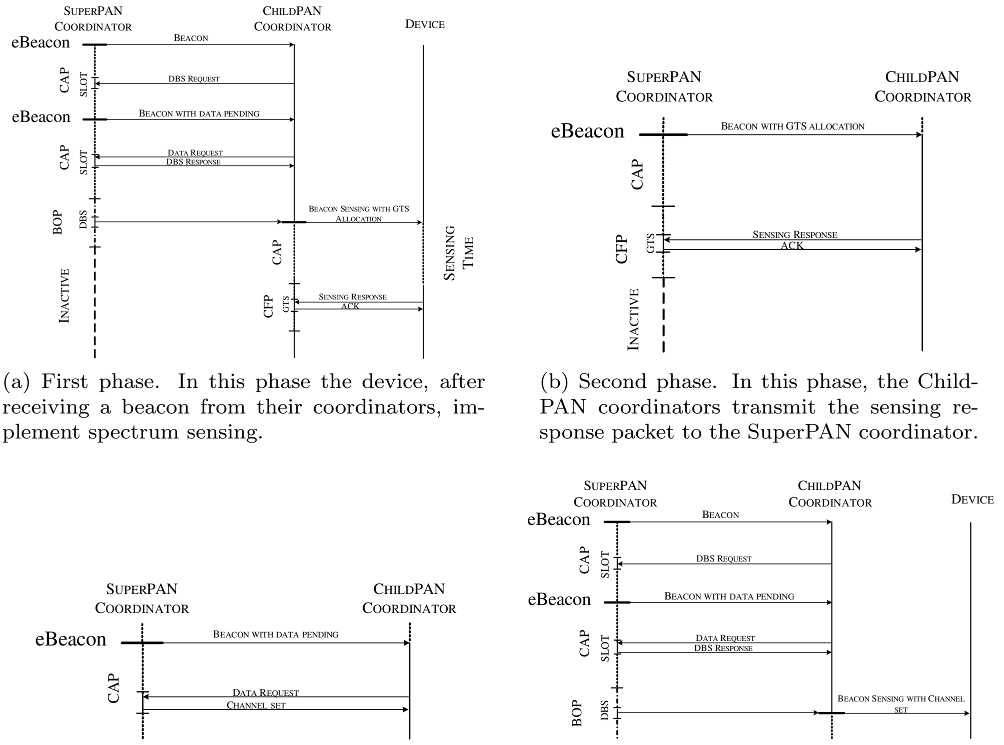
Phase III: This phase starts with the eBeacon transmitted by the SPC whit the data pending flag activate. Every CPC attempts access to the channel, when succeed, the SPC sends the channel set packet: this data packet reports the initial frequency assigned to a Child PAN. Knowing the operative band, every device can obtain the channel page and the final frequency. The related time diagram is represented in Fig. 4.9c.

Phase IV: The last phase of our protocol consist in the report to the device of a Child PAN the new channel assigned by th SPC. It is divided in two sub-phases: in the first one the CPC tries to obtain a DBS, in the second one the CPC communicates in this PAN the channel choice. The first sub-phase is the same explained in the first phase: the CPC listens to the channel, waiting for an enhanced beacon (eBeacon). As soon as it is received, it attempts to access to the channel, if succeeds, it sends a Dedicated Beacon Slot (DBS) to the SPC and waits for the following beacon frame for its DBS. When a CPC obtains the dedicated beacon slot, it waits for its DBS in the Beacon Only Period (BOP). When the DBS is received, the CPC sends a Beacon to its PAN with the information regarding the new channel within the Beacon Payload. The time diagram of this phase is represented in Fig. 4.9d.

4.2.4 Numerical Results

To evaluate the system performance we pointed out the advantages provided by the proposed cooperative spectrum sensing protocol in terms of detection probability and throughput for a realistic scenario.

As previously characterized, the cooperative spectrum sensing performance depends on the n parameter selection, which can improve the single node performance, i.e., p_d and p_{fa}. It is worth pointing out that our approach is inherently hierarchical, since it involve the ChildPAN coordinators, whose detected channels lists are collected and harmonized by the SuperPAN coordinator. In order to lower the complexity of the proposed scheme, we focus on a basic ED spectrum sensing, whose performance depends on the threshold λ. To this purpose, the IEEE 802.15.4 standard indicates a *minimum* receiver sensibility, i.e.,the lowest input power for which the Packet Error Rate (PER) condition are met [31], that is mandatory in order to operate in the TV White Space. This value depends on the physical layer and, in particular, if FEC is used: for instance, if we use the TVWS-FSK Mode 5, without FEC, according to Section 18.1.5.7 in [32], we obtain a receiver sen-



(a) First phase. In this phase the device, after receiving a beacon from their coordinators, implement spectrum sensing.

(b) Second phase. In this phase, the ChildPAN coordinators transmit the sensing response packet to the SuperPAN coordinator.

(c) Third phase. In this phase, the SuperPAN Coordinator sends to every ChildPAN coordinator its allocated channel.

(d) Fourth phase. In this phase, the ChildPAN Coordinator sends to every device in its PAN the allocated channel by the SuperPAN coordinator.

Figure 4.9: Overall phases

sibility equal to -83.2 dBm. To accomplish all the possible cases we chose a lower bound value for the sensibility equal to -120 dBm. In addition we limit to 7 the number of involved ChildPAN coordinators to accomplish the sensing reports within a single CFP and, then, avoiding the random channel access and reducing the protocol latency in Phase I (Fig. 4.9a) and Phase III (Fig. 4.9c).

The simulations are performed using MATLAB$^{\mathrm{TM}}$ framework, where accordingly we assumed $K = 7$ and number of sampling N_s equal to 128. According to PHY level used, we further assumed a bandwidth B equal to 600 KHz, then the noise variance σ_n^2 is equal to -116 dBm. So the sensing time is equal to $t_s = 106.6$ μs with a sampling frequency equal to $f_s = 2 \cdot B = 1.2$ MHz. Finally, the false alarm probability has been set to 10^{-3}.

In Fig. 4.10, the effect of n on the cooperative detection is pointed out for different SNR values. In particular, we evaluate the detection probability Q_d according to (4.7) and derived the optimum n value (4.9), also presenting the basic performance p_d, all as a function of SNR.

It can be highlighted that the best policy for the *unconstrained* optimization problem is the 1-OutOf-7 rule, which maximizes the Q_d for all SNR values. However, in this case Q_{fa} could be higher than the 10^{-3} value, so that the effective optimal policy might be derived matching this constraint (4.9). As a result, the selected policy is represented by 2-OutOf-7 rule ($n = 2$), whose Q_d is close to the best case and $Q_{fa} \leq 10^{-3}$. It is finally evident that $n = 2$ allows to achieves a remarkable performance gain in terms of detection probability especially for low-to-medium SNR range.

To evaluate the system throughput, we focus a beacon enabled slotted single ChildPAN scenario, where a specific channel has been allocated by the SuperPAN Coordinator, once the cooperative sensing has been accomplished. We resort here to the analytic framework proposed in [70], which is based on the *busy* channel probability p, and the probability τ that a node is transmitting on a generic slot. We evaluate three cases: (i) *ideal* case where no PU is active; (ii) *random* choice, where the PU is active and the channel assignment is randomly performed; (iii) *sensing based* channels selection case, where the PU is active and the channel assignment is implemented by spectrum sensing.

According to [70], it is possible to express the equilibrium equations for

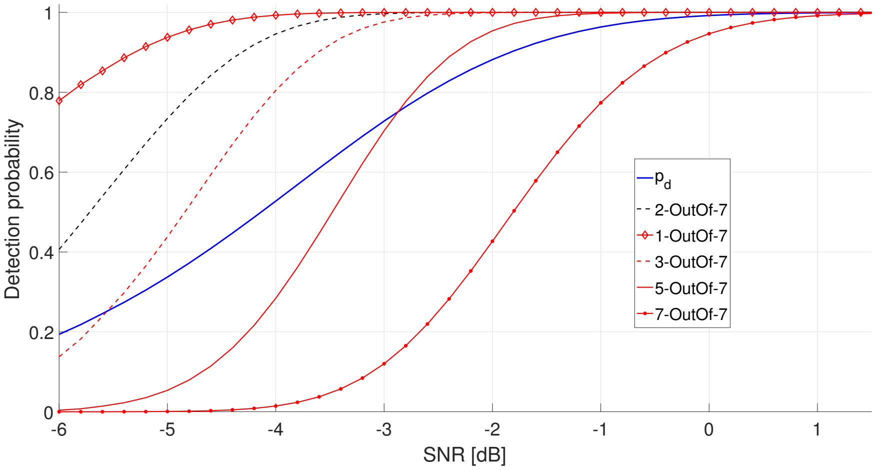


Figure 4.10: Detection probability (Q_d and p_d) comparisons for different values of n as a function of SNR.

p in the case of cooperative sensing technique as:

$$\begin{aligned}
p &= 1-(1-\tau)^{N-1} && \textit{Ideal} \\
p &= p_a + (1-p_a)\cdot\left[1-(1-\tau)^{N-1}\right] && \textit{Random} \\
p &= p_a\cdot(1-Q_d)+ \\
&\quad +\left[1-p_a\cdot(1-Q_d)\right]\cdot\left[1-(1-\tau)^{N-1}\right] && \textit{Sensing}
\end{aligned} \tag{4.10}$$

where p_a represents the PU activity factor and N the number of SU in the ChildPAN under consideration. In order to evaluate the protocol performance we assume $p_d = 0.53$ (corresponding to SNR $= -4$ dB) and $p_d = 0.73$ obtained having SNR $= -3$ dB. In particular, for the two cases under evaluation, we achieving approximately $Q_d = 0.953$ and $Q_d = 0.998$ respectively with our sub-optimal choice, a 2-OutOf-7. In Fig. 4.11, the overall normalized throughput is evaluated as a function of p_a. We can firstly notice that the upper bound is represented by the ideal case, while the lower bound is given by the random channel selection. In addition, it is worth pointing out that our cooperative protocol achieves performance similar to the best case, once the optimal (or sub-optimal) value of n have been chosen.

In this work, we proposed a novel spectrum sensing cooperative scheme

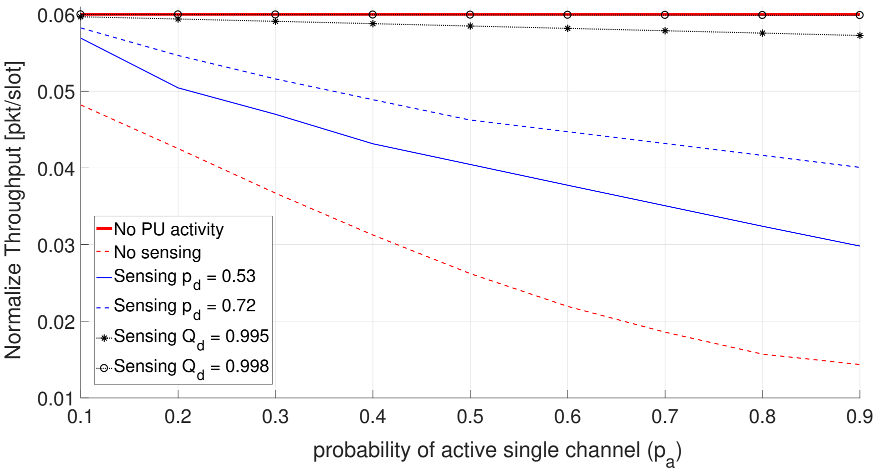


Figure 4.11: Throughput comparison for three different policies.

and a protocol designed for IEEE 802.15.4m networks. We focus on a generic system with a fixed number of devices (ChildPAN coordinator and reduced function devices), saturated ISM bands and a on-off traffic model for PUs in the licensed bands. Our aim is to effectively detect an idle PU channel within TVWS, and allocate it to a childPAN in order to achieve the best throughput.

Due to its simplicity and low computational cost, we adopted an ED detector for each device upon which the cooperative spectrum sensing is performed.

Additionally we derived and accurate an analytical model leading to a constrained optimization problem to select the number of CRs necessary to achieve the best detection.

By means of numerical simulations, we shown that the proposed approach is always able to maximize the detection probability and the system throughput especially for low-to-medium SNR value and low cost devices.

Chapter 5

System Security

Anyone who thinks that security products
alone offer true security is settling for
the illusion of security – Kevin D. Mitnick

In this section the classical and IoT security approaches are presented. The National Institute Of Standard and Technology (NIST) define the computer security as: *the protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).* In this definition three keys are defined:

- *Confidentiality.* A shared secret has to be kept between the involved persons. This is obtained by using encryption. Standard as IEEE 802.15.4 provides a cryptographic suite based on Advanced Encryption System (AES) without proper key management mechanisms.

- *Integrity.* Data must be remain the same from sender to receiver. Hash functions are used to reach data integrity.

- *Availability.* The system must be always works. Service availability is the most difficult to guarantee and it is obtained only by an accurate network planning.

When one of these keys is violated, the entire security of the network has been compromised.

5.1 Security in Classical and IoT Networks

It is worth noticing that companies spend time and money on defense against attacks, and when at least one succeeds, also in mitigation.

Depending on the incoming *type* of attack, there might be a particular countermeasure that could be performed. In this sense, there's no general-purpose approach. However, there are general-purpose approaches (and tools) that are useful in detecting if an attack is being attempted. Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) are network appliances put in place to catch "malicious" activities. The IDS objective is to *detect* any dubious network activity. It is a passive tool: when a suspicious event occurs, it react by sending a trigger to the network administrators. Differently, an IPS is an active tool [147] aimed at preventing the spread of potential malicious activities by triggering a quick response. Usually, IDSs analyze network traffic patterns, packets content or systems logs, searching for the evidence of security violations mainly at the network layer (e.g., for routing attacks). Sadly, a large class of attacks targeting the IoT data can not be easily detected by traditional IDS systems.

A *Firewall* is a software that monitors, controls and, allows incoming and outgoing network traffic. The firewall divides the network in *security zones* according to their intended security requirements. An example is an enterprise/private (i.e., Local Area Network (LAN)) network and Internet [166].

One of the most dangerous type of attack, due to its dramatic impact on the victim, is the so-called Denial of Service (DoS). The DoS main objective is to remove the availability of a service. The attack could hit the resource itself or the infrastructure. The simplest way to perform a DoS attack is to flood the victim (or its infrastructure) with "junk" traffic in order to consume all its available resources. A Distributed Denial of Service (DDoS) attack is similar to DoS but the attack implementation is quite different. DoS attack is launched from a single node while DDoS attack is launched from a *distributed* group of nodes. Fig. 5.1 show this difference.

Traditionally defense strategies against DoS-DDoS attacks are using a reverse proxy, enabling router throttling, and/or try to absorbing the attack. Whatever countermeasures network administrators adopt, they require a lot of (computational and financial) resources. For this reasons, common countermeasures are not suitable for the IoT world.

Sadly, even the right countermeasure is not always enough. In 21 Octo-

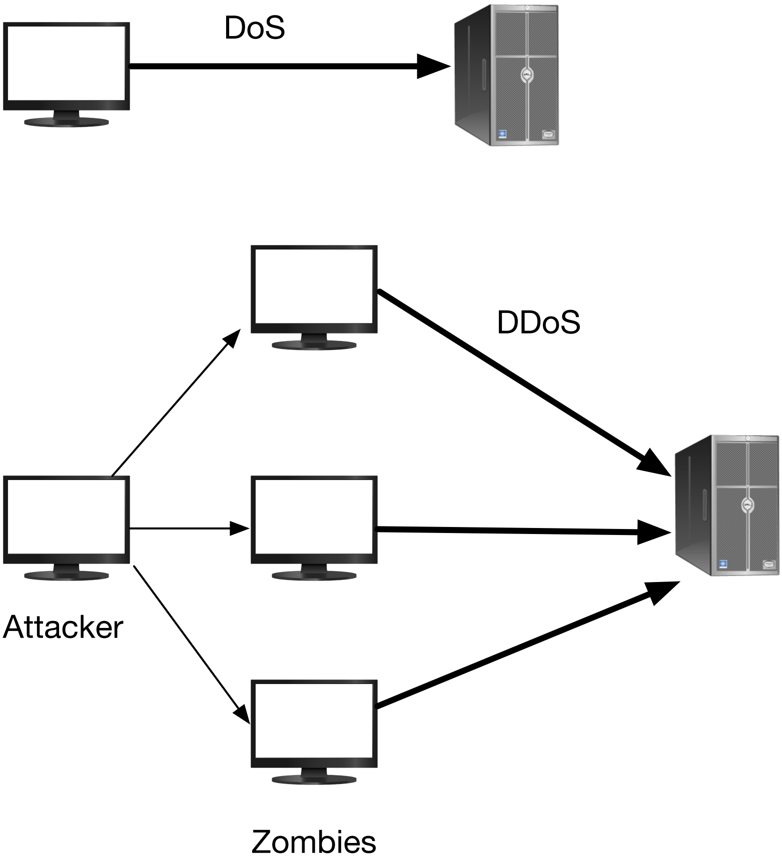


Figure 5.1: DoS versus DDoS attack.

ber 2016 3 different DDoS attacks[1] were performed against a Domain Name System (DNS) provider: Dyn. The attack goal was to make services unavailable for users in North America and in few countries of Europe (as shown in Figure 5.2a). The attack was accomplished through numerous DNS lookup (that was produced peak of 1.2 Tbps) requests coming from approximately fifty thousand of IP addresses (as shown in Figure 5.2b). Companies that offer Internet services are daily targeted by DDoS attacks, the fact that Dyn has not been able to mitigate the attack suggests how was widespread the problem.

It is easy to understand that this attack had produced significant economic losses for Dyn... since February 2017 it has lost the management of 14500 domains (more or less the 8% of the overall).

The responsible of this attack was the *Mirai* malware [109] that created a botnet of over ten million of IoT devices such as CCTV cameras, IP printer, and home routers. Mirai exploited a lackluster security practices: do not change the default passwords. An infected device performs a rapid scan on Telnet port (23 or 2323) and if a device responds, the infected ones tries to take control using one of the 68 factory default username and passwords to login[2]. One of the interesting things in the Mirai malware was the *"Don't Mess With" List*: an IP addresses blacklist used to avoid particular (such as US Postal Service, the Department of Defense, IANA, etc...) addresses during the scans. Due to the fact that Mirai was loaded into RAM, the virus remained active until the infected devices was rebooted. The Mirai programmers released the code as open source resource and it was used to create new (and dangerous) malwares.

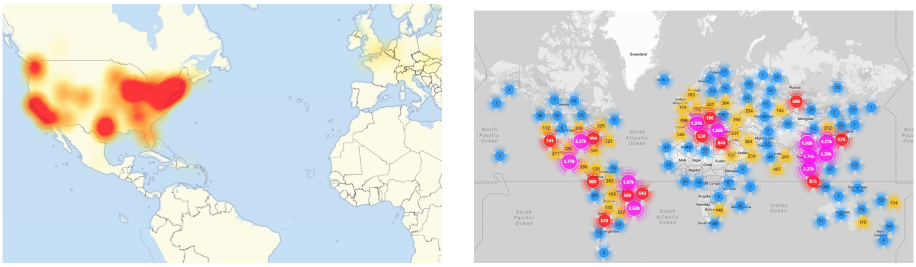
Technologies alone do not supply security: it is also important where the single network element is placed. The right choice is based on the Defense in Depth idea.

5.1.1 Defense in Depth and Whack-A-Mole Approaches

Defense in depth is a military idea and anyone who has played Age of Empire (without cheats) has applied this concept in the construction of his city. At least one city center was protected by several walls. Linking to this example, defense in depth is a layered approach to the security (and network design).

[1] The three attacks were perform at 12:10, 16:50, and 21:00 UTC.

[2] This is a brute force technique for guessing passwords, a.k.a. dictionary attack.



(a) Map of areas most affected by the second attack [15].

(b) Mirai botnet [9].

Figure 5.2: Map of most affected areas during the second attack and map of infected devices responsible for the attacks.

With this approach, networks appear sliced with multiple (security) checkpoint to control access between the levels.

The defense in depth goal is not to prevent an attack altogether (which is impossible in general). Instead, the goal is to delay the attacker with traps [130]. The delay is necessary to detect the attack and perform the right countermeasures before the attacker reaches its objective. With this approach, an attacker would traverse multiple layers in the network and every layer could have different detection and/or prevention points. Clearly, if all the wall falls down, the attacker wins.

A common defense in depth approach is represented by the so-called *Demilitarized Zone* (DMZ): a DMZ is an isolated network portion where "untrusted"[3] servers (web, emails, ftp and DNS) are placed. The DMZ servers could be communicate with the local network hots through one or more firewalls.

The term Whack-A-Mole comes from a famous game where a player obtains points by using a mallet to whack mechanical mole when it pop out of its holes, and is a good similarity to understand the actual security world: the mechanical moles are the daily threat and the player is the IT department. And as in the game, the moment when you miss the mole will come, causing the increase of the mole exit speed, that will leading to another miss... In the game it is not important but in IoT world, and more in general in the cyber-world, a little miss could be lead to huge damages.

[3] In this portion the hosts most vulnerable to attack are placed in order to protect the rest of the (local) network if any of them become compromised.

Nowadays, Whack-A-Mole approach is not a good choice. Cyber-security must be proactive, in order to anticipate the problems and, when they happen, have a (viable) solution(s).

5.1.2 Privacy Concerns

This subsection starts with a story. On December 6, 2016 American associations disclosed a compliant and request for investigation for two of the firsts "smart" toys: My Friend Cayla e I-Que Intelligent Robot, product by Genesis Toys [7]. The associations accusation was that these toys, through their microphone, recorded and collected child conversations. Moreover the relative apps, once installed, has access to: memory, microphone, Wi-Fi, and Bluetooth connections of the smartphone. On the compliant, the associations wrote that *by purpose and design, these toys record and collect the private conversations of young children without any limitations on collection, use, or disclosure of this personal information. [...]. Cayla's Privacy Policy does not mention speech data or describe the collection, use, or disclosure of such data* [19]. After this compliant, some European States banned the sale of these toys as they constituted a concealed espionage device violating the kids and adults privacy.

This is one of the many stories related to the violation of privacy by smart devices.

When we talk about privacy concerns in IoT, we must be consider three factors:

- The competition between manufacturer to become the first in the market with their products.
- The huge presence of smart devices in everyday life.
- The consumers unawareness. An user wants that the bought IoT device works and does not care where his personal data are saved or change default password(s) to enhance security.

With *wikileaks* (in 2006) and *Snowden* (in 2013) scandals privacy protection had become a topic of discussion between cyber-security and politics experts. In Europe this has leaded to discuss about new regulation. In particular, on April 14, 2016 European Parliament and Council of the European Union have issued the General Data Protection Regulation (GDPR). The main objective was to enhance the protection of personal data of European citizens, unifying the privacy legislation within the European Union (EU). It is worth noticing that this regulation is effective also for non-European companies handling data from residents of EU.

In IoT, the lack of privacy is also given by the un-used of strong cryptography algorithms due to the battery lifetime and hardware constraints.

5.2 “Network Sentiment” Framework to Improve Security and Privacy for Smart Home

In this section the work [152] is presented. This work has been published on MDPI - Future Internet.

A Smart Home is characterized by the presence of a huge number of small, low power devices, along with more classical devices. According to the IoT paradigm, all of them are expected to be always connected to the Internet in order to provide enhanced services. In this scenario, an attacker can undermine both the network security and the user’s security/privacy. Traditional security measures are not sufficient, because they are too difficult to setup and are either too weak to effectively protect the user or too limiting for the new services effectiveness. This work suggests to dynamically adapt the security level of the smart home network according to the user perceived risk level what we have called *network sentiment* analysis. The security level is not fixed, established by a central system (usually by the Internet Service Provider) but can be changed with the users cooperation. The security of the smart home network is improved by a distributed firewalling and IDS both to the smart home side as to the Internet Service Provider side. These two parts must cooperate and integrate their actions for reacting dynamically to new and ongoing threats. Moreover, the level of *network sentiment* detected can be propagate to nearby home networks (e.g. the smart home networks of the apartments inside a building) to increase/decrease their level of security, thus creating a true in-line IPS. The paper also presents a test bed for Smart Home to detect and counteract to different attacks against the IoT sensors, Wi-Fi and Ethernet connections.

5.2.1 Introduction

In a Smart Home environment several specific home automation devices, (e.g., temperature monitoring sensors, air quality devices, infotainment system, Smart TVs, fire and/or gas detectors, etc.) might need to communicate with the external networks (e.g., smoke detection alarm) and receive commands to perform various actions (e.g., increase the temperature in the house or remotely monitor with surveillance cameras unattended rooms or child's rooms).

The more the home becomes smarter, the more the problem of cyber security becomes important. As a matter of fact, a number of recent attacks have been performed by exploiting vulnerabilities of small devices (e.g., My Friend Cayla, a famous toy [13] attacked for configuration mistakes, i.e., default password unchanged), and by using this high number of devices as sources for DDoS attacks (see for example [89] or the malware Mirai in the 2016 [109], [47]). Moreover, malicious attacks may bring a significant impact not only to the network security but also to the safety of the user. For example, by using Internet device-scanning search engines such as Shodan (`https://www.shodan.io`), it is possible to obtain a list of home surveillance cameras with their IP addresses, geographic locations, etc. [125]. A burglar can control when the IP webcam is more frequently accessed and consequently can understand if the house owner is away from the house. Unfortunately, the obvious security *solutions* are not really feasible:

i) upgrading the software of the vulnerable appliances is often impossible (the end-users have often limited capabilities),

ii) substituting them with more secure devices is not a viable solution (too expensive for Home scenario), and

iii) blocking their traffic preemptively might prevent their functionality altogether.

Moreover, classical security approaches (i.e., the use of restrictive firewall rules, strong authentication system to access the network, e.g., IEEE 802.1X) are designed for those attacks which are not in the interior network while the IoT devices are vulnerable to intrusion both from Internet than from wireless attacks originated *from within* the smart home network.

To make effective the connections of resource-constrained IoT devices the 6LoWPAN are standardized [92], empowered by protocols such as 6LoWPAN adaptation layer [108], Routing Protocol for LLN [183] and the CoAP [161], [54]. The 6LoWPAN network uses compressed IPv6 protocol for networking and IEEE 802.15.4 as data-link and physical layers protocol. Each layer in 6LoW-

PAN can be vulnerable to security threats and, unfortunately, standard preventive security mechanisms, such as cryptography and authentication, can not detect all possible attacks like insider attacks (e.g. routing attacks) or a guy which uses a legal key but has a malicious behavior.
As consequence, IDSs specifically designed for IoT are necessary as a second line of defense to provide more security awareness and to add some dynamic threat protection functionalities to a network.
All these actions are not easy for a normal user, and they limit the network usability. For these reasons the network security is often neglected in many domestic networks, and even in some enterprise ones.

If the network itself must adaptively react to new threats, increasing the security measures when there is an effective, ongoing, vulnerability exploitation, and relaxing the rules when there is no real threat. In the following, the dynamic threat evaluation is called as *network sentiment analysis.*

With respect to this, the work shows an architecture, called SHIELD, with a distributed firewalling and threat analysis system. One part of the security infrastructure must be as close as possible to the user, potentially at the user's premises, and another part in the Internet Service Provider (ISP) network.

The contribution over the state of the art is about the way these elements should be integrated. In the past the firewalls acted as separate entities. All the user firewalls and IDSs should be part of an integrated ecosystem, reacting dynamically to new and ongoing threats. In this vision, the whole system should be orchestrated by a *coordinator* hosted by the ISP (or by any secure and trusted provider), which is responsible for evaluating the risk measure of each user and of the ISPs as a whole. In this vision, if the *network sentiment* level of a smart home network is increased due to the risk of attack, this information can be propagated to the nearby smart home networks (e.g. in the different apartments in the same building) which can take counteractions automatically establishing a real time Intrusion Prevention System (IPS).
The *network sentiment* approach will:

- Ease the network security setup,
- Make it more reactive toward incoming threats,
- Keep the number of security rules to the minimum necessary to guarantee an adequate security and

- Increase the security level in networks geographically close or with similar characteristics in terms of firmware of devices, connections and applications.

Moreover, the work also presents a testbed able to detect and react against attacks on Ethernet, Wi-Fi connections and IoT protocols. The testbed described in the paper has the goal to outline that the proposed SHIELD architecture is feasible. Toward this end, the hardware used in the testbed reflects as much as possible a normal Linux-based CPE. As a consequence, we expect that implementing the SHIELD functionalities on a commercial CPE will be very easy.

5.2.2 Related works

In a IoT smart home scenario many small, low power, low computation devices are used in the segment of home automation to improve the quality of services and, as consequence, the quality of experience offered to the users [154]. Different aspects for preserving privacy can be found in [106] while several papers in literature examine the security challenges and threats suited for smart homes, as in [92,115], where a survey of existing protocols for secure communications on IoT can be found, together with open challenges and research issues in this area. In [164] a protocol is proposed to secure route optimization and handover management, which uses trust between Proxy mobile IPv6 (PMIPv6) domain and smart home to ensure security as well as performance over the path between mobile nodes and home IoT devices.

In [163], the authors propose a secure scheme for data uploading on Cloud to guarantee the integrity of the data with a session key generation assisted by the home gateway.

In [125], a review of existing network techniques for enhancing IoT security is provided together with future key technologies for trusted Smart Home systems such as system auto-configuration and security update. A gateway architecture is chosen as the most appropriate for resource-constrained devices and for high system availability.

In [57] the authors propose a cross layer method to overcome the traceability of the user in smart home networks 6LoWPAN. In this case, the IEEE 802.15.4 standards [34] foresees to cipher the payload but leaves out the headers containing the Layer 2 addresses of the source and destination

of the packets. In the proposed method all the nodes change periodically all their addresses, both at Layer 2 and 3, to secure of both the IEEE 802.15.4 and 6LoWPAN protocols.
However traditional ICT standard security solutions, such as cryptography and authentication techniques, do not prevent all possible attacks and are not tailored for smart home environment due to the resource-constrained IoT devices, to their heterogeneous interaction and to their different policy and connectivity domains.

As consequence, IDSs are required to detect intruders and malicious activities to threaten the network. IDSs can be classified in signature-based or behavior-based. Signature-based IDSs use pattern-matching techniques to detect an attack, while behavior-based IDSs analyze the node behavior to detect anomalies. The first type is best suited for known attack and the second one for unknown attacks. Hybrid detection technique combining signature and anomaly based approaches can improve the efficacy of IDS.

Furthermore, IDSs can be classified in Host, Network or Distributed. Host IDSs only process the data of a single node, Network IDSs are able to monitor a network (typically one or more links), and Distributed IDSs are able to process the data of multiple, independent probes and/or multiple federated IDSs. Several IDSs are designed for WSN, a general survey on IDS and IPS can be found in [88]. However, these IDSs are not directly applicable for IoT because IoT devices are: globally accessible, resource-constrained, heterogeneous, and adopt new custom protocols such as CoAP, RPL. A survey of more IoT-oriented IDSs can be found in [55, 91, 102, 191].

Most of these IDSs focus only on threats at network layer. Examples of real-time IDS for IoT can be *SVELTE* [155] which meets the requirements of IPv6-connected IoT devices and detects routing attacks such as sink-hole, selective forwarding, and spoofed or altered routing information; Complex Event-processing (CEP) [77] which detects events in real-time by analyzing the stream of information; the IDS by [194] which targets DoS attacks on RPL.

In [93] an intrusion detection and prevention framework is proposed to detect DoS attacks on the network and attacks against the normal operations rules of the CoAP protocol. An IoT security framework for Smart Home is introduced in [148] and a general threat model to recognize the vulnerabilities for IoT services against cyber-attacks is analyzed. A smart home testbed is considered to monitor variables and control elements with different protocols

such as Wi-Fi, ZigBee, etc. An IDS based on anomaly behavior analysis (ABA) of the end nodes operations allows to detect and classify a wide range of attacks against IoT devices, such as replay attacks, delay attacks, DoS or DDoS attacks, noise injections, etc.
In [76] a security framework for home network is proposed with residential gateways as devices responsible for the exchange of information between the ISP infrastructure and the customer network to develop a vastly distributed IDS/IPS, enforcing preventive or corrective countermeasures, according to the instructions issued by the ISP. This security framework, as other papers in literature, foresee to move the security intelligence in the ISP. In this work, only the bare minimum necessary to calculate the users' similarity score is known by the ISP, while the actual countermeasures (i.e., the network configuration) is left to the smart home side (SHIELD Home device). This enables scalability (because the ISP does not have to address all the users' CPE details), fine-grained configuration (the SH device knows more precisely the actual user's network configuration), and it allows the user (if he is an expert) to override the security setup proposed by the ISP.

Moreover, the proposed solution provides a high degree of flexibility with respect to the different IoT networks, which are often user-owned and deployed. These networks are difficult to manage by the ISP in a "pure" centralized way and all the current approaches are not sufficiently reactive and dynamic to protect the smart environments adequately. As a matter of fact, in the current architectures attacking a network triggers only a *local* response, and it does not have any system-level reaction. We believe that our *network sentiment analysis* fills the gap between actual network security techniques and a more coordinated reaction system.

5.2.3 SHIELD system

In a Smart Home each Internet-connected device has its own peculiar security and availability requirements. There are still a number of issues to be addressed, particularly in the IoT network section, related to the devices deployment and initialization.

The real problem is to have a dynamic security protection system able to react to possible threats by re-configuring in real-time the security defenses of the network (e.g., firewalls). As a matter of facts, the main problem with traffic filtering is the user. If the firewall is too restrictive, the user (if he/she is an expert) will disable (or circumvent) it. If the firewall is too few restrictive, it is useless. Moreover, the vast majority of the users do not have the technical skills to understand the firewall configuration and to autonomously update the firewall rules if a new threat is discovered.

A common approach is to protect the user with an ISP-level firewall, but this architecture is not scalable and it does not adapt to the needs of different users. Moreover, the attack could be originated from *within* the Smart Home network thanks, e.g., to an infected mobile device. For this reason it is mandatory to provide both types of protection: at ISP level and in the user's premises. Therefore it is mandatory the use of IDSs to add end-to-end threats protection to the smart home network and the Internet. [158].

Signature based and anomaly behavior based IDSs are extremely useful, but they are even more difficult to configure for the end-user. Moreover, any IDS kind is subject to a lot of false warnings, either false positives or about attacks that can not succeed (e.g., an attack aimed at a type of device that is not in the network). In order to avoid an excessive computational complexity on the IDS, only the relevant threats for a given network should be trapped, plus the ones that are believed to be actively being exploited by attackers. As a consequence, IDS as well must be dynamically configured in response to ongoing threats. Moreover, signature and anomaly based IDS analyzes usually data traffic while new IDS should be specifically designed for IoT smart home environment where the attacks can also be the altering, e.g., of the reported data from sensors, or the controls of actuators.
In 6LoWPAN networks, 6LBRs are used to integrate the WSN with Internet and thanks to their resources availability they can support intrusion detection system and generate alerts. As consequence, a hybrid topology for IDSs can be envisaged where detection capability are distributed and with a central unit responsible for decision operations and countermeasures.
For this reason, the proposed architecture is outlined in Figure 5.3. This architecture is designed to guarantee the interoperability with existing Internet standards and the communications of sensing devices with other Internet components in the context of future IoT distributed applications.

In Figure 5.3, the three major components of a Smart Home environment are outlined: devices connected by high-bandwidth wireless networks (typically Wi-Fi), devices connected through cables (Ethernet, Power Line Communications, etc.), and IoT devices - in the figure the IoT devices are represented as IEEE 802.15.4 nodes, but other standards are possible. Furthermore, IoT devices can be differentiated according to their role and their

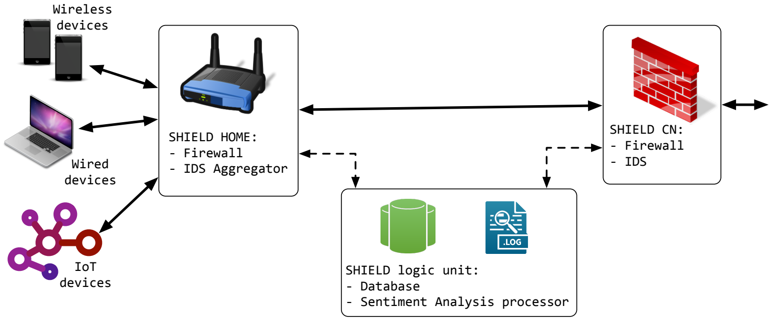


Figure 5.3: SHIELD Architecture

security/reliability requirements [85].

Each device type, and every connection technology needs its own particular detection approach. As an example, wired connections need only a wiretap on the main switch (usually the home gateway), while special receivers will be needed for wireless connections (e.g., high gain antennas).

To detect attacks in multihop networks (e.g., IEEE 802.15.4) it is possible to use multiple probing point on special nodes. It is worth noticing that the probe points should be connected to a master IDS engine through a secure side-channel, and that the load balancing between the master IDS and the probes depends on the side-channel congestion and the energy consumption of each device.

In the security system for smart home, hereafter called SHIELD system, the SHIELD – Core Network (SCN) is the element in the ISP network that is responsible for a) firewalling the ISP and users networks from attacks originating from the outside of the ISP network, b) analyzing the traffic to spot suspicious traffic and c) collecting all the data from the ISP IDS.
The network element responsible for protecting the home network side is the SHIELD – HOME (SH). From a logical point of view, the entities are two, the SHIELD Firewall and the SHIELD IDS Aggregator. The first is responsible for filtering the traffic from and to the ISP, while the second controls and harmonizes all the different IDSs present in the SH network.
The SH is responsible for i) correctly configuring the IDS to match the threats that are meaningful for the Smart Home environment (e.g., by silencing the alarms for patched devices), ii) activating security countermeasures (e.g.,

firewall rules, to block further communications from the attacked sensor) for actively exploited vulnerabilities.

The "core" of the SHIELD architecture is the SHIELD – Logic Unit (SLU). This entity receives all the (anonymized) IDSs alarms and warnings from all the SH in the ISP domain. The SLU focuses on discovering various relations between individual warnings/alerts and, according to alert correlation scores, it can change the *Network Sentiment* level and take various countermeasures, e.g., modify the firewalls configurations, block further communications from the attacked IP address, up to block further communications between the attacked Smart Home domain and Internet. In the SLU, the *Network Sentiment* processor analyzes *similarity* scores among different smart home networks, in order to implement preemptive countermeasures in these networks. As a matter of fact, it is more than possible that an attack will propagate to smart home networks "close" to the network under attack, as shown in Fig. 5.4.

The SLU logic is presented in Fig. 5.5, where it is outlined the different behavior with respect to an ongoing attack (*Alarm*) and to a possible attack (*Warning*). The first triggers an immediate reaction, while the second is evaluated according to the frequency of similar warning alerts, and the presence of similar warnings sent by different users. The SLU evaluates the attack type, the possible outcomes, and can select the most appropriate mitigation techniques, eventually modifying the ISP-level firewall rules. However, the most important element of the SLU is, in our opinion, the capability to propagate the *Network Sentiment*, i.e., the overall status of the network with respect to ongoing attacks, to the different SH units. This feature enables a *preemptive* security approach, where a SH is "immunized" from an attack that did target another SH (Fig. 5.4).

Moreover, the SLU can issue periodic or triggered warnings (e.g., e-mails, messages on the SH display, on application etc.) to inform the users about the firewalling decisions, how to improve the network security, etc. In this way, the users should be able to customize the system to better suit their needs. As an example a non-technical user could aim for maximum automatic protection, while a skilled user could decide to ignore some threats and planned actions (at his own risk).

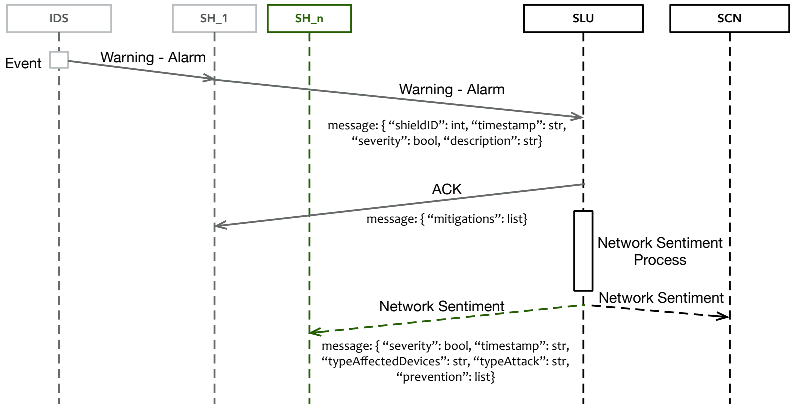

IDS
SH_1
SH_n
SLU
SCN
Event
Warning - Alarm
Warning - Alarm
message: { "shieldID": int, "timestamp": str, "severity": bool, "description": str}
ACK
message: { "mitigations": list}
Network Sentiment Process
Network Sentiment
Network Sentiment
message: { "severity": bool, "timestamp": str, "typeAffectedDevices": str, "typeAttack": str, "prevention": list}

Figure 5.4: SHIELD blocks interaction

5.2.4 Network Sentiment

The *Network Sentiment* should not be confused with a simple reaction to an ongoing threat. On the contrary, the *Network Sentiment* is something that affects each user in a different way, and it should be built according to a number of parameters that are inherently user-specific.

As an example, a user with no computers or devices of type X will not have its *Network Sentiment* changed if there is an ongoing threat specifically targeted toward this kind of devices.

The *Network Sentiment* is also aimed at evaluating and reacting to threats that are spatially or socially correlated. As an example, a port scan detected over a particular wireless network means that the attacker is physically close to the target network. As a consequence, the physically nearby networks must activate proper countermeasures.

However, the *Network Sentiment* protection can be extended to non-physical spaces, like the participation to groups, e.g., social media, common interests, etc. An attack spreading through social engineering and/or social media interactions can be actively mitigated.

As a matter of fact, the *Network Sentiment* must:

- Evaluate the presence of an attack (or the attack probability),
- Evaluate the means of the attack spreading, and

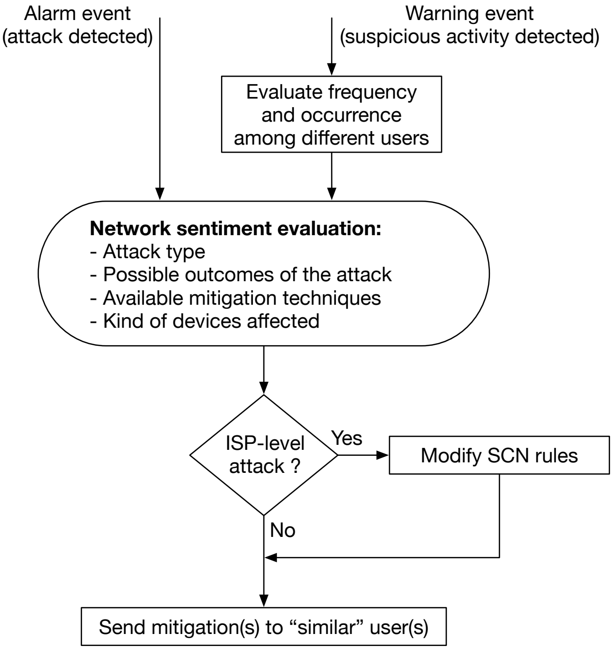

Alarm event
(attack detected)
Warning event
(suspicious activity detected)
Evaluate frequency and occurrence among different users
Network sentiment evaluation:
- Attack type
- Possible outcomes of the attack
- Available mitigation techniques
- Kind of devices affected
ISP-level attack ?
Yes
Modify SCN rules
No
Send mitigation(s) to “similar” user(s)

Figure 5.5: SLU logic

- Use the users similarities to strengthen the protection of the potential victims, where the users similarities refers to the attack type, network kind (e.g., devices, topology), users’ behavior (e.g., use of Internet services, online social relationships), network location, etc.

In order to evaluate the *Network Sentiment*, the SLU must collect several SH information e.g.,:

- The OS type and version being used in the Smart Home. This is necessary to evaluate the presence of vulnerabilities in the devices.
- The user’s location - to evaluate the likeliness of “geographical correlated” attacks (e.g., Wi-Fi password cracking attempts)
- The user’s social behavior - to predict the spread of social-spreading malware, e.g., malware carried by social platform like Facebook.
- Ongoing attacks (low confidence, high confidence, confirmed),

- Attack type (e.g. DoS, fault data injection),
- Attack effectiveness (i.e., if there are infected hosts in the user network),
- Number of blocked attacks,
- etc.

These information can be organized in an Intrusion Detection Message Exchange Format (IDMEF) message [78] and sent to the ISP. The exact message encoding is not important in this context, but it should be standardized to allow the interoperability between different vendors. Moreover, the SLU must evaluate Common Vulnerabilities and Exposures (CVE) reports [126] to properly block potential or ongoing threats and to suggest possible actions to the users (e.g., system upgrades).

Thanks to the users reports, the ISP can compute different types of metrics. Some alert reports (e.g., a confirmed ongoing attack) will result in an immediate response by the system. The reaction will still be dependent on the attack type, e.g., a phishing attack could result in an alert to the 'friends' of the attacked used (measured by the mail messages, social media connections, etc.). Other attacks will need more reports, possibly by different users, to trigger a response. This is particularly true for suspicious activities that could be simply an unexpected, albeit normal, user behavior. In this case, a consensus-based algorithm can be used to evaluate the threat.

The feedback from the SLU to the SH can be performed by IDMEF messages or, like in the previous case, any other standard message type.

Summarizing, the exchange of the *Network Sentiment* informs both the SH and the SCN about the ongoing attacks in the network. In this way, the whole ISP network is treated as one whole network, without the distinction between CN and users LANs.

Enhanced services enabled by SHIELD

The SHIELD system is meant to protect the user network with minimal interaction with complex devices like firewalls and IDS. However, to think about it as a simple security framework would be reductive. As a matter of fact, the SHIELD devices can greatly improve the user experience and, at the same time, provide a way to keep the user up-to-date with the current and ongoing threats to his/her network without generating overreactions.

The user should be constantly aware of the status of their network, including the ongoing attacks. However, this must not raise anxiety in the user, but promote a conscious utilization of the network. As a consequence, the SH device must constantly communicate with the user, e.g., by infographics on a display or via smartphone applications. Putting the user in the loop can also increase the good behaviors of the users, like keeping their systems up-to-date, and even the device vendors ones, increasing the chances that a vendor will actively support its products by patching security bugs in the devices firmware.

Moreover, SHIELD acts as a true IPS. As a matter of fact a “traditional” IPS can block an *ongoing* threat, while SHIELD can block a threat *beforehand*, simply because another user in close network is subject to the same threat. As a consequence, SHIELD is really a *prevention* mechanism.

Security considerations

Like every networked system, also SHIELD is sensible to threats. As a matter of fact, an attacker could take advantage of the SHIELD system to fake an ongoing attack, causing (for example) a DoS. For this reason, it is important that the SHIELD system is protected and considered as a primary asset of the ISP. If we define a *security zone* as a network area with a well-defined perimeter and a strict boundary protection, we have that:

- The SLU and the SCN are in the same security zone,
- The SLU and the SH are in *different* security zones,
- The SCN and the SH are in *different* security zones.

In other terms, the SLU and the SCN are inside the administrative domain of the ISP and their communication security is automatically guaranteed. On the contrary, a Man in The Middle (MitM) attack between the SH and the ISP network is considered possible because an attacker can disguise itself as a legitimate user. As a consequence, the communications between the IDS probes (or the distributed IDS system) and the SH must be adequately secured.

It is out of the scope of this work to describe the security algorithms that can be used to properly secure the above mentioned communication channels but the system should, at minimum, provide a strong authentication between the SH and the other two SHIELD entities, possibly by using

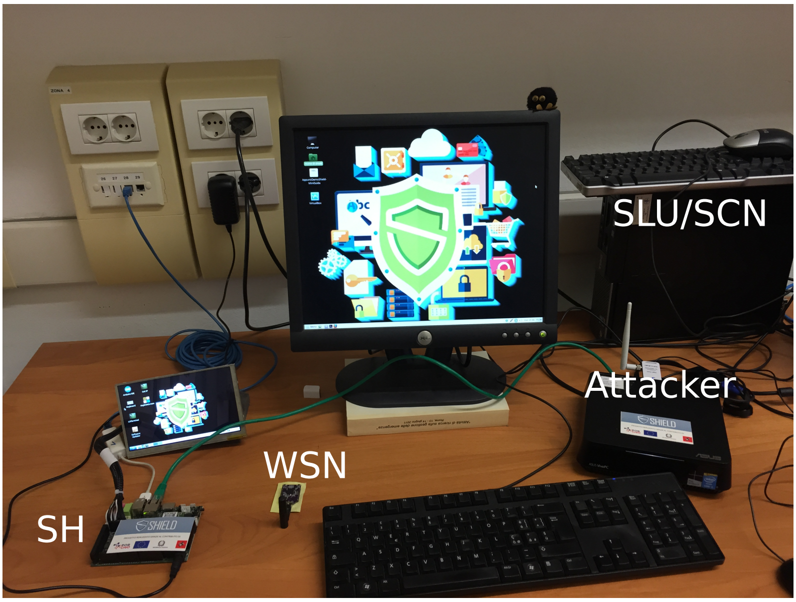


Figure 5.6: SHIELD testbed

certificates and/or smart cards. Moreover, the SH device should pass severe vulnerability assessment tests and be tamper-proof. If this is not the case, the SHIELD system must consider all the attacks reports from the users as simple warnings (low confidence reports), and use consensus algorithms to reject spurious data.

5.2.5 Smart Home Testbed

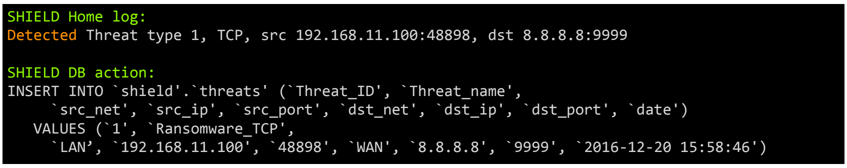
In order to evaluate the SHIELD framework, a prototype of the SH and the SLU is building. In particular, the SH prototype is equipped with Ethernet, Wi-Fi and IEEE 802.15.4 interfaces. Moreover, different attack types have been tested to evaluate the system feasibility.

The SHIELD testbed is shown in Fig. 5.6. The SH (shown to the left) is an UDOO board and an OpenMote CC2538 module as 6LBR. The UDOO board is a single board with a ARM Cortex-A9 CPU, RAM DDR3 (1 GB), GPIOs, mircoUSB ports, Gigabit Ethernet and WiFi module while OpenMote-CC2538 is based on Texas Instruments CC2538 System on Chip with an IEEE 802.15.4 transceiver. The SH has three local interfaces (Wi-Fi, Ethernet, and IEEE 802.15.4) and an Ethernet link to the (emulated) ISP network. The SLU and part of the SCN have been emulated with a virtual machine (PC at top-right of Fig.5.6). The used Smart Home devices are connected through the Ethernet, Wi-Fi and IEEE 802.15.4 interfaces of the SH. The attacks have been performed with a normal PC (center right in the Fig. 5.6).

The SH monitors all data traffic of the private network on all involved interfaces. Whenever SH detects an attack to some device connected to the network, it generates an alert that is sent to the SLU.

Without loss of generality, we used the Bro Network Security Monitor [10] to monitor the traffic, with custom rules to detect the possible attacks used during the tests. Moreover, we have prepared a python3 script in order to parse the IDS file logs searching for warning/alarms. The script parses both Ethernet and Wi-Fi logs and, when a warning/alarm is found, contacts the SLU. In particular, the Bro alerts have been converted to an appropriate inter-exchange format (IDMEF) and sent to the SLU for further processing. In response to an alert, the SLU will send a command to the SH which will take appropriate actions according to the type of threat detected. A simplified and interactive visualization of the alert is also provided to the user through a graphical interface.

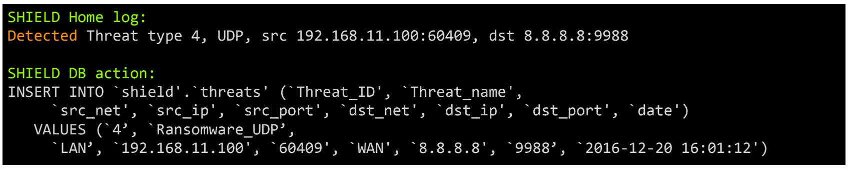
As mentioned earlier, the threat report could also not trigger any action of the SH in case the *Network Sentiment* for that particular user, according to that particular attack, is not changed (i.e., the attack is not relevant for the user). Nonetheless, the attack could be relevant for other users, and it is important to report its presence. For this reason, the SLU inserts the threat in the Shield database. In our experiment, the database is a MySQL db with customized tables. The chosen hardware reflects as much as possible a normal Linux-based CPE. As a consequence, implementation of the SH functionalities on a commercial CPE will be very easy. In this case, the CPE becomes the central element of the Smart Home network, guaranteeing not only the device connectivity, but also the whole network security. However, it is also possible to use an external SH unit, provided that it can monitor the home network links (wired and wireless).



```
SHIELD Home log:
Detected Threat type 1, TCP, src 192.168.11.100:48898, dst 8.8.8.8:9999

SHIELD DB action:
INSERT INTO `shield'.`threats' (`Threat_ID', `Threat_name',
     `src_net', `src_ip', `src_port', `dst_net', `dst_ip', `dst_port', `date')
   VALUES (`1', `Ransomware_TCP',
     `LAN’, `192.168.11.100', `48898', `WAN', `8.8.8.8', `9999', `2016-12-20 15:58:46')
```

Figure 5.7: Detection of Ransomware attack on TCP



```
SHIELD Home log:
Detected Threat type 4, UDP, src 192.168.11.100:60409, dst 8.8.8.8:9988

SHIELD DB action:
INSERT INTO `shield'.`threats' (`Threat_ID', `Threat_name',
     `src_net', `src_ip', `src_port', `dst_net', `dst_ip', `dst_port', `date')
   VALUES (`4’, `Ransomware_UDP’,
     `LAN’, `192.168.11.100', `60409', `WAN', `8.8.8.8', `9988’, `2016-12-20 16:01:12')
```

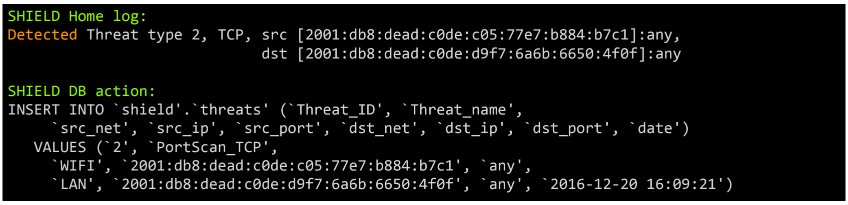
Figure 5.8: Detection of Ransomware attack on UDP

Attacks

To test the SHIELD system functionalities, we implemented three types of attacks: a ransomware, a port scan attack and an unauthorized access / query to the sensors via the CoAP.
Ransomware. To simulate the attack, we simulated a connection from a PC connected by Ethernet to an external host on a particular set of ports and with a given payload signature. The attack can use Transmission Control Protocol (TCP), UDP, IPv4 or IPv6. Fig. 5.7 and Fig. 5.8 show, respectively, the detection of the two threats (ransomware on TCP and UDP) by the parser log. Once the threat is detected, the IDS sends an alert to the SLU. In this case, the reaction is to block the communication and promptly alert the user to take immediate actions.
Port scanning. To simulate unauthorized intrusion, we performed a port scan on a LAN-connected host from a machine connected via the Wi-Fi network. The command is:

```
# nmap -6 -sS --data-string deadbeef 2001:db8:dead:c0de::b981
```

Fig. 5.9 shows the detection of the port scanning attack by the IDS and, consequently, the notification of the attack from the SH to the SLU. The reaction to this threat is twofold: the attacker is isolated (e.g., by discon-



```
SHIELD Home log:
Detected Threat type 2, TCP, src [2001:db8:dead:c0de:c05:77e7:b884:b7c1]:any,
                             dst [2001:db8:dead:c0de:d9f7:6a6b:6650:4f0f]:any

SHIELD DB action:
INSERT INTO `shield'.`threats' (`Threat_ID', `Threat_name',
     `src_net', `src_ip', `src_port', `dst_net', `dst_ip', `dst_port', `date')
   VALUES (`2', `PortScan_TCP',
     `WIFI', `2001:db8:dead:c0de:c05:77e7:b884:b7c1', `any',
     `LAN', `2001:db8:dead:c0de:d9f7:6a6b:6650:4f0f', `any', `2016-12-20 16:09:21')
```

Figure 5.9: Detection of a port scan from an host in the Wi-Fi network

necting the terminal from the Wi-Fi network), and the SH configuration is updated to react to port scanning attacks. Moreover, the Network sentiment processor sends updated configurations to the geographically close SHs. The updated configurations will also include increased UDP port scan detection and an increased security against unauthorized access, for example by forcing a key refresh on all the wireless devices. The last action can be justified by assuming that the attacker gained access to the victim's Wi-Fi network and, by extension, could also try to attack the neighbor networks.

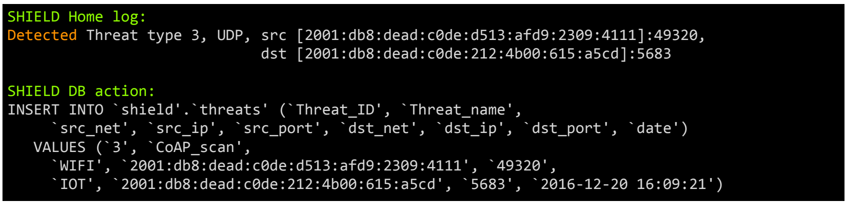
IoT devices attacks. In 6LoWPAN network, CoAP was designed for resource-constrained devices with the goal of guaranteeing interoperability with the web. It uses UDP over IP as transport stack. A 4-byte fixed header and a compact encoding of options are taken on to reduce the transmission overhead by limiting the fragmentation on the link layer.

As previous shown in Section 5.2.2 the attacks on routing operations with RPL are analyzed mainly in literature although the using of security mechanism such as Datagram Transport Layer Security (DTLS) in CoAP (CoAPs) does not assure protection against DoS and other types of messages, e.g. malformed CoAP requests. As consequence attacks against the application layer can be underestimated. In this case, an attacker attempts to interrogate a sensor via the CoAP protocol [161], trying to read the device hardware and firmware characteristics. This kind of attack can have different outcomes, ranging from violation of the privacy, to the devices battery draining. In these tests, the attacker would read the sensor temperature and the board firmware details:

```
# coap-client -v 5 coap://[v6addr]/sensor/temp
# coap-client -v 1 coap://[v6addr]/riot/board
```

where v6addr is `2001:db8:dead:c0de:7dfe:dff9:f7ff:ddf7`.

In Fig. 5.10 the attack is detected by the IDS and through the SH sent



```
SHIELD Home log:
Detected Threat type 3, UDP, src [2001:db8:dead:c0de:d513:afd9:2309:4111]:49320,
                             dst [2001:db8:dead:c0de:212:4b00:615:a5cd]:5683

SHIELD DB action:
INSERT INTO `shield'.`threats' (`Threat_ID', `Threat_name',
     `src_net', `src_ip', `src_port', `dst_net', `dst_ip', `dst_port', `date')
   VALUES (`3', `CoAP_scan',
     `WIFI', `2001:db8:dead:c0de:d513:afd9:2309:4111', `49320',
     `IOT', `2001:db8:dead:c0de:212:4b00:615:a5cd', `5683', `2016-12-20 16:09:21')
```

Figure 5.10: Detection for IoT attack

to the SLU. The detection is triggered by the requests frequency and the suspicious request of the board firmware version. As a matter of fact, the user only needs this information when upgrading the node firmware, while for an attacker it represents an useful information to perform a targeted attack. The reaction includes a limitation on the requests from the specific user (rate limit), alerts to the authorized users about the suspicious activity, a more detailed analysis on the traffic from the attacker up to the advertisement to the SCN unit.

In conclusion in this work an IoT security framework for smart home environment is presented. The idea is to reach a dynamic security level for this smart infrastructure based on a *network sentiment analysis*. In the proposed architecture, smart gateway/firewalls at the ISP side and close to the user smart home cooperate to detect and react against different types of attacks from within the smart home network, i.e. the Wi-Fi, Ethernet world or from IoT devices. In particular, the user gateway has not a predefined and fixed security level established when it was installed but can change its security rules and actions in reaction to the dynamic threat level measured by the SLU.

The two zones ISP and the smart home LAN cooperate in a unique integrated security framework. For example, if the smart home gateway (the IDS system) detects an attack, it reacts and the same information, distributed to the ISP, can be spread to the other gateways near to the smart home under attack (e.g., in a building we can consider a gateway for each smart home) to increase also their security and prevention level in a *social* security vision.

A testbed validates the approach feasibility and that a simple CPE can easily perform all the required tasks to guarantee a Smart Home security.

The examples demonstrate that the SHIELD architecture allows a great flexibility on the kind of attacks to react to, without the installation of complex rules on the SH. Moreover, the *Network Sentiment* analysis allows the integration of behavioral, signature or hybrid based IDSs, enhanced by the knowledge of similarity-driven activity reports.

Future works will be oriented toward the design and evaluation of automatic attack correlation engines, in order to enhance the SLU sentiment analysis processor. In particular, machine learning algorithms are considered as potential candidates for the automatic interrelationship analysis between attacks and users relationships (physical or social).

5.3 IoT Security via Address Shuffling: the Easy Way

In this section the work [146] is presented. This work has been published on IEEE Internet of Things Journal.

Securing IoT devices and protecting their applications from privacy leaks is a challenge, due to their weak (computational and storage) capabilities, and their proximity with sensitive data. Considering the resource-constrains of such devices, their long lifetime, and the intermittent connections, classical security approaches are often too difficult or impractical to apply. Moving Target Defense is an established technique whose goal is to lower the attack surface to malicious users by constantly modifying device footprint. Changing the address to an IoT device without privacy leaks is, however, a non-trivial task. To add a MAC-level randomness, a system could periodically change the MAC address of each node. This technique alone has, however, various drawbacks. In particular, the signaling overhead required to coordinate the address change is significant. While in wired network the overhead may not be a concern, when devices are constrained by bandwidth and power, such overhead may severely impact the IoT system performance. It is hence important to devise (MAC) address renewal methods with minimal impact on the network signaling overhead.

To this aim, in section an a novel address shuffling technique is presented. This novel algorithm is called *AShA*, as in **A**ddress **Sh**uffling **A**lgorithm. *AShA* is energy-efficient, has minimal impact on the network overhead, and it is easy to implement. The key novelty behind *AShA* is a cryptographic hash that enables a controlled and collision-free address shuffling. Only the legitimate nodes and the network controller are able to predict the address renew outcomes, and from the point of view of the attacker, the addresses follow a random pattern over time.

The efficiency of our proposed method is evaluated with respect to the number of nodes in the network both theoretically and through simulations. The results show that, for typical network sizes (i.e., less than 700 nodes for each PAN), the address renew procedure does not add any overhead, while for larger PANs (i.e., up to 2300 nodes) the overhead is tunable, and it depends on network management decisions.

5.3.1 Threat Model and Motivating Scenarios

Debates regarding what are the major security threats in Ad-Hoc Networks with an IP address auto-configuration have been floated before (see, *e.g.*, [182]). We have identified four major security threats:

1. *Address Spoofing Attack.* An attacker may spoof IP addresses and inject (replayed or forged) packets.

2. *False Address Conflict Attack.* An attacker may leverage address collision prevention mechanisms to perform a denial of service.

3. *Address Exhaustion Attack.* An attacker may exhaust the space of available IP addressing by declaring usage of a large number of them, preventing new devices to to join the network for lack of available IP addresses.

4. *Negative Reply Attack.* An attacker may continually transmits a (false) deny message when a new node tries to obtain a new address. In this case, the attacker maliciously acts as a Network Coordinator.

Beside address spoofing, all attacks are based on vulnerabilities of the link-local protocols, and in particular, DAD and NDP. Message encryption and validation at layer 2 is normally used to mitigate such attacks.

Preserving Privacy in Internet of Medical Things. Preserving privacy of medical records is often a law requirement [60]. Aside from the potential benefits of protecting electronic medical records, which include lab tests, images, diagnoses, prescriptions and medical histories, recently body area networks have been used for therapeutics monitoring, for example of pacemakers. With such technology, healthcare providers may instantly access patient histories that are relevant to future care and patients can take ownership of their medical records. In general, medical record as well as IoT devices offer the potential for greater privacy and better access to patient data when they are needed. Attackers may, however, leverage these information for data leaks or even to hijack pacemakers or other body area devices via address spoofing or address exhaustion attacks[4].

Preserving Privacy in Edge Computing. Edge computing is a paradigm based on outsourcing computational tasks from a distributed set of end-hosts, or viceversa, edge nodes can be used to release pressure from data centers [181]. In edge computing, privacy preservation is more challenging since end-nodes may collect sensitive information such as identity or locations of end users (that may contain sensitive information otherwise stored at the network core). Moreover, since edge end-hosts are scattered in large areas, centralized control is becoming difficult. A poorly secured distributed edge computing network can be the entry point for intruders. Once inside the network, the intruder can mine, steal, damage or leak users or machine sensitive data. Location privacy is arguably one of the most important models for privacy, since pieces of hardware can be linked to owners (data). Since in edge computing, offloaded tasks may include location, trajectory and even mobility habits, such information revealed to an adversary may be part of a weaponization phase and may hence lead to subsequent more severe attacks.

Securing Industrial Control Systems and Smart Grids. The integration of IoT into industrial systems has been fostered to catalyze the potential automation of tasks such as manufacturing or product tracking. These systems often rely on strict timings and precise execution, requiring close monitoring to function safely and effectively. Control systems like these have long been used, but recently, several sensors have been attached to the Internet for remote maintenance shifting to IoT architectures. An example of that are power grids, that are quickly becoming smart grids, using automated metering infrastructure to remotely sample usage data from residential meters. A simple address spoofing attack may potentially jeopardize an entire production chain with costly consequences. As shown in [98] smart meters readings can disclose information about the time that the house is empty (i.e., it is safe to break in) or even the TV programs that a user prefers to

[4] `https://www.theguardian.com/technology/2017/aug/31/hacking-risk-recall-pacemakers-patient-death-fears-fda-firmware-update`

watch.

Protecting Injured or Lost People in a Disaster Scenario. During or after a disaster scenario, such as hurricane Katrina, mobile ad-hoc networks are established by first responders to cope with the lack of (disruptive) network infrastructure; attackers may intercept network traffic among first responders to arrive first on victims and identify subjects for human or organ traffic; these attacks may be delivered for example, via a False Address Conflict Attack.

5.3.2 Related Work

In literature, several solutions for IP address auto-configuration have been proposed. The authors in [184] classify these solutions in three major groups: stateful, stateless and hybrid. In *stateful* address auto-configuration approaches, all nodes consult a logically centralized node or a set of distributed agents to obtain a valid IP address. The Dynamic Host Configuration Protocol (DHCP) protocol [82] is part of this family. In *stateless* address auto-configuration approaches, every joining node self-assigns an address; such address may be chosen at random or with a predetermined algorithm. Stateless address auto-configuration approaches lead to addresses conflicts, that are resolved with overhead, delays or with a centralized coordinator and, as consequence, they scale poorly. Without a coordinator, two nodes may end up choosing the same address. To avoid such address uniqueness issue, the DAD [175] algorithm is used. Every node, after selecting an address, performs DAD to detect any collision. SLAAC [99] is part of this family. The proposed solution enables to auto-configure addresses without the need of explicit address duplicate checking, as this function is already fulfilled by the network coordinator. However, unlike DHCP, this method does not require any message exchange between the node coordinator and the nodes, with the exception of a network-wide address renew message broadcasted from the network coordinator to all the nodes.

It may be unsuitable to use the DHCP protocol in an ad-hoc network, due to the limited device's energy, its (possible) mobility, and the insufficiently resilient infrastructure (a dedicated server may become unavailable). Existing works have proposed two approaches in response to the drawbacks of DHCP: either an adjustment of DHCP to enable a dynamic IP address allocation, or the use of a coordinator that assigns addresses. For example, in [46], authors assign an IP address to the mobile devices using an Ad-Hoc

DHCP scheme, while in [185] a central node (called ZigBee Coordinator) is responsible for assigning addresses.

In [140], the authors propose a dynamic host configuration to assign the IP addresses based on a distributed agreement problem: when a new node joins the network, a selected node proposes a candidate IP address: if such proposal reaches consensus, the proposed address is assigned to the new node, otherwise another address is proposed (a given number of times).

The Internet-Draft on *Ad Hoc Address Auto-configuration* [153] proposes a stateless approach to auto-configure IP addresses that are unique in the network. A DAD-like protocol is used to check the proposed address uniqueness before the auto-assignment. Briefly, a node chooses two addresses: a temporary used as IP sender address only, and a proposed sent to the other nodes to check if it is unique in the network. If no answer is received within a given interval, the node assigns the proposed address to itself. Obviously, if an answer is received, the node changes the proposed address and tries again.

In [170], an hybrid scheme is used: the auto-configuration is performed by the device but, to maintain the information in the network and to detect duplicate addresses, an elected coordinator (called Address Authority) is used.

These solutions focus only on the IP address auto-configuration and do no consider several critical points, such as the static nature of MAC address/Interface Identifier (IID). In [173], the author studies some problems arising from a static IID. The author suggests that not only the IP address must be periodically changed, but also the layer 2 address (MAC address) must be changed as well.

The authors in [57] provided a method based on an extension of 6LoWPAN-ND to change both L2 and L3 addresses, while maintaining the header compression (as well as all other functionalities of the 6LoWPAN) and a small routing table. The technique presented in this work, with respect to [57], maintains all the original benefits, reducing the network overhead to zero for small to medium size networks. Moreover, the active sessions and routing path management during the address renew are greatly simplified.

5.3.3 Proposed Method

Existing address shuffling techniques are based on complex protocols [36, 116]. All of them assume that a central entity (a coordinator) knows about all the devices associated to the network, and it can manage reliably the association/disassociation of each node in the network. This assumption is usually true for all the practical IoT networks. In this paper we will assume that the association/disassociation phases are managed at MAC level.

As seen in Section 5.3.2, the coordinator usually (re)assigns the node addresses and ensures a lack of address collision. A simple way to re-assign an address to a node is to let the coordinator choose it randomly, avoiding address collisions. However, the coordinator must send a unicast message to every device in the network, incurring in huge signaling costs. Another solution is to let each node choose its new address randomly, and to let the coordinator the duty to correct collisions. To do so, however, the coordinator is required to send a unicast message to inform every device in the network about its correction decision. This solution is costly (the signaling cost is very high), and, especially when applied to medium or large networks, often requires extensive convergence time. An alternative is to let the node itself choose the address randomly, using a DAD (or an equivalent protocol); this solution is complex, also requires extensive signaling, and may have similar execution times. Moreover, in a multi-hop ad-hoc networks, multicast messages might not be supported, requiring a fallback to even less efficient unicast communication.

In this section a novel algorithm for address auto-configuration in which each node autonomously calculates the new address, and such address is known to the network coordinator in advance. The coordinator will only signal to all nodes that the address needs to be changed, while avoiding potential conflicts. The idea is to allow each node to choose autonomously a new (MAC-IPv6) address (pair), guaranteeing at the same time that each new address is unique in the network. In this proposed scheme, the PAN coordinator role is to choose a set of parameters that, once spread to the nodes, will trigger a collision-free address renew.

The goals of an optimal ad-hoc network addresses auto-configuration algorithm are: *dynamic address configuration*, *uniqueness*, *robustness* and *scalability* [171]. With such auto-configuration algorithm, a node is able to obtain dynamically a unique IP address without static or manual configuration; moreover, the algorithm must be able to adapt as the network conditions evolve and without performance degradation as the number of devices grows. The proposed algorithm meets all the above requirements.

AShA

Address Shuffling Algorithm with HMAC (AShA) is based on a clever use of hash functions, and in particular Keyed-Hash Message Authentication Code (HMAC). An *Hash function* is a function that takes as input an arbitrary length string and produces a fixed-size result [86]. The output is called *digest* and it is function of the input bits. Since hash functions are realized with elementary operations (i.e. SHIFT and XOR) they are very fast computationally. Hash functions have some (important) properties:

- The already mentioned *compression* and *ease of computation.* An Hash function (H) maps an input x of arbitrary finite bit-length to an output $y = H(x)$ of fixed bit-length n and given H and x is easy to compute y;
- *Pre-image resistance*: given an output y, it is computationally infeasible find any input x;
- 2*nd pre-image resistance*: given x_1 it is computationally infeasible find a different input x_2 that applied to the same hash function H gives the same output;
- *Collision resistance*: it is computationally infeasible find any two distinct inputs x_1, x_2 (arbitrarily chosen) that applied to the same hash function give the same output $H(x_1) = H(x_2) = y$;

Hash function are useful to guarantee messages integrity but to verify the authentication of a message the HMAC function is used. Eq. 5.1 describes the HMAC function as defined in [110]:

$$\mathrm{HMAC}(K, m) = H\big((K' \oplus \mathrm{opad}) \parallel H((K' \oplus \mathrm{ipad}) \parallel m)\big) \tag{5.1}$$

where:

- $H(\cdot)$ is a cryptography hash function that operates on blocks of B bytes iteratively,
- m is the message to be hashed,
- K is an arbitrary secret key.
- K' is a B-bytes key derived from K,

- ipad is the byte `0x36` repeated B times,
- opad is the byte `0x5C` repeated B times.

The properties of the HMAC function greatly depends on the hash function being used. However, is expect that the hash function fulfills the three hash functions properties, i.e., pre-image resistance, second pre-image resistance, and collision resistance. An interesting consequence of the collision resistance is that the hash function should be able to map the domain in the whole codomain, that is, all output bits are used.

Thanks to the hash properties, it is possible to derive a pseudo-random (IPv6, MAC or both) address of node i from two parameters:

$$\mathrm{ADDR}_i(K) = \mathrm{HMAC}(K, \mathrm{ID}_i) \tag{5.2}$$

where ADDR_i is the new address of node i, ID_i is an identifier for node i (e.g., its serial number), and K is a secret key. Note that it is possible (and advisable) to have a per-node key K_i. Without loss of generality, in the rest of the paper we will assume a shared key K for all the nodes of the network.

This method can be used by the nodes to self-assign a pseudo-random MAC/IPv6 address. As shown by the Birthday paradox, address collisions are improbable, but still possible, and the collision probability is given by [132]:

$$P(n,d) = 1 - \frac{d!}{(d-n)!d^n} \approx 1 - \exp\left(\frac{-n(n-1)}{2d}\right) \tag{5.3}$$

where d is the hash function image size (i.e., the address length) and n is the number of nodes. As a consequence, an address collision resolution mechanism is needed (see also [174]).

As an example, in Fig. 5.11 the birthday paradox effects for a 16-bit number (2^{16} available addresses) are shown. In other terms, using a simple HMAC (or any random generated number) to decide a new address would lead to higher collision probability with a network of merely 200 nodes.

This mechanism is impractical for periodic address changes since the node identifier and the key are constant. Even though a key renewal mechanism is in place, the key validity time should be considerably longer than the address renewal time, simply because the key renewal protocols generates significant overhead, which is against our design goals.

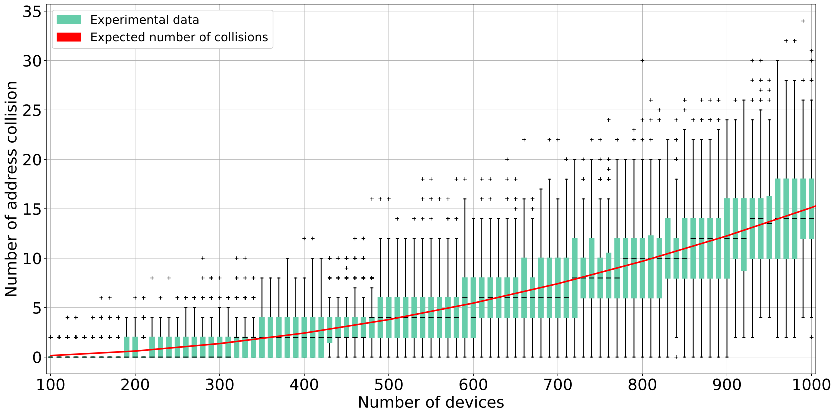


Figure 5.11: Birthday Paradox with 2^{16} MAC-16 addresses.

To allow a *periodic* and *overhead controlled* address change, the HMAC function must be changed by adding one more parameter, which can be set by the network coordinator requesting the address change action. By carefully choosing the new parameter value, the network coordinator can also avoid address collisions, as we demonstrate in Sec. 5.3.4. In the proposed system, a new address is created from a *triplet*:

$$\mathrm{ADDR}_i(K, r) = \mathrm{HMAC}\left(K, (\mathrm{ID}_i \parallel r)\right) \tag{5.4}$$

where r is an address refresh index (named in the following METAindex) managed by the network coordinator. With this method, the secret key can be updated independently from the address refresh procedure, and the network coordinator can control the address collisions by not using the values of r that would cause one, i.e., the coordinator changes r when a collision is detected. The secret K should be changed periodically in order to prevent possible attacks. As a matter of fact, it should be avoided to use more than once the same METAindex r without changing the secret key K. Toward this end, the coordinator must use a suitable key distribution protocol to refresh K.

The AShA algorithm

The proposed algorithm, called AShA, is previously run by the network coordinator and, at a later time, by the devices. The key K and the node ID are known by the device, while the METAindex parameter (r) is provided by the network coordinator. As a consequence, any node can calculate its MAC and IPv6 address independently.

When an address shuffling is required, for example as part of a moving target defense strategy, the network coordinator sets r to a particular value and evaluates, for a given K, all new addresses of the devices associated to its network. If a collision is found, the coordinator changes the METAindex and re-calculates all addresses. The loop ends when the network coordinator does not find collisions and subsequently, a multicast message with the r value to be used is sent to all devices. Algorithm 4 shows the pseudo-code of the network coordinator AShA algorithm. Parameters (K, r) are omitted for brevity.

Algorithm 4 network coordinator AShA

```
 1: Init: $r \leftarrow \text{METAindex}_{\text{Cur}} + 1$
 2: while $r \leq \text{METAindex}_{\text{max}}$ do
 3:     for all nodes do
 4:         $\text{ADDR}_i \leftarrow \text{HMAC}\,(K, (\text{ID}_i \parallel r))$
 5:     Collisions $\leftarrow$ False
 6:     if $\exists\, i, j \in \text{nodes} : \text{ADDR}_i = \text{ADDR}_j$ then
 7:         Collisions $\leftarrow$ True
 8:     if Collisions = False then
 9:         $\text{METAindex}_{\text{Cur}} \leftarrow r$
10:         Send $r$
11:         Break
12:     else
13:         $r \leftarrow r + 1$
14: if $r = \text{METAindex}_{\text{max}}$ then
15:     Generate and distribute new key
16:     $r \leftarrow 0$
17:     Go to 2
```

When the devices in the network receive the *address change multicast message*, with the new r value, they automatically evaluate their new MAC and IPv6 addresses.

This preventive control of the address collisions by the network coordinator, in addition to enable a simultaneous addresses shuffling in the network, avoids the use of onerous collision detection mechanism (e.g., DAD) used in IP networks. When the address reconfiguration is performed, the node is sure that its new (IPv6 and MAC) address does not collide with any other address in the network.

Moreover, AShA is able to mitigate all the security threats presented in Sec. 5.3.1. In particular, address spoofing is automatically resolved once the network addresses are shuffled, false address conflict is not anymore possible because addresses are handled by the coordinator, and address exhaustion can be mitigated by jointly using address shuffling and per-device authentication. Negative reply attack is mitigated by AShA for the nodes already in the network, while for new nodes trying to join the network it is necessary to use MAC-level encryption.

Theoretical Analysis

In this subsection the AShA mathematical properties are analyzed, and how to leverage the METAindex to further obfuscate the network.

Since AShA is based on a HMAC function, the probability of not having any collision in the network is based on the number of the nodes in the network. Once all the range of METAindexes has been used, it is necessary to renew the secret key K.

In order to avoid a frequent key renew process, it is advisable to be able to have a large enough pool of METAindexes to choose from. However, an attacker could discover the approximate network size by checking how many METAindexes are skipped and/or by counting the number of unicast address change messages. This potential information leakage could be used to perform further attacks, and it should be prevented.

An easy, and yet very effective way to further obfuscate the network operations is to split the METAindex in two parts: the Primary Index and the Secondary Index. The Primary Index is used to trigger the address change procedure, while the Secondary Index is used to find the right METAindex that does not generate a collision. In other terms, for each address shuffling the Primary Index is incremented, and the Secondary Index can be chosen randomly among the numbers that does not generate a collision. In this way, an attacker will not be able to analyze the AShA procedure to infer the network size.

The Primary Index and Secondary Index effects on AShA can be analyzed mathematically. Eq. (5.3) shows the classical Collision Probability formula. It can be demonstrated that, if the hash function follows the necessary hash properties (i.e., pre-image resistance, second pre-image resistance, and collision resistance), then eq. (5.3) can be applied also to the AShA case.

$P_H(n,d)$ is the probability that Primary Index r does not generate a collision using $\text{HMAC}\,(K, (\text{ID}_i \parallel r))$, and $\widehat{P}_H(n,d,k)$ is the probability that exist at least one Secondary Index r' such as a Primary Index r does not generate a collision using $\text{HMAC}\,(K, (\text{ID}_i \parallel r \parallel r'))$, where n is the number of addresses generated, d the length of the address being generated, and k is the length of r' (Secondary Index).

It is possible to show that:

$$P_H(n,d) = \frac{d!}{(d-n)!d^n} \approx \exp\left(\frac{-n(n-1)}{2d}\right) \tag{5.5}$$

$$\widehat{P}_H(n,d,k) = 1 - \big(1 - P_H(n,d)\big)^k \tag{5.6}$$

The effect of the two Indexes is shown in Fig. 5.12, where is clear how the introduction of a Secondary Index obfuscates the AShA dependency on the network size. It must be stressed that, thanks to the Secondary Index, a Primary Index can be used multiple times, while without the Secondary Index, each Primary Index can be used only once before renewing the key K.

The overall algorithm complexity for each secret key is $O(n \times 2^k)$ where n is the number of devices, k is the METAindex size in bits, $O(n)$ is complexity of the hash and the collision search, and $O(2^k)$ is the complexity of the while loop. However, this value is spread among multiple METAindexes. The complexity needed to find a new viable METAindex depends on eq. (5.5).

AShA implementation on LR-WPAN

In this subsection, we discuss how AShA may be apply to LR-WPAN.

Multi-hop WSNs are often organized in tree-like structures as, for example, in IEEE 802.15.4 Cluster-Tree PAN, RPL, etc. In such network topologies there are a *root* node, which corresponds to the PAN coordinator, all the other nodes communicate with the root by using multi-hop communications. A node, except for the root, has one (or more) *parent(s).* If a parent node is re-configuring, it blocks the communications of all the nodes using that

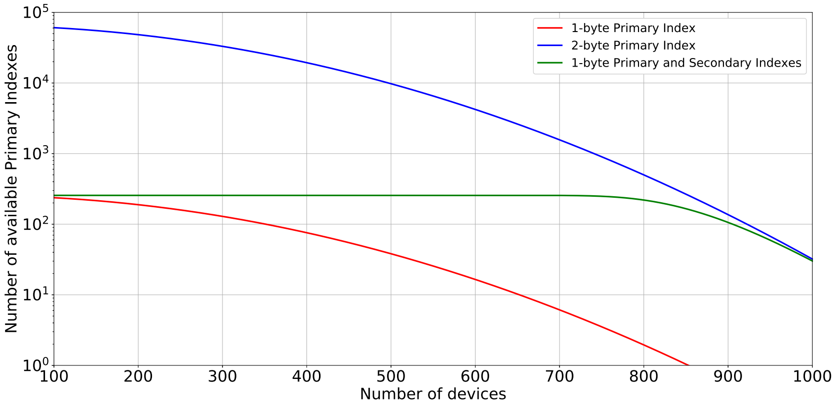


Figure 5.12: Effects of Primary and Secondary Indexes on *AShA*.

node as a parent. As a consequence, there is a delay in the reconfiguration process and a possible energy waste.

The address reconfiguration approach suggested in [57] is to start the reconfiguration from the tree leafs. However, this solution requires an exact topology knowledge and an additional coordination among the nodes.

The proposed solution completely eliminates the need for a complex exchange of messages with the network coordinator: they can switch to the new address autonomously as soon as they receive the address renew message.

To keep existing connections alive, it is useful to split the address space into two sets: one used before the addressing renew, and one after. In this manner, the PAN coordinator can keep forwarding, for a given period of time, the packets to the nodes that have active connections, even when their actual address is changed. Upon an address change, applications will have to re-negotiate a new connection with their endpoints.

For the sake of simplicity, the Least Significant Bit (LSB) is used for this purpose but any bit is a good choice. It is worth noticing that this bit does not add anything to the security or secrecy of the network, as it is fairly easy for an attacker to discover the used bit by controlling the network address pattern before and after an addressing renewal operation. All the involved technologies are explained in 2.2 and as mentioned in Sec. 2.2.1 the available MAC addresses are 57342, so each node can choose one of the 28671 available addresses at each change of address.

Each node knows its own MAC-64, which is globally unique by definition. Moreover, we assume that all the nodes share with the PAN coordinator a Secret key. The Secret key could be installed during the network setup by using a secure network parameters installation method (see for example [151]). Key derivation and management are standard procedure and hence are out of the scope of this work. However, a good approach could be to use the HKDF Scheme [111, 112].

The *DODAGVersion Number* is used to implement the AShA algorithm in a LR-WPAN: when a *DODAGVersion Number* changes, any node in the network is forced to use the new value. As a consequence, we can use the *DODAGVersion Number* as a reliable mean to spread the address renew data and to trigger an address shuffling.

The birthday paradox must be carefully evaluated when using AShA to calculate short (MAC-16) addresses. In general, for HMAC, the first collision is expected after evaluating approximately $\sqrt{\frac{\pi}{2}2^n} \approxeq 1.25 \cdot 2^{\frac{n}{2}}$ hashes, with n being the hash length. As an example, random or pseudo-random address generation makes sense in case of IPv6 addresses, where the full host part (64 bits) is used: the probability of having a collision becomes non-negligible only with 5.38×10^9 addresses actively used in the network. However, when the IPv6 is derived from the MAC-16 address as required by 6LoWPAN header compression, collisions are likely to happen even in relatively small networks, i.e., around 300 nodes.

To avoid address collisions, the network coordinator has to avoid all values of the parameter r that will lead to such collisions. The *DODAG Version Number* can be used for this purpose, since it is not mandatory to use consecutive numbers when increasing the DODAG Versions. Using the *DODAG Version Number* has, however, some notable drawbacks, mainly due to the length of this field: only 8 bits. Moreover, the address must not fall in the reserved group of addresses group, and (at least) one bit have to be reserved to recognize the address before and after an address change. As a consequence, the number of available address space is reduced, leading to a higher collision probability.

The reserved address problem is easily solved by letting nodes with a reserved address perform a new AShA evaluation with an extended METAindex, e.g., by having an internal counter concatenated to the *METAindex* such as

$$r = \text{counter} \,||\, \text{METAindex}; \tag{5.7}$$

the counter is increased by 1 until a non-reserved address is found.

Nevertheless, the number of available *DODAG Version Numbers* is very limited, and with a large number of nodes in the network, a collision could be so frequent to leave only a few available *DODAG Version Numbers* to be used.

In order to solve this problem, the DODAG Version Number as the Primary Index is proposed to be used, due to the RPL internal mechanisms ensuring its network-wide spread, and a separate Secondary Index, to be carried either in the unused bits parts of the DIO messages (8 bits) or as a separate RPL Option Header. In the second case it is possible to use a larger Secondary Index. It should be noted that a typical DIO message is quite short (up to 56 bytes, when the IEEE 802.15.4 security is used). As a consequence, there is enough space in a single IEEE 802.15.4 message to carry the Secondary Index. As a matter of fact, an Option Header is suggested to be used for all but the simplest networks because 1) it does not require a non-standard implementation, and 2) it allows to choose the Secondary Index length dynamically.

Due to the fact that upward routes are mandatory, DIO messages to trigger AShA algorithm are used. Note that, in the extreme case of a shortage of viable alternatives, the Network Coordinator can always fallback to unicast messages to resolve collisions. Fig. 5.13 shows the worst-case sequence diagram of AShA algorithm: the DODAG Root evaluates the collisions, chooses the appropriate new values for *Secondary Index* and *DODAG Version Number*, and (eventually) sends a unicast address change message to some nodes. As shown in the figure, the DODAG Root also cooperates with the PAN Coordinator to update its association tables and allows a smooth address transition.

The network downtime during an address reconfiguration is null for a Secondary Index change, because the routing tables can be updated on-the-fly. Moreover, the old addresses can be still used, thanks to the bit used to indicate the “old” and “new” address set (before and after the shuffling). For a Primary Index change, the network downtime is equivalent to the one normally experienced in an RPLnetwork for a DODAGVersion change, and it is usually limited to a few seconds. However, the DODAG Version is periodically increased by RPL to ensure an optimal routing topology, and AShA does not add any further network downtime to this procedure. As a consequence, we can conclude that AShA does not add any downtime to the

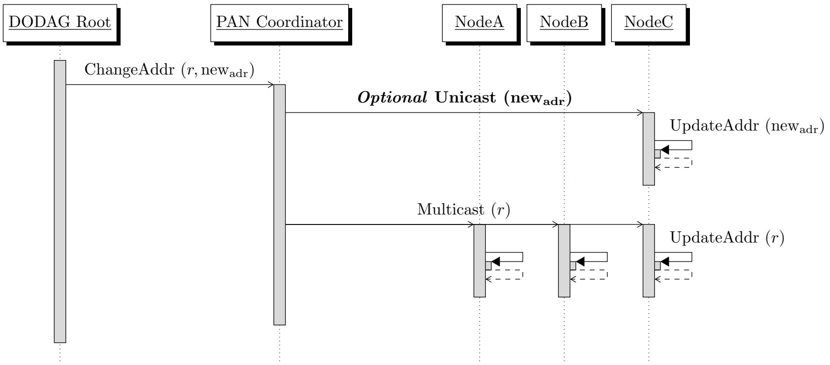

DODAG Root
PAN Coordinator
NodeA
NodeB
NodeC
ChangeAddr (r, new$_{\text{adr}}$)
Optional **Unicast (new$_{\text{adr}}$)**
UpdateAddr (new$_{\text{adr}}$)
Multicast (r)
UpdateAddr (r)

Figure 5.13: Message sequence diagram for AShA address renew procedure.

network.

5.3.4 Simulation Results

To assess the validity of the proposed method, an AShA server is prototyped in Python. With a simulation campaign the number of collisions in a DODAG Version Number sequence is evaluated, i.e., how many DODAG Version Numbers are available depending on the number of nodes in the network. As outlined in the previous section, it is always possible to use unicast messages to resolve the collisions. However, avoiding unicast messages has a number of advantages: no overhead (address renew command is included in RPL DIO messages), less attack footprint (the attacker can not intercept the unicast message), the address change is faster, and no extra energy consumption is needed.

For the sake of simplicity, in the simulations the reserved addresses checks is disabled, and the full 16 bits range has been used.

Fig. 5.14 shows the comparison between using only the DODAGVersion Number (8 bits) or also the Secondary Index (8 + 8 bits). Using only the DODAGVersion Number, collisions happens already with 100 nodes, requiring either to avoid the use of some DODAGVersion Numbers or to use unicast messages to resolve the collisions. On the contrary, the Secondary Index allows a broader range of possibilities, making it possible to use all the DODAGVersion Numbers up to with 700 nodes.

It is evident the perfect agreement between the experimental data and

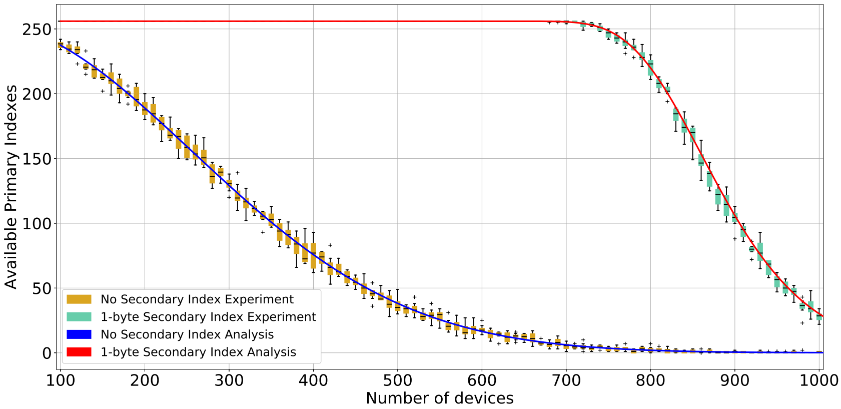


Figure 5.14: Available DODAGversion number with the Primary Index and Primary & Secondary Indexes.

Table 5.1: DODAGVersion Numbers Vs the number of nodes

	Number of nodes	
Available DODAGs	*r = 8*	*r = 16*
2/3 (171)	220	880
1/2 (128)	290	900
1/3 (85)	380	930

the one foreseen by eq. 5.5 and 5.6.

Table 5.1 reports the maximum allowed number of nodes in order to have at least one third, half, and two third of the DODAGVersion Numbers available.

Fig. 5.15 shows the average number of collisions when only the Primary Index is used. As noted before, in case of collisions it is possible to use unicast messages to resolve collisions, but this solution is not energy efficient (it requires extra signaling), and requires extra reconfiguration time.

Fig. 5.16 reports the number of available Secondary Indexes if we just use 1-byte Secondary Index size. It is evident that the curve follows the same distribution shown in eq. (5.5). This is easily explained by considering that, for each Secondary Index, the Primary Index becomes a constant. As a consequence, they can be switched in the formulas.

Fig. 5.17 shows the effect of increasing the Secondary Index length. Al-

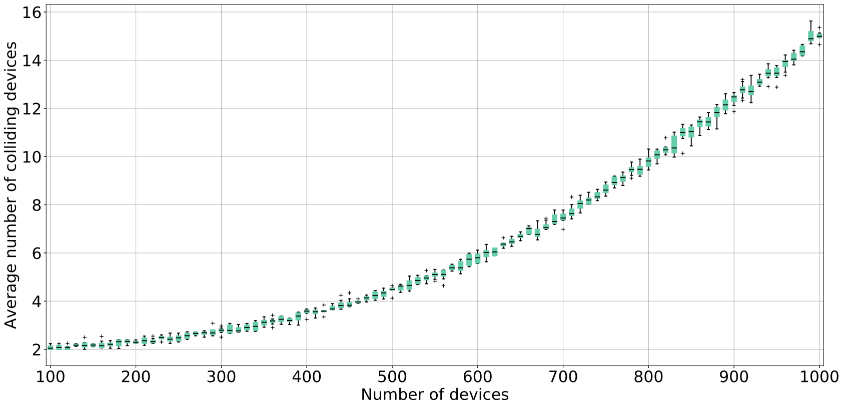


Figure 5.15: Number of colliding nodes using only the Primary Index.

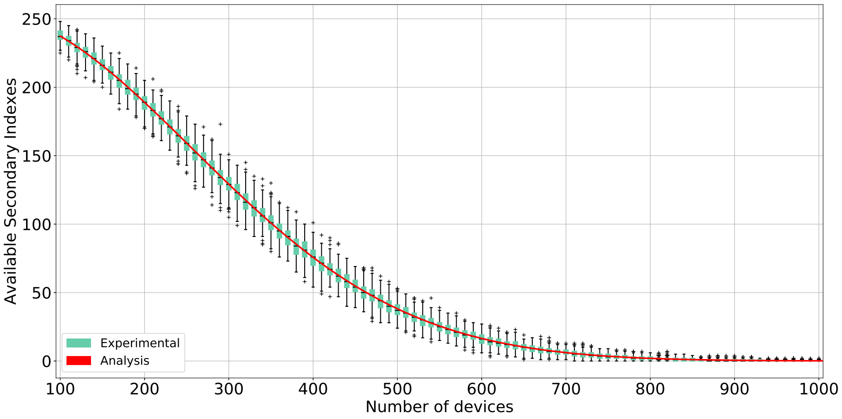


Figure 5.16: Average number of available Secondary Indexes (1 bytes case).

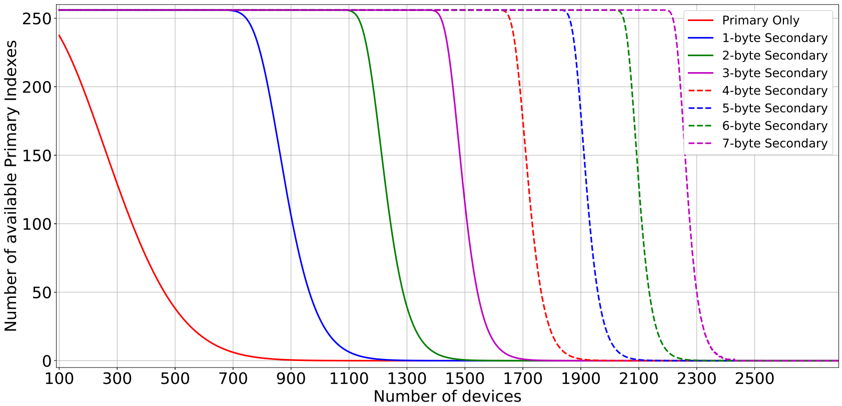


Figure 5.17: Average number of available Primary Indexes varying the Secondary Index length.

though increasing the size of the Secondary Index allows a larger network size, the benefits becomes progressively smaller, and the computational costs for the Network Coordinator becomes larger.

Although it is possible to use any Secondary Index size, it is advisable to choose its correct size according to the actual number of nodes in the network, mainly to conserve energy and storage space.

It is suggested to choose a Secondary Index size that, according to the network size, guarantees an almost complete availability of Primary Indexes (see Fig. 5.17). Toward this end, it is possible to dynamically adjust the Secondary Index length, varying it according to the number of nodes in the network. In this way, it is possible to keep a minimal overhead while maintaining a high Primary Index availability.

In conclusion, in this work Address Shuffling Algorithm with HMAC (AShA) is presented. This algorithm is a novel method allowing a fast, secure, and collision-free address renew in any (IPv6) network. the features and limitations are analyzed throughout an analytical formulation and with simulations, and how there is a match between our analysis and simulation results are shown. The mathematical analysis can be used to predict AShA performances, thus serving as a guideline within the network security planning phase. With respect to the previous work, AShA has the benefits of eliminating any possible information leakage, such as the number of nodes in the network.

AShA is an excellent tool to provide the (network) confusion necessary in Moving Target Defense strategies, especially in IoT scenarios, where energy efficiency is a strict requirement. AShA address renew messages can be piggybacked in normal routing maintenance (or any network management) semi-periodic message, reducing the network overhead to practically zero.

5.4 False Data Detection for Fog and Internet of Things Networks

In this section the work [84] is presented. This work has been published on MDPI - Sensors. This work was realized in collaboration with Politecnico di Milano.

The IoT context brings new security issues due to billions of smart end-devices both interconnected in wireless networks and connected to Internet by using different technologies. In this paper, we propose an attack detection method, named Data Intrusion Detection System (DataIDS), based on real-time data analysis. As end-devices are mainly resources-constrained, Fog Computing (FC) is introduced to implement the DataIDS. FC increases storage, computation capabilities, and processing capabilities, allowing to detect promptly an attack with respect to security solution on the Cloud. This paper also considers an attack tree to model threats and vulnerabilities of Fog/IoT scenario with heterogeneous devices and suggests countermeasures costs. We verify the performance of the proposed DataIDS implementing a testbed with several devices that measure different physical quantities and by using standard data gathering protocols.

5.4.1 Introduction

The IoT advent opens up new vulnerabilities for both security and privacy due to the massive number of resource-constrained devices connected to Internet by using various technologies.

The IoT paradigm is worsening the overall security issues due to the heterogeneity of connected IoT hardware platforms (i.e., different firmware types, revisions, etc.) and to the variety of network technologies for interconnections (e.g., Bluetooth, 802.15.4, NarrowBand IoT (NB-IoT), etc.), all with potential flaws and vulnerability to attacks. An IoT device (*a thing*) can be a light bulb, a thermostat, a smartphone, a personal computer, or potentially everything. IoT devices have to face many threats originated from the Internet, and can also become a source of attacks towards the Internet. Many IoT devices might become easy targets to cyber adversaries due to configuration mistakes, e.g., default password unchanged or unpatched vulnerabilities. A fairly recent example of this issue is the aforementioned malware Mirai.

Due to massive number of interconnected devices and their low power and limited processing power, IoT networks need to share data with Cloud for storage and processing, entailing new security requirements. Fog Computing is a novel paradigm that complements Cloud Computing by moving storage, computation and application services from the Cloud towards the edge of network. This is really useful for IoT applications, as in this way data can be kept local for enabling novel and more efficient security and privacy methods. Therefore, this paper suggests Fog Unit (FU) for supporting a novel Data Intrusion Detection Systems (DataIDS) to detect malicious activities in IoT end-devices.

A typical Fog/IoT scenario is shown in Figure 5.18. IoT devices are organized into *clusters* and each cluster is managed by one or more FUs with higher computational power used to locally collect, store, and process data. The FU acts as a bridge between IoT devices and Cloud, possibly decoupling the IoT-based protocols from the protocols used on the Internet, enabling moreover better energy efficiency.

Usually, the devices and network details are masqueraded by an appropriate abstraction level. However, this also implies that the security layer cannot leverage the intrinsic information of the physical system. Performing some security procedures in the Fog enables to leverage the physical system (e.g., the network topology), along with all the information that are usually not transferred to the Cloud. As an example, sensors produce a high number of data readings but only the data subscribers are informed of the readings, and usually only when a given threshold is reached. The FU, on the opposite, can perform more accurate and prompt analysis of the IoT system behavior, react faster than an equivalent Cloud-based solution, minimize the amount of data that is exchanged on the Internet, and prevent or promptly react to an attack with respect to a security action performed on the Cloud, enhancing the IoT network security and privacy.

In IoT deployments, the standard security mechanisms, such as cryptography and authentication are mandatory. Nevertheless, devices are often vulnerable to a broader attack range, due to the particular attack surface (e.g., large number of devices, installation in non-monitored environments, resource contains leading to weaker cryptography, etc.). As a consequence, IDSs are needed.

Usually, IDSs analyze network traffic patterns, packets content or systems logs, searching for the evidence of security violations mainly at the network layer (e.g., for routing attacks). Sadly, a large class of attacks targeting the IoT data cannot be easily detected by traditional IDSs.

In this section, we propose a novel IDS, named DataIDS, specifically designed for Fog/IoT networks based on the analysis of physical (sensed) data to better recognize vulnerabilities against the end-devices. The measurements carried out by sensors are sent to the FU, which locally processes the data streams, and, if an anomalous behavior is detected, it can raise an alarm and manage appropriate countermeasures, e.g., to isolate the devices under attack, or discard their data or authenticate a sensor and its data, or to reconfigure the IP-addresses.

The DataIDS distinguishing features are i) the ability to detect a malicious (or false) data injection by analyzing the datastreams acquired by the devices, and ii) at the same time to find the devices which are currently misbehaving. The key idea is to build a *dependency graph* by analyzing the cross-correlation among the respective data streams of sensors, and to use that information to highlight any anomaly in the system. This allows to react promptly to a threat with the appropriate actions, and/or to trigger further analysis mechanisms aimed at verifying the sensor health conditions. It is worth mentioning that DataIDS can be easily integrated into Fog nodes without significantly impact on their performance, by enabling a Fog node to control a very large number of IoT devices and to raise an alarm and related countermeasures if one or more devices are under attack.

To complement the DataIDS approach, we propose a novel attack tree with associate risks, costs, and level of potential system damage. According to the detected threats, the attack tree is a valid method to select the appropriate action to be undertaken, which can span from simply discarding the data of attacked sensors to a full network reconfiguration.

We implemented a testbed to validate the proposed DataIDS performance on real datasets acquired with several sensors measuring different physical quantities when different data injection attacks occurred, such as stuck-at,

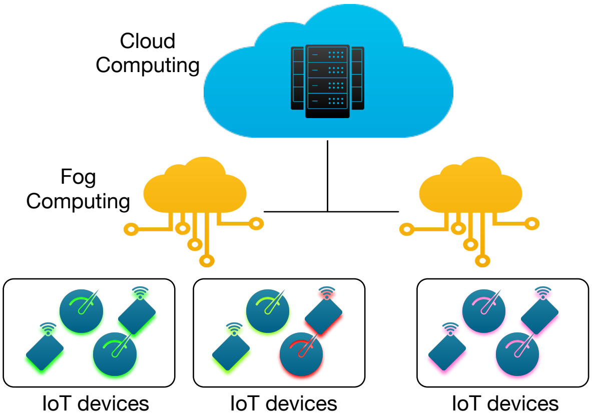


Figure 5.18: Fog and IoT scenario

replay and sensor replacement.

5.4.2 Related Works

A Fog/IoT system is subjected to the attacks both from the Internet and from within the wireless sensor network, therefore firewalls to isolate the sensitive part of the network and IDS to detect attacks are needed.

An IDS is a software tool [55, 150] that collects and analyzes input data coming from a network, in order to find possibles security breaches. Usually IDSs are classified in two categories:

- *Signature detection system.* The possible intrusions are identified through traffic patterns and/or predetermined attack signatures. The main benefit of this technique is the high detection reliably. On the other hand, the signature of each known attack should be stored with significant storage and computational costs increasing with the number of attacks. Moreover, the attack signatures database must be always up-to-date.

- *Anomaly detection system.* The IDS compares users behaviors with a model. If the behavior differs from the model, an alarm is raised. It can detects unknowns attacks (the so called *zero-day*) but it requires the definition of the model of the *normal* system behavior.

Another IDS classification can be based on the type of data monitored by the IDS: a Network-based IDS (NIDS) analyzes network traffic, while a Host-based IDS (HIDS) monitors a computer (its running programs, application logs, etc.) [191]. The two types can be also used jointly, in order to provide a comprehensive networked system protection.

Surveys on different IDS types can be found in [90, 150, 187, 191] highlighting that IDSs mainly work by analyzing log files and/or network traffic patterns. Moreover, most of them are not specifically designed for IoT.

Applying a NIDS to the IoT scenario raises some noteworthy issues, like the number of traffic flows to be analyzed, and the need to collect traffic from multiple network points, which can be extremely costly in a multi-hop network. Moreover, the traffic pattern is not suitable for anomaly-based IDS, due to the huge differences in traffic patterns in case of particular events (e.g., a sensor might increase suddenly its sampling rate depending on the environment it is controlling). HDISs are not suitable either, due to the limited sensor computational and energy resources. It is possible to add mechanisms to prevent firmware tampering but it is not a common solution for commercial systems. Furthermore, NDISs and HDISs cannot detect a wide range of attacks highlighted in the attack tree in Figure 5.19, and in particular the attacks targeting (or consequence of) a change in the physical world, i.e., environment modifications (e.g., placing a heat source near a sensor), modifications to the device hardware components, etc.

In literature, the IDSs designed for IoT mainly consider attacks at network layer (usually routing attacks). Examples are SVELTE [155], used to detect sinkhole and selective-forwarding attacks, CEP [77], able to analyze the information streams to detect events in real-time, or the approach presented in [194] able to detect Denial of Service Attacks (DoS) targeting the Routing Protocol for Low Power and Lossy Networks (RPL).

The authors in [101] propose an algorithm based on four phases, i.e., *initialization*, *estimation*, *similarity check*, and *characterization*. During the first phase an estimation model is produced and a similarity check is defined. The second phase, that is the core of the overall system, extracts and iteratively aggregates the estimates of the measurements (following the information defined in the first step) that are then sequentially analyzed by two different tests. When a change is detected, the *characterization* phase is activated to identify the compromised sensor. This solution encompasses only a linear fixed model among acquired measurement and is applied only to homogeneous measurements (hence gathered by the same type of sensors). Rather, our proposed DataIDS can work on different heterogeneous measurements.

An IDS for the detection of malicious data injection based on wavelet transform is proposed in [102]. Even in this case the algorithm is dived into three phases: *detection*, *characterization* and *diagnosis*. In the first phase an anomaly score based on the wavelet coefficient is sequentially analyzed over time inspecting for changes by means of a thresholding mechanism. When a change is detected, the next characterization and diagnosis phases are activated. Such solution focuses only on the spatial correlation not exploiting the temporal correlation present in the acquired data as in the detection phase of our proposed DataIDS. In addition, such solution requires the knowledge of the conditions during the "event" target and relies on the information about the position of the nodes.

In [149], the authors present an anomaly behavior analysis IDS able to detect attacks in smart home system. This framework builds sensor profiles by using the Discrete Wavelet Transform method on the sent data and the euclidean distance (ED) is utilized for the comparison with the reference profiles, obtained during the offline training phase, to detect abnormal behaviors. As our proposed algorithm, these processes are performed in runtime. The major difference with our DataIDS is the learning phase: DataIDS does not need to know data nature provided during training because the *dependency graph* could have measurements of different types (for example, humidity and temperature). Then, this leads to have a more flexible system in the monitoring phase: the devices must monitor that their behavior is consistent with the other members of the dependency graph (if a sensor evaluates a change and the other ones the same, it means that it is really the environment changing).

5.4.3 Attack Tree and Attack Models

A threat model, and the associate risk management help to find security policies and countermeasures that could prevent an attack or mitigate its outcomes [137]. As a matter of fact, without a proper threat model, the system security cannot be guaranteed, because some threats could be underestimated or, on the opposite, some threats could be overestimated, leading to unnecessary security restrictions and extra costs.

A successful risk management process has also to balance the cost of security techniques and the system usability for each potential attack. Therefore, the optimal security system is the one where the implementation does not become more expensive than the possible damage of the attack that is trying to prevent.

Attack tree

We consider the attack tree to model the possible threats and vulnerabilities of our system. The term attack tree was introduced by Schneier in [159] and represents a tool to evaluate the effectiveness of an attack and appropriateness of a countermeasure, depending on the attack type and extent. An attack tree describes the possible attacks to the network system through a graphical tree structure where the root node is the target of the attacker (the goal) and the leafs are all the possible (and impossible) means to compromise the target (i.e., the attacks) [127]. It is worth noticing that several roots (targets) might exist in the same system. In this case multiple attack trees must be considered.

Building an attack tree consists of four main steps:

1. Define the main attack goal.
2. Decompose the main attack goal into sub-targets.
3. Assign values to the leafs.
4. Calculate the cost of an attack.

The values assigned to the attack tree leafs can represent different properties of the attack, and they can be boolean or continuous on a specified range. As an example of boolean properties, we can list if the attack is easy, if it is expensive, if particular skills of the attacker are required, etc. Continuous values can represent the attack cost, its likelihood, the time required to perform the attack, etc. Moreover, if more than one condition must be fulfilled to perform an attack, nodes can be connected, e.g., in case of an attack that could be exploited only after a different one has been performed. The resulting values can be used to make assumptions about the attack and the attacker, i.e., to build the threat model.

The attack tree evaluation is helpful in the risk management because if an attack is easy or the cost is low, its occurrence is likely, or if the cost of countermeasures is much higher than the attack outcomes, the attack can be ignored.

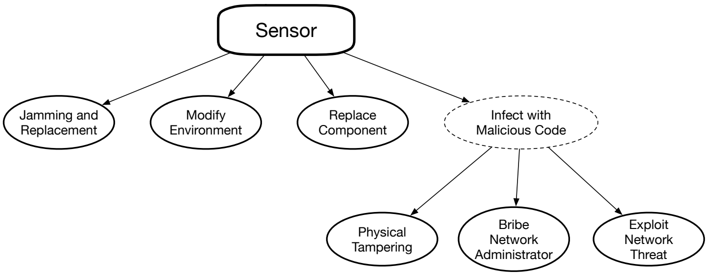


Figure 5.19: Attack tree for our IoT system.

Table 5.2: Attack detection comparison

	Traditional IDS	DataIDS
Jamming & Replacement	Difficult	Yes
Modify Environment	No	Yes
Replace Component	No	Yes
Physical Tampering	No	Yes
Bribe Network Administrator	No	Yes (if data are modified)
Exploit Network Threat	Yes	Yes (if data are modified)

The attack tree for our Fog/IoT system is shown in Figure 5.19. We only highlighted the possible attacks to the IoT domain without consider the well-known vulnerabilities of FU and gateway.

Looking at Figure 5.19, we notice that some attacks can be detected by "traditional" system, such as IDS, logging programs, etc., but some attacks are specific, and they do not leave any trace in the parameters analyzed by the techniques mentioned above, as explained in [103]. Therefore, we need a technique to detect possible attacks by analyzing alternative parameters, such as *data measurements* sent by sensors as in the proposed DataIDS. The advantages of our approach is summarized in Table 5.2.

The sensors can be classified according to the importance of the sensed value (e.g., if the reading cannot be inferred from other sensors, if the reading is particularly critical for the IoT application, etc.) or the topology of the network (e.g., if the sensor node acts as a router in a multi-hop topology).

The threats have to be analyzed according to their likelihood and damage factors. In the damage and attack costs, we estimate respectively the cost of the countermeasure and the difficulty for an attacker to execute successfully

Table 5.3: Damage and Attack costs

Damage	Routing tree position Number of nodes (Cluster) under attack Dependency graph position Data importance
Attack	Cost to find the attack Time required for the attack Equipment cost Skill required Physical access to the nodes Attack reproducibility

a particular attack.

To evaluate the damage cost, we consider the following factors:

- The node position in the routing tree: the damage cost is different if a node is a leaf or closer to the root;
- The node position in the dependency graph in our DataIDS;
- The number of nodes under attack (i.e., the cluster in the dependency graph);
- The importance of the data damaged;
- The time and signaling required to perform a countermeasure.

Attack cost is more difficult to evaluate because we must consider some features that are unknown a-priori, such as the time needed to perform the attack, the required skills, and the cost to buy a particular equipment. All these elements are strictly dependent on the particular IoT device vulnerability and hardware availability. However, we can assume that the hardware needed for the attack is affordable (sensors are low-cost normally), while for the time and skills we expect high costs, because we can assume that the device firmware does not contain simple and easily exploitable flaws. The damage and attacks are summarized in Table 5.3.

According to the damage and the attack costs, we can accept the risk or take the proper countermeasures to mitigate the attack. As an example, we can accept the risk when the attack and countermeasure costs are high but the damage cost is low. In case of a likely attack, we must either apply a countermeasure or increase the attack cost, e.g., by removing the vulnerabilities that lead to that particular attack.

As an example, if in a Fog/IoT network there are several temperature sensors and only one is under attack, we can evaluate if we can accept the risk that the attack propagates and simply apply a low cost countermeasure by discarding the data from the device under attack or isolate the node and re-authenticate it. Instead, if the device is a central node which routes data towards the FU, we need, e.g., to apply network reconfiguration with higher time and energy costs [57]. Therefore, the countermeasure must be correlated to the attack according to the assessed risk outcomes.

Attack Models

Let us consider an IoT system composed by N IoT units $U = \{u^1, \ldots, u^N\}$, each endowed with one sensor.

Without loss of generality, we assume that units in U are synchronous, i.e., at each time instant t is created a vector of scalar measurements $\mathbf{x}_t = \{x_t^1, \ldots, x_t^N\}$ with $x_t^i \in R_{x^i} \subset \mathbb{R}$, $i \in \{1, \ldots, N\}$, and R_{x^i} is the range of allowed values from sensor i. This assumption can be relaxed by using appropriate data processing techniques (e.g., interpolation, re-synchronization, etc.). We do not make any assumption about the process generating the data stream $\mathbf{x}_t$, which is considered unknown a-priori. We emphasize that we are not assuming the stationary of $\mathbf{x}_t$ that might evolve following the dynamic of the physical phenomenon monitored.

It is worth stressing that differently from the literature where the homogeneity or monotonicity assumption is considered [150], [101], in our work units in U might be weakly or strongly related to each other, i.e., sensors can be heterogeneous (they measure different physical quantities, e.g., temperature and humidity) .

We only assume that our Fog/IoT system initially behaves in attack-free situations; an attack might occur only later during the system lifetime. This assumption reflects the fact that an attack requires some time to be performed, and that we can assume that the system is behaving as intended at the beginning of our modeled period.

We consider the case where a subset U_A of units, with $U_A \subset U$, could be gained by an attacker, modifying data coming from units in U_A as follows:

$$x_t^j = \begin{cases} x_t^j & t < t_j^* \\ f_{\theta_j}\left(x_t^j\right), & t \geq t_j^*, \end{cases} \tag{5.8}$$

where $u_j \in U_A$, $f_{\theta_j}\left(x_t^j\right)$ models the (possibly time-variant) perturbation affecting u_j, and t_j^* is on-set attack time of u_j.

We model four different type of malicious data injections, i.e., stuck-at, replay and, sensor replacement that is dived in two sub cases noise addition and dynamic perturbation attacks to IoT units:

- *Stuck-at:* the attacker gains access to unit u_j at time t_j^* and replaces the values $x_t^j, t \geq t_j^*$, with the constant value $x_{t_j^*}^j$, i.e.,

$$f_{\theta_j}\left(x_t^j\right) = x_{t_j^*}^j, \quad t \geq t_j^*; \tag{5.9}$$

- *Replay:* the attacker gains access to unit u_j at time t_j^* and replaces the values $x_t^j, t \geq t_j^*$ with the data acquired up to time t_j^*, i.e.,

$$f_{\theta_j}\left(x_t^j\right) = \Pi\left(t, t_j^*\right), \quad t \geq t_j^*, \tag{5.10}$$

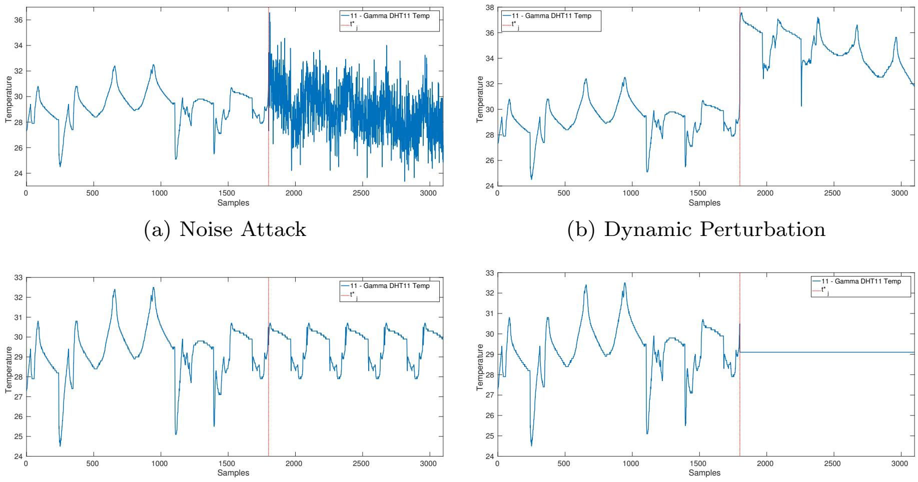
  where $\Pi\left(t, t_j^*\right)$ models the repetition at time t of data acquired before time t_j^*;

- *Sensor Replacement (Noise addition):* the attacker gains access to unit u_j at time t_j^* and introduces a random perturbation to the values $x_t^j, t \geq t_j^*$, i.e.,

$$f_{\theta_j}\left(x_t^j\right) = x_t^j + \eta_j, \quad t \geq t_j^*; \tag{5.11}$$

  where η_j is an independent and identically distributed random variable accounting, e.g., for an additional noise affecting the original measurement x_t^j;

- *Sensor Replacement (Dynamic Perturbation):* the attacker gains the access to unit u_j at time t_j^* and perturbs the values $x_t^j, t \geq t_j^*$ by



(a) Noise Attack

(b) Dynamic Perturbation

(c) Replay Attack

(d) Stuck-at Attack

Figure 5.20: Examples of the attack models at sampling time $t_j^* = 1800$ acquired in our testbed

modifying the signal dynamic, i.e.,

$$f_{\theta_j}\left(x_t^j\right) = (1+\delta)\cdot x_t^j, \quad t \geq t_j^*; \tag{5.12}$$

where $\delta \in \mathbb{R}$ accounts for the magnitude of the perturbation.

Figure 5.20 shows an example for each of these four types of considered attacks. For instance, those attack types would be realized by an attacker if he substitutes or modifies the code installed in the sensors with a malicious one.

Our goal is to analyze the datastreams $\mathbf{x}_t$ to promptly identify and isolate an attack affecting U.

5.4.4 Data IDS

The idea of the proposed attack detection and isolation mechanisms is to characterize the relationships existing among the acquired datastreams and analyze them over time looking for an unexpected behavior of one node. In more details, the proposed algorithm relies on an initial data sequence D_S

storing the first S samples acquired by all the sensors $u_i \in U$, i.e.,

$$D_S = \begin{bmatrix} x_1^1 & \cdots & x_S^1 \\ \vdots & \ddots & \vdots \\ x_1^N & \cdots & x_S^N \end{bmatrix}. \tag{5.13}$$

D_S represents the measurements (e.g., temperature or humidity) acquired in an initial attack-free situation and it is partitioned into training set T_S and validation set V_S, where T_S stores the first T samples acquired by all the sensors and V_S the remaining $S-T$ samples, i.e.,

$$T_S = \begin{bmatrix} x_1^1 & \cdots & x_T^1 \\ \vdots & \ddots & \vdots \\ x_1^N & \cdots & x_T^N \end{bmatrix}, \quad V_S = \begin{bmatrix} x_{T+1}^1 & \cdots & x_S^1 \\ \vdots & \ddots & \vdots \\ x_{T+1}^N & \cdots & x_S^N \end{bmatrix}. \tag{5.14}$$

T_S is used to learn the relationships among the sensors in U by analyzing the cross-correlation between the respective data streams $\mathbf{x}_t$. To achieve this goal we rely on the concept of *dependency graph* [43] that has been introduced to capture and model the relationships among sensors. A dependency graph is an undirected graph $G = \{\mathcal{N}, \mathcal{E}\}$, where nodes $\mathcal{N}$ represent the N sensors in U and edges $\mathcal{E}$ represent the relationships between couples of sensors. In our specific case the edge $e_{i,j}$ between u_i and u_j exists in $\mathcal{E}$ when

$$r_{i,j}^T > \gamma \tag{5.15}$$

where $r_{i,j}^T$ is a *cross-correlation index* measured as the normalized absolute value of the peak of the cross-correlation between the data sequences $\{x_1^i, \cdots, x_T^i\}$ and $\{x_1^j, \cdots, x_T^j\}$, and $\gamma \in [0,1]$ represents the user-defined threshold value for the cross-correlation index. Such a value has a statistical interpretation representing the minumum value of the cross-correlation between two datasequences to create an edge in the dependency graph. γ could range ranges from 0.8 to 0.99 (i.e., from 80% to 99%). In the experimental analysis described later, γ has been set to 0.9.

The idea of using the cross-correlation resides in the ability to define an index, i.e., a scalar index bounded between -1 and 1, characterizing the relationship between the functional behaviors of two data streams. In such a way, we can move the analysis between data streams to the analysis of a scalar value for each couple of data streams.

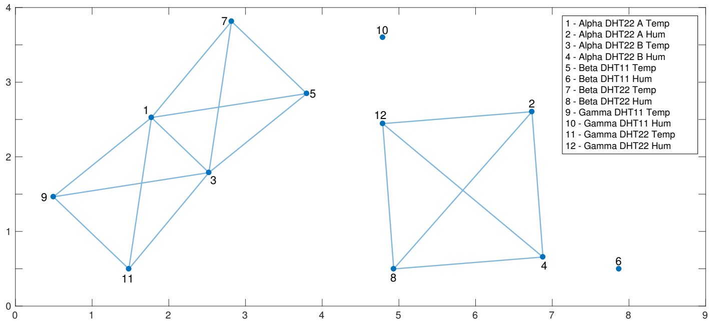


Figure 5.21: The estimated Dependency Graph for the considered testbed.

An example of a dependency graph, built from our testbed, is shown in Figure 5.21. The edges are particularly interesting, because they show that temperature sensors (i.e., nodes u_1, u_3, u_5, u_7, u_9, u_{11}) are related each other, ant the humidity sensors (i.e., nodes u_2, u_4, u_8, u_{12} related to DHT22 sensors in testbed) exhibit a similar behavior. Differently, other humidity sensors (i.e., nodes u_6, u_{10} and DHT11 sensors in the testbed) are not related with any of the sensors. Moreover, we do not detect relationships in the dependency graph between temperature and humidity sensors even if they are close in position. This validates the accuracy of our framework also in terms of heterogeneity among the sensors. We emphasize that the physical position of sensors is not considered in the building of dependency graph that only comprises the information content present in data, i.e., two physically-close sensors are connected through an edge in the dependency graph only if they are cross-correlated according to Eq. (5.15). For example, if the attacked sensor is the number 11, it is placed in the same position of number 9 (and with sensors 10 and 12), as shown in Figure 5.24, but in the dependency graph (in Figure 5.21) it is also related with 1 and 3, that are in another location[5].

This is an example of dependency graph derived for the testbed with a limited number of sensors, however the dependency graph shows the relationships of each device with the other ones and does not depend on the number of sensors considered. If there is a high number of sensors, it is possible to expect an increased calculation time to build the dependency graph

[5] We emphasize that the absence of an edge connecting sensors 6 and 10 does not mean that those two sensors are not related to the other sensors. It means that the cross-correlations they have with the other sensors is below the threshold value γ.

at the beginning.

Attack detection

Once the dependency graph has been computed, FU monitors the relationships of each sensor with the "most related" sensors (as shown in the dependency graph) over time searching for changes. It checks for changes in the cross correlation index only for the sensors connected with the considered sensor in the dependency graph (i.e. the most 'related' in terms of cross-correlations index). Here, changes refer to attack perturbing the acquired data streams as previously shown.

In more detail, let C_i be the set of sensors connected to u_i according to the dependency graph. At each time instant $t > S$, the following *change-detection index* is calculated:

$$R_i(t) = \sum_{j \in C_i} r_{i,j}^{t,W}, \tag{5.16}$$

where $r_{i,j}^{t,W}$ is the cross-correlation index defined above computed over the last W recently acquired samples $\{x_{t-W+1}^{i}, \cdots, x_t^i\}$ and $\{x_{t-W+1}^{j}, \cdots, x_t^j\}$ coming from sensors u_i and u_j, respectively.

A detection occurs when

$$R_i(t) < \Theta_d^i \tag{5.17}$$

where Θ_d^i is an automatically-computed threshold defined as

$$\Theta_d^i = M^i - \lambda_d (M^i - m^i) \tag{5.18}$$

and

$$M^i = \frac{1}{S-T} \sum_{t=T+1}^{S} R_i(t), \tag{5.19}$$

$$m^i = \min\{R_i(T+1), \ldots, R_i(S)\}, \tag{5.20}$$

and $\lambda_d > 1$ is a user-defined parameter representing a confidence parameter for the detection phase acting as a multiplier coefficient in computing the threshold for detecting changes. The larger λ_d, the smaller the threshold, i.e., a smaller threshold would reduce false positive detection but at the expenses

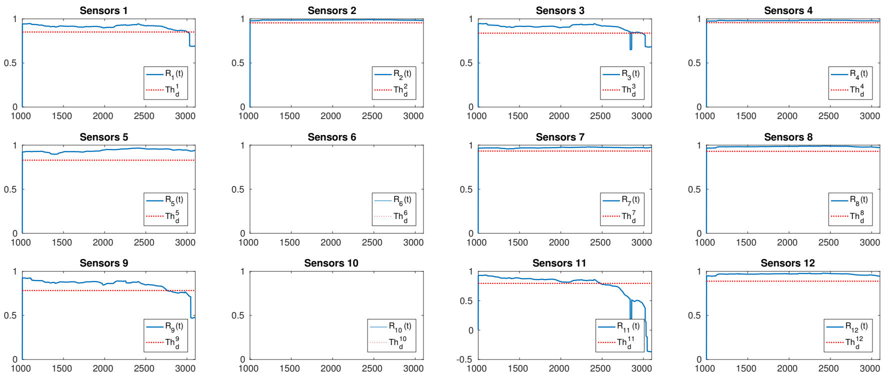


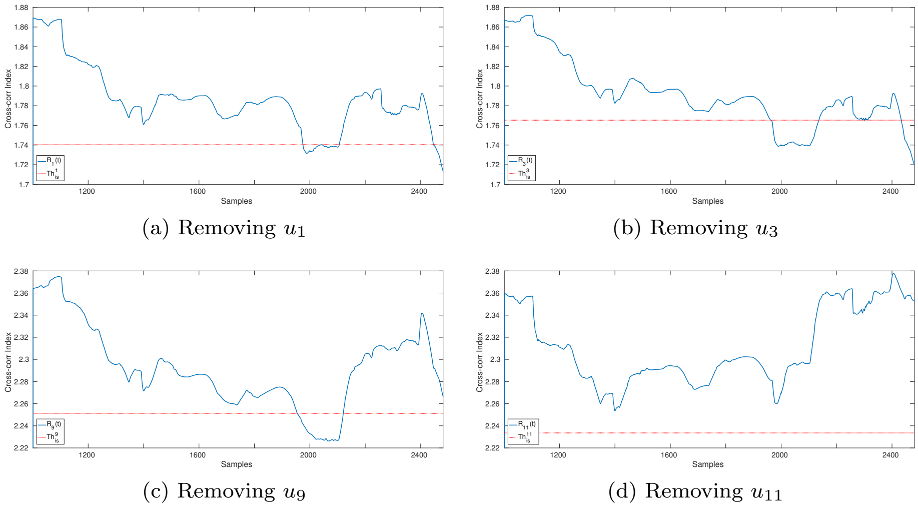
Figure 5.22: An example of change detection analysis carried out on cross-correlation indexes $R_i(t)$s of all the sensors when the attacked sensor is u_{11}. The first $S = 1000$ samples belong to the training sequence. In each subfigure, on the x axis we have the samples and on the y axis we have the cross-correlation index. In the legends the symbol $\mathrm{Th}_\mathrm{d}^\mathrm{i}$ corresponds to $\Theta_\mathrm{d}^\mathrm{i}$ in the text.

of the (possible) increase of false negative detection and detection delays .

An example of detection is given in Figure 5.22. Here, the attacked sensor is u_{11} at time $t_j^* = 1800$ and, as expected by looking at the dependency graph in Figure 5.21, the cross-correlation indexes $R_i(t)$s computed in sensors u_1, u_3, u_9 and u_{11} perceived a change. The cross-correlation indexes $R_i(t)$s in the other sensors do not exhibit changes.

From the technological point of view the IoT devices are constrained in memory, computation and energy and, for this reason, the detection step is carried out by the FU layer where more computation and storage resources are available. The use of FU leads to less network overhead, but it also increases the attack risks (if the same FU is attacked). However, if the IoT devices have processing capabilities to calculate cross-correlation indexes, the change detection phase can be implemented in a distributed way directly at the IoT devices, under the assumption that the sensors connected in the dependency graph can exchange the acquired information. Hence, each sensor in U monitors its own $R_i(t)$ over time and the first sensor detecting a change according to Eq. (5.16) raises an alarm and activates the next isolation phase.

We emphasize that, in both cases, the change detection phase is carried



(a) Removing u_1

(b) Removing u_3

(c) Removing u_9

(d) Removing u_{11}

Figure 5.23: An example of isolation when the attacked sensor is u_{11}. The detection has been raised by sensor u_9 at time $\widehat{t} = 2481$. The first 1000 samples belong to the training sequence, i.e., $S = 1000$. In the legends, the symbol $\mathrm{Th}^{\mathrm{i}}_{\mathrm{is}}$ corresponds to $\Theta^{\mathrm{i}}_{\mathrm{is}}$ in the text.

out for sensors for which $C_i \neq \emptyset$, i.e., the sensor must be related to at least one of the other sensors in U according to the dependency graph. When $C_i = \emptyset$, as in the case of sensors u_6 and u_{10}, the analysis based on cross-correlation cannot be considered, and one could resort on change-detection analysis based on the inspection of the residual between the output of a suitably-trained prediction model (e.g., linear input-output models or recurrent neural networks) on the acquired data (see for example [42]).

Attack isolation

Once an attack has been detected on an IoT device in U, the isolation procedure is activated to identify the device representing the target of the attack. We emphasize that, thanks to the analysis of cross-correlations, the sensor with the changing in the cross-correlation index could not be the attacked one (and, in most of the cases, the attacked sensor could be interested in not raising an alarm at all).

Being $u_{\widehat{i}}$ the sensor with the cross-correlation changing at time $\widehat{t}$, the isolation procedure analyses acquired data from $\widehat{C} = \{u_{\widehat{i}} \bigcup C_{\widehat{i}}\}$, being $C_{\widehat{i}}$ the set of sensors connected to $u_{\widehat{i}}$ according to the dependency graph, up to time $\widehat{t}$ to identify the attacked IoT device. The isolation procedure must run on the FU, being able to store high amount of data and execute more computational-demanding procedures.

More specifically, the isolation procedure removes one IoT device at a time from $\widehat{C}$ and analyses the behavior of the cross-correlation among the remaining sensors, i.e.,

$$\widehat{R}_i(t) = \sum_{j,k \neq i \in \widehat{C}} r_{j,k}^{t,W}, \quad t = S+1, \ldots, \widehat{t}, \tag{5.21}$$

for all $i \in \widehat{C}$. The cross-correlation isolation index $\widehat{R}_i(t)$ for $t = S+1, \ldots, \widehat{t}$ is inspected looking for changes by relying on an automatically-computed threshold defined as

$$\Theta_{is}^i = M^i - \lambda_{is}(M^i - m^i) \tag{5.22}$$

where $1 < \lambda_{is} < \lambda_d$ is a user-defined parameter. Once we remove sensor u_i from $\widehat{C}$ and compute $\widehat{R}_i(t)$, two different situations arise:

- $\widehat{R}_i(t) < \Theta_{is}^i$ for $t = S+1, \ldots, \widehat{t}$: the set of sensors $\widehat{C} - u_i$ includes the attacked sensor since the cross-correlation isolation index still shows a decreasing behavior (revealing that the sensor providing perturbed behavior is still in $\widehat{C} - u_i$);
- $\widehat{R}_i(t) > \Theta_{is}^i$ for $t = S+1, \ldots, \widehat{t}$: the sensor u_i can be safely considered the target of the attack, since its removal from $\widehat{C}$ prevents the decreasing of $\widehat{R}_i(t)$, meaning that the considered data sequences still exhibit the expected behavior.

In addition, once the attacked sensor $\hat{u}_i$ has been isolated, the isolation procedure also computes an estimate $\hat{t}$ of the time instant t_j^* the attack occurred. $\hat{t}$ is computed by averaging the largest time instant for which $\widehat{R}_i(t) \leq \Theta_{is}^i$ for all the sensors $\widehat{C} - \hat{u}_i$.

An example of isolation is given in Figure 5.23, where the attacked sensor is u_{11}. The detection has been raised by sensor u_9 at time $\widehat{t} = 2481$. Here, $\widehat{C} = \{u_1, u_3, u_9, u_{11}\}$ and no detection occurs for $\widehat{R}_{11}(t)$, $t = 1001, \ldots, 2481$ meaning that u_{11} is the attacked sensor. Conversely, $\widehat{R}_1(t)$, $\widehat{R}_3(t)$, and $\widehat{R}_9(t)$ raises a detection before $\widehat{t} = 2481$.

Table 5.4: DHT 11 - DHT 22 details.

	Max sampling rate	Type	Readings interval	Accuracy
DHT 11	1 Hz	Humidity	$(20 \div 80)$ %	5 %
		Temperature	$(0 \div 50)$°C	±2°C
DHT 22	0.5 Hz	Humidity	$(0 \div 100)$ %	$(2 \div 5)$ %
		Temperature	$(-40 \div 80)$°C	±0.5°C

When none of the IoT devices in $\widehat{C}$ revealed to be the target of the attack, i.e., no detection occurs in any of the sensors in $\widehat{C}$, our isolation procedure is not able to isolate the attacked sensor. In this case, a general “attack alarm” message is raised signaling that one of the sensors in $\widehat{C}$ has been attacked.

We emphasize that the proposed isolation procedure implicitly assumes that only one IoT device in $\widehat{C}$ has been attacked. If multiple sensors are under attack, this assumption can be weakened by forcing the dependency graph to create clusters of IoT devices characterized by smaller cardinalities to isolate the only one sensor compromised.

It is worth noting that this approach is not able to divided attacks from faults, but if we implement the relative countermeasures and the algorithm is triggered very quickly it is certainly a sensor fault.

5.4.5 Experimental results

In this section, we describe the testbed used to validate the methods presented.

Description of the testbed

The testbed at the University of Florence is made by 6 DHT11 - DHT22 devices by Aosong, generating 12 independent humidity and temperature data streams. DHT11 and DHT22 are low cost environmental devices and are made of two parts: a thermal resistor and a capacitive humidity sensor. Details about the used sensors are summarized in Table 5.4. Each sensing device is connected to a Raspberry Pi3 which is responsible for data collection.

We want to emphasize that our measurements are obtained in a real context. The devices are placed in different positions characterized by unevenly distributed air conditioning system. As a result, the readings are expected

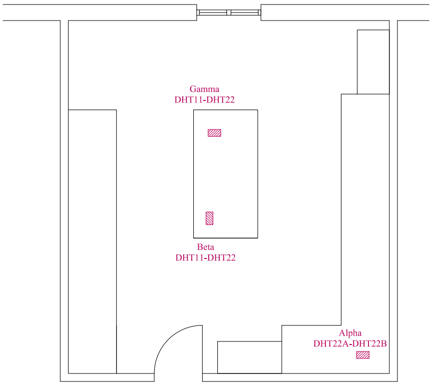


Figure 5.24: Laboratory floor map and devices positions.

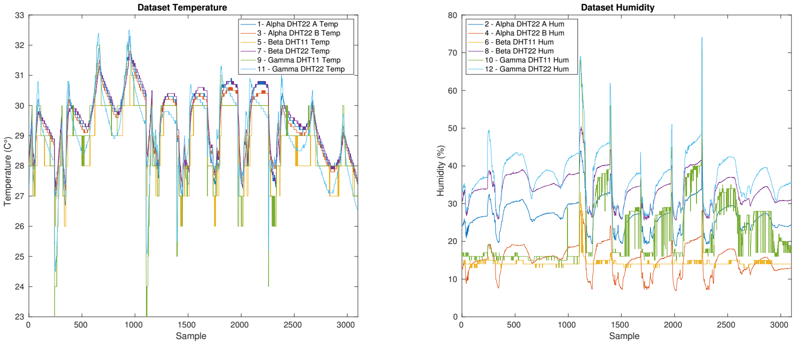


Figure 5.25: Measured temperature and humidity dataset.

to be similar, but not identical. The assignment (i.e., name sensor, type sensors) is summarized in Table 5.5, and the map of the Raspberry Pi3 positions is shown in Figure 5.24.

The configuration of the system and the dataset (about 10430 samples) acquired every 5 minutes are available at the link `https://www.gaucho.unifi.it`. The dataset is shown in Figure 5.25.

Description of the considered attacks

- *Stuck-at* where $x_t^j = x_{1800}^j, t \geq t_j^*$;

Table 5.5: Testbed sensors names and type.

Pi3	Sensors	Number	Type
Alpha	DHT22 A	#1	Temperature
		#2	Humidity
	DHT22 B	#3	Temperature
		#4	Humidity
Beta	DHT11	#5	Temperature
		#6	Humidity
	DHT22	#7	Temperature
		#8	Humidity
Gamma	DHT11	#9	Temperature
		#10	Humidity
	DHT22	#11	Temperature
		#12	Humidity

- *Replay* where $\Pi\,(t, 1800)$ replaces $x_t^j, t \geq t_j^*$ with values acquired in the 24 hours before $t_j^* = 1800$;
- *Sensor Replacement (Noise addition)* where $\eta_j = N(0, 1.5)$ is a Gaussian random variable with zero mean and standard deviation equal to 1.5;
- *Sensor Replacement (Dynamic Perturbation)* where the magnitude of the perturbation is $\delta = 0.2$.

Every attack starts at $t = 1800$. Each attack is repeated on all the sensors (one at a time), for a total of 48 experiments.

Figures of merit

In order to evaluate the effectiveness of the proposed algorithm we defined the following seven figures of merits:

- *Attack Detected*: binary value describing whether the attack has been detected (1) or not (0);
- *Detection counter*: number of sensors within the network that detected an attack (excluding the attacked sensor);
- *Min Detection Time*: when the first sensor detected an attack within the network;

- *MW Detection Time*: when at least half the sensors connected to the sensor under the attack detect the attack;
- *Max Detection Time*: when the last sensor detected an attack within the network;
- *Isolated Attack*: binary value describing whether the attack has been correctly isolated (1) or not (0);
- $\hat{t}$: estimation time.

Experimental results

The experimental results are summarized in Table 5.6. The four types of cyber-attacks described in Section 5.4.5 have been applied to all the IoT devices in the testbed, i.e., #1 – #12. In this experimental analysis the parameters of the proposed solution have been set as follows: $\gamma = 0.9$, $\lambda_d = 4$ and $\lambda_{is} = 1.4$.

The cyber-attacks have been detected in 97.5% of the cases, i.e., 39 of "Detected Attacks" over 40 experiments. We emphasize that we did not experience false positive detection, equivalent to a *precision* of 100% and to a *recall* of 97.5%. The isolation capability worked in 82.5% of the cases, i.e., 33 of "Isolated Attacks" over 40 tests, which is still a very good result. We emphasize that, in those cases where a correct isolation is not achieved, the proposed solution is not able to isolate an attacked sensor. Hence, in the considered experimental analysis, the proposed solution is either able to correctly isolate the attacked sensor or it does not provide any isolation (i.e., we do not isolate wrong sensors). This lead to a precision of 100% and to a *recall* of 82.5%

For what concerns the sensibility of the proposed methods to the attack type, it is worth observing that the *Replay*, *Stuck-at*, and *Dynamic Perturbation* attacks have been correctly detected and isolated with extreme high accuracy (respectively 100% detection and 96.7% correct isolation). The most difficult attack to deal with is the *Noise Addition* attack, where the detection still performs well (90% of detection), but the isolation presents poor performances (40% correct isolation).

The results about the "Detection counter" show the efficiency of the proposed distributed analysis. In almost all of *Replay*, *Stuck-at*, and *Dynamic Perturbation* attacks, all the IoT devices belonging to the cluster where the specific sensor has been attacked detect the change. Even in the case of *Noise Addition* attack (the most critical for the isolation procedure), the “Detection Counter” is larger or equal to 1 even when the attacked IoT device cannot been isolated successfully. This means that the proposed algorithm is able to detect the presence of the attack at cluster level even when it was not able to specifically isolate the attacked IoT device.

The detection times (i.e., Min, MW and Max Detection Times) show the ability of the proposed solution to promptly detect the presence of an attack. The results about $\hat{t}$, i.e., the estimate of the time-instant when the attack started, show the excellent capability of the proposed solution to correctly estimate when a given attack started in a given sensor. This allows, e.g., to discard the erroneous values transmitted by the attacked sensor, obtaining a cleaner dataset.

We also evaluated the average computational time[6] for the creation of the dependency graph and the detection/isolation phases of the proposed solution. Creating the dependency graph is the most time-consuming process and requires (in average) 36.2s, while the average computational time per unit of the detection and isolation phases is 26.4ms. We want to stress that the creation of the dependency graph can be done in the FU, and the detection and isolation phases can be performed even by a small device like a Raspberry Pi.

Countermeasures

In order to evaluate the countermeasure decision process, we will assume that a IoT device in the network is being attacked (e.g., sensor #11, Gamma DHT22-Temp). The attack has been successfully detected, and we must decide the appropriate countermeasure to be applied. As shown in Figure 5.19, we divided the attacks into families depending on the attack type: *stuck-at* and *replay* are a (probable) consequence of an attack to the sensor software, while *noise addition* and *dynamic perturbation* are most probably related to sensor replacement. Without loss of generality, all the value costs will be in the range $[0 - 10]$, where 10 is the maximum value.

The attack cost is summarized in table 5.7, and it assumes that the attacker is able to physically access the nodes. Moreover, we assume that replacing a node is more difficult than tampering an existing one. As a

[6] The reference hardware platform is a 2,5 GHz Intel Core i7 with 16 GB RAM at 2133MHz LPDDR3.

Table 5.6: Experimental results on the considered dataset. The symbol ✓means that the attack has been detected/isolated, while the symbol – means that the attack has not been detected/isolated.

	Sensor	#1	#2	#3	#4	#5	#7	#8	#9	#11	#12
Replay	Detected Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Detection counter	5 / 5	3 / 3	5 / 5	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
	Min Detection Time	2088	2144	2091	2160	2542	2862	2151	2442	2760	2150
	MW Detection Time	2783	2256	2824	2254	2747	3022	2168	3095	2840	2166
	Max Detection Time	3086	2440	3072	2472	2867	3043	2454	3095	3006	2260
	Isolated Attack	✓	✓	✓	✓	–	✓	✓	✓	✓	✓
	$\hat{t}$	2009.9	2063.3	2027.4	2055.3	–	2253.6	2068.3	2162.1	2291.4	2114.0
Stuck-at	Detected Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Detection counter	5 / 5	3 / 3	5 / 5	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
	Min Detection Time	1865	1911	1863	1925	1857	1950	1900	1924	2431	1938
	MW Detection Time	1952	2114	1945	1980	1938	1955	1939	2181	2694	1974
	Max Detection Time	1964	2137	1961	2133	1954	1967	2136	2241	2718	2122
	Isolated Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	$\hat{t}$	1816.0	1854.0	1818.0	1839.7	1831.1	1830.2	1856.7	1843.7	1909.4	1886.3
Noise Add.	Detected Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	–
	Detection counter	5 / 5	1 / 3	5 / 5	1 / 3	2 / 3	2 / 3	1 / 3	1 / 3	2 / 3	0 / 3
	Min Detection Time	1974	2795	1920	3055	1973	2715	3067	2062	2431	–
	MW Detection Time	2473	–	2431	–	2749	2747	–	–	2734	–
	Max Detection Time	2742	2795	2493	3055	2749	2747	3067	2062	2734	–
	Isolated Attack	✓	–	✓	–	–	✓	–	–	✓	–
	$\hat{t}$	1846.6	–	1834.4	–	–	1847.0	–	–	1863.6	–
Dynamic Pert.	Detected Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Detection counter	5 / 5	3 / 3	5 / 5	2 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3	3 / 3
	Min Detection Time	1810	1947	1807	2557	1813	1859	1896	1933	1862	1975
	MW Detection Time	1936	2147	1872	2750	1920	1938	1944	2001	1974	2094
	Max Detection Time	2003	2702	1896	2750	1975	1982	2509	2060	2017	2311
	Isolated Attack	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	$\hat{t}$	1803.5	1858.3	1802.7	1974.0	1807.0	1804.3	1847.0	1809.2	1803.8	1910.3

Table 5.7: Attack table cost

	Stuck-At	Replay	Noise add.	Dyn. Pert.
Time required	2	5	7	7
Equipment cost	3	3	3	3
Skill required	5	5	7	7
Physical access	4	4	4	4
Average	3.5	4.25	5.25	5.25

Table 5.8: Damage cost for sensor 11

	Routing tree position	Cluster under attack	Dependency graph position	Average
Cost	4	3	2	3

Table 5.9: Countermeasure cost for sensor 11

	Refresh address	Refresh routing tree	Change keys
Signaling Cost	3	5	10
Time Cost	4	6	7
Average	3.5	5.5	8.5

matter of fact, using a software vulnerability should be easier than gaining access to the network and replacing an existing device (without triggering an immediate alarm).

The damage costs for the related IoT device are shown in table 5.8. Here, we do not consider the data relevance, as in our system all collected data are equally significative.

The possible countermeasures costs are summarized in table 5.9. Note that a higher security countermeasure should imply also the use of the lower level ones.

The adopted countermeasure will have total cost less or equal to the damage cost: in our example, when an attack is detected, we can change the MAC-16 (MAC short) [57,146] address of our devices and, if a new attack is perform, we can modify the routing algorithm tree with a complete network re-configuration.

From the analysis, it is possible to conclude that an Address Refresh (e.g., by using the techniques outlined in [57, 146]) is a valid countermeasure. On the contrary, if the sensor is more central in the dependency graph (i.e., sensors #1 and #3), the appropriate countermeasure would be to apply an Address Refresh and a Routing Reconfiguration. Cryptography keys renewal will be used as a last resort in case of an attack to sensors generating important data or if the attack persists after the network reconfiguration.

In this paper, we proposed a novel IDS based on the analysis of data acquired in real-time by different Fog/IoT devices. The DataIDS can promptly detect a cyber-attack affecting a device of the FC/IoT system as well as effectively isolate it within the network to support the reaction phase. In order to react to the attack, we propose an attack-tree based evaluation system, which has the advantage of avoiding countermeasures that are disproportionate with respect to the attack and the damage costs.

We like to stress that the proposed system can be used also to *strengthen* the robustness of a Fog/IoT system against attacks. From the dependency graph, it is in fact possible to highlight 1) the nodes that are unconnected, and 2) the nodes with high correlation index. In the first case, the nodes are either collecting outlier measures (thus discardable) or important measures (thus more nodes should be installed in that particular point). It is obvious that nodes with high correlation index, i.e more related in the dependency graph, should be more protected. As a consequence, it is possible to choose the best candidate nodes to be, for example, hardened by physical security measures (e.g., anti-tampering hardware).

We implemented a testbed to validate the performance of the DataIDS on a real dataset and as shown by our results, the proposed intrusion detection system has several advantages over other kinds of approaches, and it can be easily implemented in constrained resource devices.

In future works, we plan to extend our model in order to better address the problem of data privacy by using fog devices. This will allow to maintain the user data private while enabling cooperative intrusion detection capabilities among different and logically separated sensors zones [152].

Chapter 6

Conclusion and Future Works

Per aspera sic itur ad astra
Through the roughness to the stars – Cicerone

This chapter summarizes the contribution of this dissertation and outlines some possible future research topics.

IoT is expected to play a major role in future applications, with a forecast of billions of Internet-connected devices. Securing IoT devices and protecting their applications from privacy leaks is a challenge, due to their weak (computational and storage) capabilities, and their proximity with sensitive data. Considering the resource-constrains of such devices, their long lifetime, and the intermittent connections, classical security approaches are often too difficult or impractical to apply.

In my opinion, IoT technologies are growing rapidly but the security and privacy technologies applied to these devices are still insufficient. Recent DDoS attack have shown that there is still work to be done, both on technology and end-users awareness. While not everyone has the technical skills to apply security measures individually, it is necessary to educate the users on using some basic good practices, such as changing the default password immediately. Moreover, it is worth noticing that security must address not only the confidentiality and integrity of the information, but also the availability of the network. As a matter of fact, availability is the most difficult part to obtain, and it will be reached only with a good network design and a continuous monitoring of network activity.

My research focused on two major topics: protocols and application, and security for IoT.

Protocols and applications for IoT. A first research done in this field was to evaluate the coexistence of protocols designed for different purposes. We have found a negative interaction between two protocols. In RPL standard, 6LoWPAN-ND is not mandatory to perform correct routing operations, moreover in 6LoWPAN-ND is not mentioned that its use could affect in someway routing protocols. Unexpectedly, I found that 6LoWPAN-ND can have a direct impact on RPL and can affect routing operations, leading to energy waste in the notwork.
The second result obtained in this field was the creation of an algorithm for time constraint applications in a VANET context. In particular, in the proposed solution a particular packet, called token, circulates in the network collecting information stored in each visited vehicle. The constrain was the time to perform the network polling: it shall return to the first node before the available time expired.
Moreover, a cooperative spectrum sensing protocol for WSN was developed. In this scenario the objective was to find if the cooperative spectrum sensing may be helpful during the network set-up to assign the best channel to the single PAN. The allocation is performed by the Super PAN coordinator while the spectrum sensing is coordinated by the Child PAN Coordinators and performed by each device in the PANs.

Security in IoT. The first result obtained was represented by the proposed shuffling algorithm used to change periodically MAC/IP addresses. In this context an ad-hoc modified version of the HMAC was used. Obviously, using hash functions, we must take into account the birthday paradox, and check if an address collision occurs. The main advantage of this algorithm is that the address shuffling can be triggered by messages piggybacked in normal routing maintenance semi-periodic messages, reducing the network overhead to almost zero.
The second result obtained was the development of a novel IDS based on the analysis of data acquired in real-time by different Fog/IoT devices. Using the sensor measures allows to detect the so-called False Data Injection attacks, in which an attacker tries to send fake measurement. The entire IDS is based on the concept of dependency graph: a graph where if two or more nodes send “similar” data stream are connected. To validate the performance of the IDS a testbed was implemented.

On the same topic I proposed another system to achieve a dynamic security level for smart home environment. Smart gateways (i.e., firewall) located at the ISP side and close to the users cooperate to detect and react against different types of attacks.

All the security topics analyzed during my Ph.D. studies can be subject to further investigations. As a matter of fact, the topics I analyzed are going to increase their importance in the future, and more studies are for sure needed.
One of the first topic that I suggest to investigate is the possibility to create an authentication method for IoT devices based on their hardware components. In particular, my idea is to investigate if hardware components leave fingerprints and if so, using these fingerprints to create new authentication methods.

About Moving Target Defense, to reach the best network confusion, MAC/IP addresses pseudo-random shuffling might not be sufficient. For this reason I will investigate the possibility to create ad-hoc objective function/metric for RPL able to select the “preferred parent” in a pseudo-random mode (from the attacker point of view) in order to create routing confusion, and prevent attacks based on the correlation of data paths.

Another interesting topic is related to the routing table compression for RPL protocol. In the last year many researchers pointed out that even though RPL is the best routing protocol for WSN and more in general IoT, its routing tables can become large enough to exceed a node memory and computation capability. In particular, the closer the nodes are to the root, the more entries in the routing table they have. My objective is to find novel ways to optimize the routing table. This, of course must not go against the security goals of the network.

Finally, Physical-layer security (PLS) has gained the attention of the research community in recent years, particularly for IoT applications. Contrary to classical cryptography, PLS provides security at physical layer, and it can be proven secure regardless of the computational power owned by the attacker. The investigations on PLS are numerous in the literature, but one main issue seems to be kept apart: how to measure the benefit that PLS can bring to cryptography (and vice-versa)? As a matter of fact, ‘classical’ cryptography and PLS are not mutually exclusive, and they can benefit from each other. This is particularly compelling in IoT scenarios, where finding the best way to engineer the system leads to smaller, cheaper, and more powerful devices.

That's All Folks !

Appendix A

GAUChO Project

This Appendix is related to the *GAUChO - A Green Adaptive Fog Computing and Networking Architecture* project funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W_004.

The GAUChO project is promoted by four Italian Universities: University of Florence (UNIFI), University Rome – La Sapienza (UNIRM), Alma Mater Studiorum – University of Bologna (UNIBO), and Politecnico di Milano (POLIMI).

In this section, part of [144] is presented.

A.1 The GAUChO Project: Overview and Challenges

In recent years, the Cloud Computing (CC) paradigm [38, 136] has become very popular by providing customers and enterprises with an ubiquitous on-demand network access to remote computing and storage platforms, or even services that can be rapidly provisioned and released with minimal design and operational effort. However, several constraints in terms of requested bandwidth and real time processing suggest to move processing as *close* as possible to data generation units (i.e., directly at the end-device or cluster of end-devices suitably formed) so as to minimize the "data production to decision making" latency.

In this perspective, Fog Computing (FC) is an emerging paradigm that extends CC towards the *edge* of the network [44, 96, 134]. In particular, FC refers to a distributed computing infrastructure confined on a limited geographical area in which some applications/services run directly at the network edge in smart end-devices. The goal of FC is to improve efficiency and reduce the amount of data that needs to be transported to the Cloud for massive data processing, analysis and storage.

Furthermore, the design of efficient solutions within FC also requires investigate a novel communication/networking paradigm, called Fog Networking (FN), in order to meet specific configurability, adaptability, flexibility and energy/spectrum-efficiency constraints. To this purpose, self-adaptive design solutions [113], where the application autonomously adapts to follow context changes need to be introduced.

Generally speaking, FN leverages past experience in wireless communication and networking research, and fuses the latest advances in devices and network systems in the ecosystem of computing and networking. As a consequence, effective and efficient FN methodologies are of paramount importance to meet specific requirements.

Despite FN exhibits promising technical features, new challenging issues are raised at the same time. In particular, FN has to guarantee the data delivery reliability among edge entities as well as to efficiently support end-devices mobility and service continuity. Device-to-Device communications [40,122,156] and, recently, the new concept of mmWave communication technology [141] are gaining importance in the literature since they appear to be promising solutions yielding efficient FN capabilities [117]. However, it appears from the existing literature, that significant research efforts have to be devoted to identify and tailor solutions to specific FN needs and to pursue an efficient and functional networking and computing capabilities integration in a same FC+FN platform. This is corroborated by the fact that generally FC and FN are designed and developed independently each other, hence not being able to fully exploit the complementary characteristics.

The "Green Adaptive Fog Computing and Networking Architecture" (GAUChO) project, funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W-004 aims at designing a novel distributed and heterogeneous architecture able to functionally integrate and jointly optimize FC and FN capabilities in the same platform.

The main objective of GAUChO [17] is to propose a novel heterogeneous and distributed architecture able to integrate and optimize Fog Computing (FC) and Fog Networking (FN) functionalities in the same platform. The joint FC+FN architecture, representing the overall outcome of the project, aims at supporting low-latency and energy-efficiency as well as security, self-adaptation, and spectrum efficiency by means of a strict collaboration among end-devices and FC+FN units in a same integrated platform. Additionally, the development of suitable analytic methods and definition of appropriate techniques will enable extra relevant characteristics of the FC+FN platform including ubiquity, decentralized management, cooperation, proximity to end users, dense geographical distribution, efficient support for mobility and real-time applications. To achieve this goal, the GAUChO project foresees to address several relevant and challenging research topics that require skills and knowledge in different scientific fields.

A.2 Contribution

This project is expected to have a very large impact on a wide range of applications dealing with the monitor and control of large-scale distributed systems such as Connected Vehicles (CVs) and Intelligent Monitoring Systems (IMSs).

The scientific contributions that the project aims at achieving, are significant and relevant in several aspects, such as:

1. Efficient schemes for the coordinated management of resources and interference in heterogeneous wireless communication systems;

2. Joint optimization of communication and computing capabilities to support energy-efficient management and self-reconfigurations;

3. A learning modality permitting software agents to detect variations within the integrated FC+FN platform;

4. Model-free fault diagnosis systems and comprehensive methodology integrating intelligence-based mechanisms for optimally managing energy consumption, detecting changes in environment/system under inspection, and evaluating and mitigating the possible occurrence of faults affecting the end-devices computing units.

The GAUChO project is expected to have a significant technological impact on many up-to-date and relevant families of technologies, such as

Smart Wireless Sensor Networks (SWSNs), Smart Objects of Internet-of-Things and Intelligent Embedded Systems. All in all, the project will allow the FC+FN paradigm to enter into a new phase where real-world problems emerging from complex applications are addressed and effectively solved. It is expected that project outcomes will move from basic research to mass production in years 2018-25 providing significant economic benefits to masses of potential end users, together with the added value of the offered application services, limited cost of the communication and processing infrastructure.

A.3 GAUChO Platform

The GAUChO platform is composed by three-level hierarchy tiers as shown in Fig. A.1.

End-devices are connected to FC units (either directly or through gateways) that, in turn, may be interconnected and linked to the Cloud depending on application needs and constraints. This is where FN comes into play by providing, in an integrated approach with the FC platform, methodologies and tools to enable end-devices and FC units to carry out distributed storage and processing.

To achieve these goals, this project will propose novel solutions in the following relevant research fields:

- computing methodologies;
- communication and networking strategies;
- intelligent/adaptation mechanisms;

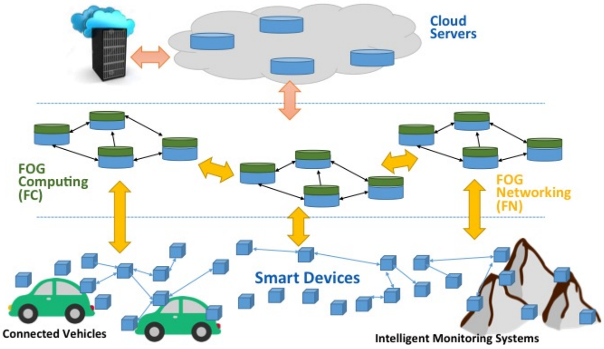


Figure A.1: GAUChO platform.

Appendix B

Publications

International Journals

1. L. Pierucci, T. Pecorella, **F. Nizzi**. "Network Sentiment Framework to Improve Security and Privacy for Smart Home", *Future Internet*, vol. 10, 2018. [DOI:10.3390/fi10120125]

2. **F. Nizzi**, T. Pecorella, F. Esposito, L. Pierucci, R. Fantacci. "IoT Security via Address Shuffling: the Easy Way", *IEEE Internet Of Things Journal*, vol. 6, iss. 2, pp.3764-3774, 2019.

   [DOI: 10.1109/JIOT.2019.2892003]

3. F. Chiti, R. Fantacci, **F. Nizzi**, L. Pierucci, C. Borrego. "A Distributed Token Passing Protocol for Time Constrained Data Gathering in VANETs", *Electronics*, vol. in press, 2019. (Special Issue: Vehicular Networks and Communications) [DOI: 10.3390/electronics8080823]

4. R. Fantacci, **F. Nizzi**, T. Pecorella, L. Pierucci, M. Roveri. "False Data Detection for Fog and Internet of Things Networks", *Sensors*, vol. in press, 2019. (Special Issue: Security, Privacy, and Trustworthiness of Sensor Networks and Internet of Things) [DOI: 10.3390/s19194235].

Submitted

1. T. Pecorella, **F. Nizzi**, F. Chiti, R. Fantacci. "Interactions between RPL and ND protocols: incorrect loop detection and unnecessary recovery", *IOT Journal* (Submitted).

2. D. Marabissi, S. Caputo, L. Mucchi, **F. Nizzi**, T. Pecorella, R. Fantacci, T. Nawaz, M. Seminara, J. Catani. "Experimental measurements of a joint 5G-VLC communication for future vehicular networks", *IEEE Vehicular Technology Magazine*, 2019. (Special Issue: IEEE Future Networks Series on 5G) (Submitted).

International Conferences and Workshops

1. F. Chiti, R. Fantacci, **F. Nizzi**, L. Pierucci, T. Pecorella. "A cooperative spectrum sensing protocol for IEEE 802.15.4m wide-area WSNs", in *Proc. of IEEE International Conference on Communications (IEEE ICC'17)*, Paris (France), 2016.

2. **F. Nizzi**, T. Pecorella, A. Bonadio, F. Chiti, R. Fantacci, D. Tarchi, W. Cerroni. "FOG-oriented Joint Computing and Networking: the GAUChO Project Vision", in *Proc. of AEIT International Annual Conference (AEIT 2018)*, Bari (Italy), 2018.

3. **F. Nizzi**, T. Pecorella, M. Bertini, R. Fantacci, M. Bastianini, C. Cerboni, A. Buzzigoli, M. Gattoni, A. Fratini. "Evaluation of IoT and videosurveillance applications in a 5G Smart City: the Italian 5G experimentation in Prato", in *Proc. of AEIT International Annual Conference (AEIT 2018)*, Bari (Italy), 2018.

4. **F. Nizzi**, T. Pecorella, M. Bastianini, C. Cerboni, A. Buzzigoli, A. Fratini "The Role of Network Simulator in the 5G Experimentation", in *Proc. of Workshop on Next-Generation Wireless with ns-3 (WNGW 2019)*, Florence (Italy), 2019.

5. **F. Nizzi**, T. Pecorella. "Protocol Prototype Implementation Using ns-3: a Use-Case", in *Proc. of Workshop on Next-Generation Wireless with ns-3 (WNGW 2019)*, Florence (Italy), 2019.

6. L. Mucchi, **F. Nizzi**, T. Pecorella, R. Fantacci, F. Esposito. "Benefits of Physical Layer Security to Cryptography: Tradeoff and Applications", in *Proc. of IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom 2019)*, Sochi (Russia), 2019.

7. **F. Nizzi**, T. Pecorella, S. Caputo, L. Mucchi, R. Fantacci, M. Bastianini, C. Cerboni, A. Buzzigoli, A. Fratini, T. Nawaz, M. Seminara, J.

Catani. “Data dissemination to vehicles using 5G and VLC for Smart Cities”, in *Proc. of AEIT International Annual Conference (AEIT 2019)*, Florence (Italy), 2019.

Bibliography

[1] "5 città per il 5g." [Online]. Available: https://www.mise.gov.it/index.php/it/194-comunicati-stampa/2036228-5-citta-per-il-5g

[2] "5G for europe: An action plan." [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52016DC0588&from=IT

[3] "5G PPP phase1 security landscape." [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_White-Paper_Phase-1-Security-Landscape_June-2017.pdf

[4] "5G PPP use cases and performance evaluation models." [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1.0.pdf

[5] "5G public private partnership (5 GPPP)." [Online]. Available: https://5g-ppp.eu/projects/

[6] "5G vision." [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[7] "Attenzione a quei giocattoli smart, spiano i vostri bambini." [Online]. Available: https://www.repubblica.it/tecnologia/sicurezza/2016/12/12/news/attenzione_a_quei_giocattoli_smart_sono_delle_spie-153970997/

[8] "Best intentions of 2008." [Online]. Available: http://content.time.com/time/specials/packages/completelist/0,29569,1852747,00.html

[9] "Breaking down mirai: An iot ddos botnet analysis." [Online]. Available: https://www.imperva.com/blog/malware-analysis-mirai-ddos-botnet/

[10] The bro network security monitor. [Online]. Available: https://www.bro.org

[11] "CEO to shareholders: 50 billion connections 2020." [Online]. Available: https://www.ericsson.com/en/press-releases/2010/4/ceo-to-shareholders-50-billion-connections-2020

[12] "Connected and automated vehicles." [Online]. Available: http://autocaat.org/Technologies/Automated_and_Connected_Vehicles/

[13] "A cute toy just brought a hacker into your home." [Online]. Available: https://www.nytimes.com/2017/12/21/technology/connected-toys-hacking.html

[14] "Detailed specifications of the terrestrial radio interfaces of international mobile telecommunications-advanced (IMT-advanced)." [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2012-3-201801-I!!PDF-E.pdf

[15] "Downdetector." [Online]. Available: http://downdetector.com/status/level3/map/

[16] "Free pool of ipv4 address space depleted." [Online]. Available: https://www.nro.net/ipv4-free-pool-depleted

[17] "GAUChO - a green adaptive fog computing and networking architecture." [Online]. Available: https://www.gaucho.unifi.it/index.php

[18] "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond." [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf

[19] "In the matter of genesis toys and nuance communications." [Online]. Available: https://epic.org/privacy/kids/EPIC-IPR-FTC-Genesis-Complaint.pdf

[20] "The internet of things." [Online]. Available: https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf

[21] "The internet of things - how the next evolution of the internet is changing everything." [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[22] "Ipv6: How many ip addresses can dance on the head of a pin?" [Online]. Available: https://www.edn.com/electronics-blogs/other/4306822/IPV6-How-Many-IP-Addresses-Can-Dance-on-the-Head-of-a-Pin-

[23] "ITU finds way forward for 3g mobile systems." [Online]. Available: https://www.itu.int/itunews/issue/1999/04/imt2000.html

[24] "Popular internet of things forecast of 50 billion devices by 2020 is outdated." [Online]. Available: https://www.telecompaper.com/news/lg-unveils-internetready-refrigerator--221266

[25] "Popular internet of things forecast of 50 billion devices by 2020 is outdated." [Online]. Available: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[26] "Sintesi "PROGETTO 5G" area 2: Prato e L'Aquila." [Online]. Available: http://www.sviluppoeconomico.gov.it/images/stories/documenti/Sintesi-progetto-5g-presentazione-13102017.pdf

[27] “Sperimentazione 5g alle porte, selezionati i migliori progetti.” [Online]. Available: http://bandaultralarga.italia.it/sperimentazione-5g-alle-porte-selezionati-migliori-progetti/

[28] “The year in ideas; news that glows.” [Online]. Available: https://www.nytimes.com/2002/12/15/magazine/the-year-in-ideas-news-that-glows.html

[29] “Wrc 19 wrap-up: Additional spectrum allocations agreed for imt-2020 (5g mobile).” [Online]. Available: https://techblog.comsoc.org/2019/11/22/wrc-19-wrap-up-additional-spectrum-allocations-agreed-for-imt-2020-5g-mobile/

[30] “IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments,” *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, July 2010.

[31] “Ieee standard for local and metropolitan area networks – part 15.4: Low-rate wireless personal area networks (lr-wpans),” *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, September 2011.

[32] “Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 3: Physical layer (phy) specifications for low-data-rate, wireless, smart metering utility networks,” *IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–252, April 2012.

[33] “Ieee standard for local and metropolitan area networks – part 15.4: Low-rate wireless personal area networks (lr-wpans) – amendment 6: Tv white space between 54 mhz and 862 mhz physical layer,” *IEEE Std 802.15.4m-2014 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–118, April 2014.

[34] “IEEE Standard for Low-Rate Wireless Networks,” *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.

[35] “View on 5G architecture, version 2.0,” 2017. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf

[36] *MTD ’18: Proceedings of the 2018 Workshop on Moving Target Defense.* New York, NY, USA: ACM, 2018. [Online]. Available: http://csis.gmu.edu/MTD-2018/

[37] “IEEE Standard for Local and metropolitan area networks–Part 15.7: Short-Range Optical Wireless Communications - Redline,” *IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011) - Redline*, pp. 1–670, April 2019.

[38] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 337–368, First 2014.

[39] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996–1014, March 2018.

[40] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, "A survey on socially-aware device-to-device communications," *IEEE Communications Surveys Tutorials*, 2018.

[41] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, no. 1, pp. 40–62, 2011.

[42] C. Alippi, V. D'Alto, M. Falchetto, D. Pau, and M. Roveri, "Detecting changes at the sensor level in cyber-physical systems: Methodology and technological implementation," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 1780–1786.

[43] C. Alippi, S. Ntalampiras, and M. Roveri, "A cognitive fault diagnosis system for distributed sensor networks," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 8, pp. 1213–1226, 2013.

[44] A. Aljumah and T. A. Ahanger, "Fog computing and security issues: A review," in *2018 7th International Conference on Computers Communications and Control (ICCCC)*, May 2018, pp. 237–239.

[45] S. Allani, T. Yeferny, R. Chbeir, and S. B. Yahia, "A novel vanet data dissemination approach based on geospatial data," *Procedia Computer Science*, vol. 98, pp. 572 – 577, 2016, the 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2016)/Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916322347

[46] E. Ancillotti, R. Bruno, M. Conti, and A. Pinizzotto, "Dynamic address autoconfiguration in hybrid ad hoc networks," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 300–317, 2009.

[47] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *Proceedings*

of the 26th USENIX Conference on Security Symposium, ser. SEC'17. Berkeley, CA, USA: USENIX Association, 2017, pp. 1093–1110. [Online]. Available: http://dl.acm.org/citation.cfm?id=3241189.3241275

[48] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[49] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009. [Online]. Available: http://www.rfidjournal.com/articles/view?4986

[50] R. Baldemair, E. Dahlman, G. Fodor, G. Mildh, S. Parkvall, Y. Selén, H. Tullberg, and K. Balachandran, "Evolving wireless communications: Addressing the challenges and expectations of the future," *IEEE Vehicular Technology Magazine*, vol. 8, no. 1, pp. 24–30, 2013.

[51] L. Bartolozzi, T. Pecorella, and R. Fantacci, "ns-3 RPL module: IPv6 Routing Protocol for Low power and Lossy Networks," in *WNS3 - Workshop on ns-3.* ACM, 6 2012.

[52] M. V. Belen, "Implications of WRC-15 on spectrum and 5G," Joint Research Centre (JRC) - European Commission, Tech. Rep., 2016. [Online]. Available: http://publications.jrc.ec.europa.eu/repository/bitstream/JRC102401/jrc102401_jrc%20technical%20report%20-deliverable_i_5g_fv-6.pdf

[53] Y. Bi, K.-H. Liu, L. X. Cai, X. Shen, and H. Zhao, "A multi-channel token ring protocol for qos provisioning in inter-vehicle communications," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, 2009.

[54] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, March 2012.

[55] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach," *Computer Communications*, vol. 98, no. C, pp. 52–71, 2017.

[56] B. Brik, N. Lagraa, A. Lakas, and A. Cheddad, "Ddgp: Distributed data gathering protocol for vehicular networks," *Vehicular Communications*, vol. 4, pp. 15 – 29, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214209616000024

[57] L. Brilli, T. Pecorella, L. Pierucci, and R. Fantacci, "A novel 6LoWPAN-ND extension to enhance privacy in ieee 802.15.4 networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[58] D. M. C. Alippi, R. Fantacci and M. Roveri, "A cloud to the ground: the new frontier of intelligent and autonomous networks of things," *to appear on IEEE Transaction On Vehicular Tecnology.*

[59] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, “Performance comparison between ieee 802.11p and lte-v2v in-coverage and out-of-coverage for cooperative awareness,” in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 109–114.

[60] Centers for Medicare & Medicaid Services, “The Health Insurance Portability and Accountability Act of 1996 (HIPAA),” Online at http://www.cms.hhs.gov/hipaa/, 1996.

[61] W. Chang, H. Lin, and B. Chen, “Trafficgather: An efficient and scalable data collection protocol for vehicular ad hoc networks,” in *2008 5th IEEE Consumer Communications and Networking Conference*, Jan 2008, pp. 365–369.

[62] M. Chaqfeh, A. Lakas, and I. Jawhar, “A survey on data dissemination in vehicular ad hoc networks,” *Vehicular Communications*, vol. 1, no. 4, pp. 214 – 225, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214209614000448

[63] A. Chelli, M. Bagaa, D. Djenouri, I. Balasingham, and T. Taleb, “One-step approach for two-tiered constrained relay node placement in wireless sensor networks,” *IEEE Wireless Communications Letters*, vol. 5, no. 4, pp. 448–451, Aug 2016.

[64] R.-G. Cheng and R.-I. Chang, “Data transmitting method with multiple token mechanism in wireless token ring protocol,” U.S. Patent 7 975 074, 2011.

[65] R.-G. Cheng, C.-Y. Wang, L.-H. Liao, J.-S. Yang *et al.*, “Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks,” *IEEE Communications Letters*, vol. 10, no. 2, pp. 123–125, 2006.

[66] M. O. Cherif, S. M. Senouci, and B. Ducourthial, “A new framework of self-organization of vehicular networks,” *2009 Global Information Infrastructure Symposium*, pp. 1–6, 2009.

[67] F. Chiti, D. Di Giacomo, R. Fantacci, and L. Pierucci, “Interference aware approach for d2d communications,” in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[68] F. Chiti, R. Fantacci, M. Loreti, and R. Pugliese, “Context-aware wireless mobile autonomic computing and communications: research trends and emerging applications,” *IEEE Wireless Communications*, vol. 23, no. 2, pp. 86–92, April 2016.

[69] F. Chiti, R. Fantacci, and L. Pierucci, “Social-aware relay selection for cooperative multicast device-to-device communications,” *Future Internet*, vol. 92, no. 4, 2017.

[70] F. Chiti, R. Fantacci, and A. Tani, “Performance evaluation of an adaptive channel allocation technique for cognitive wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, 2016.

[71] F. Chiti, R. Fantacci, D. Giuli, F. Paganelli, and G. Rigazzi. Springer, 2016, ch. Communications Protocol Design for 5G Vehicular Networks, pp. 625–649.

[72] F. Chiti, R. Fantacci, F. Nizzi, L. Pierucci, and C. Borrego, “A distributed token passing protocol for time constrained data gathering in vanets,” *ELECTRONICS*, pp. 1–10, 2019.

[73] F. Chiti, R. Fantacci, F. Nizzi, L. Pierucci, and T. Pecorella, “A cooperative spectrum sensing protocol for ieee 802.15. 4m wide-area wsns,” in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[74] H. Chour, Y. Nasser, H. Artail, A. Kachouh, and A. Al-Dubai, “Vanet aided d2d discovery: Delay analysis and performance,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8059–8071, Sep. 2017.

[75] A. Conta, S. Deering, and M. Gupta, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification,” Internet Request for Comments, pp. 1 – 24, March 2006. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4443.txt

[76] T. Cruz, P. Simoes, E. Monteiro, F. Bastos, and A. Laranjeira, “Cooperative security management for broadband network environments,” *Security and Communication Networks*, vol. 8, no. 18, pp. 3953–3977.

[77] G. Cugola and A. Margara, “Processing flows of information: From data stream to complex event processing,” *ACM Comput. Surv.*, vol. 44, no. 3, 2012. [Online]. Available: http://doi.acm.org/10.1145/2187671.2187677

[78] H. Debar, D. Curry, and B. Feinstein, “The Intrusion Detection Message Exchange Format (IDMEF),” Internet Request for Comments, pp. 1 – 157, March 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4765.txt

[79] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” Internet Request for Comments, pp. 1 – 37, December 1995. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1883.txt

[80] Z. Deng, Y. Lu, C. Wang, and W. Wang, “Ewtrp: enhanced wireless token ring protocol for small-scale wireless ad hoc networks,” in *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, vol. 1. IEEE, 2004, pp. 398–401.

[81] F. F. Digham, M.-S. Alouini, and M. K. Simon, “On the energy detection of unknown signals over fading channels,” *IEEE transactions on communications*, vol. 55, no. 1, pp. 21–24, 2007.

[82] R. Droms, "Dynamic Host Configuration Protocol," Internet Request for Comments, pp. 1 – 45, March 1997. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2131.txt

[83] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "Wtrp-wireless token ring protocol," *IEEE transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1863–1881, 2004.

[84] R. Fantacci, F. Nizzi, T. Pecorella, L. Pierucci, and M. Roveri, "False data detection for fog and internet of things networks," *Sensors*, 2019.

[85] R. Fantacci, T. Pecorella, R. Viti, and C. Carlini, "A network architecture solution for efficient IoT WSN backhauling: challenges and opportunities," *Wireless Communications, IEEE*, vol. 21, no. 4, pp. 113–119, August 2014.

[86] N. Ferguson, B. Schneier, and T. Kohno, *Cryptography engineering.* Wiley Online Library, 2010.

[87] A. Festag, "Standards for vehicular communication—from IEEE 802.11p to 5g," *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, Nov 2015. [Online]. Available: https://doi.org/10.1007/s00502-015-0343-0

[88] A. Fuchsberger, "Intrusion detection systems and intrusion prevention systems," *Information Security Technical Report*, vol. 10, no. 3, pp. 134 – 139, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1363412705000415

[89] S. Gallagher, "How one rent-a-botnet army of cameras, DVRs caused Internet chaos," *Ars Technica*, 2016. [Online]. Available: https://arstechnica.com/?post_type=post&p=981883

[90] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[91] A. A. Gendreau and M. Moorman, "Survey of intrusion detection systems towards an end to end secure internet of things," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Aug 2016, pp. 84–90.

[92] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1294–1312, thirdquarter 2015.

[93] J. Granjal and A. Pedroso, "An intrusion detection and prevention framework for internet-integrated CoAP WSN," *Security and Communication Networks*, 2018. [Online]. Available: https://doi.org/10.1155/2018/1753897

[94] Z. Hameed Mir and F. Filali, “Lte and ieee 802.11p for vehicular networking: a performance evaluation,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, May 2014. [Online]. Available: https://doi.org/10.1186/1687-1499-2014-89

[95] H. Hartenstein and L. Laberteaux, “A tutorial survey on vehicular ad hoc networks,” *IEEE Communications magazine*, vol. 46, no. 6, 2008.

[96] Q. F. Hassan, “Cloud and fog computing in the internet of things,” in *Internet of Things A to Z: Technologies and Applications.* Wiley-IEEE Press, 2018.

[97] Z. He and D. Zhang, “Cost-efficient traffic-aware data collection protocol in vanet,” *Ad Hoc Networks*, vol. 55, pp. 28 – 39, 2017, self-organizing and Smart Protocols for Heterogeneous Ad hoc Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870516302360

[98] Y. Hong, W. M. Liu, and L. Wang, “Privacy preserving smart meter streaming against information leakage of appliance status,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2227–2241, Sept 2017.

[99] R. Housley and B. Aboba, “Guidance for Authentication, Authorization, and Accounting (AAA) Key Management,” Internet Request for Comments, pp. 1 – 23, July 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4962.txt

[100] J. Hui and P. Thubert, “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks,” Internet Request for Comments, pp. 1 – 24, September 2011. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6282.txt

[101] V. P. Illiano and E. C. Lupu, “Detecting malicious data injections in event detection wireless sensor networks,” *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 496–510, Sep. 2015.

[102] V. P. Illiano, L. Muñoz-González, and E. C. Lupu, “Don’t fool me!: Detection, characterisation and diagnosis of spoofed and masked events in wireless sensor networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 279–293, May 2017.

[103] V. P. Illiano and E. C. Lupu, “Detecting malicious data injections in wireless sensor networks: A survey,” *ACM Comput. Surv.*, vol. 48, no. 2, pp. 24:1–24:33, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2818184

[104] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C. Lin, and X. Liu, “Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects,” *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[105] O. Kaiwartya and S. Kumar, *Enhanced Caching for Geocast Routing in Vehicular Ad Hoc Network*. Springer, New Delhi, 2014, vol. 243, ch. Intelligent Computing, Networking, and Informatics.

[106] G. Kambourakis, "Anonymity and closely related terms in the cyberspace: An analysis by example," *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 2 – 17, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214212614000209

[107] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 584–616, Fourth 2011.

[108] H. Kim, "Protection against packet fragmentation attacks at 6lowpan adaptation layer," in *2008 International Conference on Convergence and Hybrid Information Technology*, Aug 2008, pp. 796–801.

[109] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[110] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," Internet Request for Comments, pp. 1 – 11, February 1997. [Online]. Available: https://www.rfc-editor.org/rfc/rfc2104.txt

[111] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," Internet Request for Comments, pp. 1 – 14, May 2010. [Online]. Available: https://www.rfc-editor.org/rfc/rfc5869.txt

[112] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 631–648.

[113] C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervasive and Mobile Computing*, vol. 17, pp. 184–206, 2015, 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian.

[114] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Internet Request for Comments, pp. 1 – 12, August 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4919.txt

[115] C. Lee, L. Zappaterra, K. Choi, and H.-A. Choi, "Securing smart home: Technologies, security challenges, and security requirements," in *2014 IEEE Conference on Communications and Network Security*, Oct 2014, pp. 67–72.

[116] C. Lei, H. Zhang, J. Tan, Y. Zhang, and X. Liu, "Moving target defense techniques: A survey," *Security and Communication Networks*, vol. 2018, pp. 1–25, 2018. [Online]. Available: https://doi.org/10.1155/2018/3759626

[117] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, June 2013.

[118] K. B. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 878–893, 2009.

[119] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," Internet Request for Comments, pp. 1 – 13, March 2011. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6206.txt

[120] Z. Li, P. Wu, Y. Song, and J. Bi, "Intermittent data dissemination using node forwarding capability estimation in vehicle delay tolerant networks," in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, March 2016, pp. 119–125.

[121] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE transactions on Wireless Communications*, vol. 7, no. 4, pp. 1326–1337, 2008.

[122] S. Y. Lien, C. C. Chien, F. M. Tseng, and T. C. Ho, "3GPP device-to-device communications for beyond 4G cellular networks," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 29–35, March 2016.

[123] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, "5g new radio: waveform, frame structure, multiple access, and initial access," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 64–71, 2017.

[124] D. Lin, T.-S. Moh, and M. Moh, "A delay-bounded multi-channel routing protocol for wireless mesh networks using multiple token rings: extended summary," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on.* IEEE, 2006, pp. 845–847.

[125] H. Lin and N. W. Bergmann, "Iot privacy and security challenges for smart home environments," in *Informafion MDPI*, July 2016, pp. 67–72.

[126] R. A. Martin, "Managing vulnerabilities in networked systems," *Computer*, vol. 34, no. 11, pp. 32–38, Nov 2001.

[127] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *International Conference on Information Security and Cryptology.* Springer, 2005, pp. 186–198.

[128] N. F. Maxemchuk, "Reliable multicast with delay guarantees," *IEEE Communications Magazine*, vol. 40, no. 9, pp. 96–102, Sep. 2002.

[129] A. Mesodiakaki, F. Adelantado, L. Alonso, and C. Verikoukis, "Energy-efficient contention-aware channel selection in cognitive radio ad-hoc networks," in *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2012, pp. 46–50.

[130] R. Messier, *CEH v10 Certified Ethical Hacker Study Guide*. Wiley, 2019.

[131] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," Internet Request for Comments, pp. 1 – 30, September 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4944.txt

[132] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6," Internet Request for Comments, pp. 1 – 17, April 2006. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4429.txt

[133] L. Mucchi, F. S. Cataliotti, L. Ronga, S. Caputo, and P. Marcocci, "Experimental-based propagation model for vlc," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017, pp. 1–5.

[134] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys Tutorials*, 2018.

[135] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 78–82.

[136] S. Murugesan and I. Bojanova, "Mobile cloud computing," in *Encyclopedia of Cloud Computing*. Wiley-IEEE Press, 2016.

[137] S. Myagmar, A. J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on requirements engineering for information security (SREIS)*, vol. 2005, 2005, pp. 1–8.

[138] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," Internet Request for Comments, pp. 1 – 97, September 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4861.txt

[139] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1448–1459, April 2013.

[140] S. Nesargi and R. Prakash, "MANETconf: configuration of hosts in a Mobile Ad Hoc Network," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2002, pp. 1059–1068 vol.2.

[141] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges," *Wireless Networks*, vol. 21, no. 8, pp. 2657–2676, Nov 2015.

[142] F. Nizzi, T. Pecorella, M. Bastianini, C. Cerboni, A. Buzzigoli, and A. Fratini, "The role of network simulator in the 5g experimentation," in *Proceedings of the 2019 Workshop on Next-Generation Wireless with ns-3*. ACM, 2019, pp. 13–17.

[143] F. Nizzi, T. Pecorella, M. Bertini, R. Fantacci, M. Bastianini, C. Cerboni, A. Buzzigoli, M. Gattoni, and A. Fratini, "Evaluation of iot and video-surveillance applications in a 5g smart city: the italian 5g experimentation in prato," in *2018 AEIT International Annual Conference*. AEIT, 2018, pp. 1–5.

[144] F. Nizzi, T. Pecorella, A. Bonadio, F. Chiti, R. Fantacci, D. Tarchi, and W. Cerroni, "Fog-oriented joint computing and networking: the gaucho project vision," in *2018 AEIT International Annual Conference*. IEEE, 2018, pp. 1–6.

[145] F. Nizzi, T. Pecorella, S. Caputo, L. Mucchi, R. Fantacci, M. Bastianini, C. Cerboni, A. Buzzigoli, A. Fratini, T. Nawaz, J. Catani, and M. Seminara, "Data dissemination to vehicles using 5g and vlc for smart cities," in *2019 AEIT International Annual Conference*. AEIT, 2019, pp. 1–5.

[146] F. Nizzi, T. Pecorella, L. Pierucci, F. Esposito, and R. Fantacci, "Iot security via address shuffling: the easy way," *Internet of Things Journal*, vol. 6, pp. 3764–3774. [Online]. Available: https://ieeexplore.ieee.org/document/8606197

[147] S.-P. Oriyano, *Ceh: Certified ethical hacker version 9 study guide*. New York : John Wiley & Sons, Incorporated, 2016.

[148] J. Pacheco and S. Hariri, "Iot security framework for smart cyber infrastructures," in *Foundations and Applications of Self* Systems, IEEE International Workshops on*. IEEE, 2016, pp. 242–247.

[149] ——, "Anomaly behavior analysis for iot sensors," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, p. e3188, 2018.

[150] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[151] T. Pecorella, L. Brilli, and L. Mucchi, "The Role of Physical Layer Security in IoT: A Novel Perspective," *Information*, vol. 7, no. 3, p. 49, Aug. 2016. [Online]. Available: http://www.mdpi.com/2078-2489/7/3/49

[152] T. Pecorella, L. Pierucci, and F. Nizzi, “”network sentiment” framework to improve security and privacy for smart home,” *FUTURE INTERNET*, vol. 10, pp. 1–10, 2018. [Online]. Available: https://www.mdpi.com/1999-5903/10/12/125

[153] C. E. Perkins, J. T. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, “Ad hoc address autoconfiguration,” *draft-ietf-manet-autoconf-01.txt*, 2001.

[154] L. Pierucci, “The quality of experience perspective toward 5g technology,” *IEEE Wireless Communications*, vol. 22, no. 4, pp. 10–16, August 2015.

[155] S. Raza, L. Wallgren, and T. Voigt, “Svelte: Real-time intrusion detection in the internet of things,” *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661 – 2674, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870513001005

[156] G. Rigazzi, F. Chiti, R. Fantacci, and C. Carlini, “Multi-hop d2d networking and resource management scheme for m2m communications over lte-a systems,” in *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2014, pp. 973–978.

[157] G. Rémy, S. Senouci, F. Jan, and Y. Gourhant, “Lte4v2x — collection, dissemination and multi-hop forwarding,” in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 120–125.

[158] K. A. Scarfone and P. M. Mell, “Guide to intrusion detection and prevention systems (idps),” National Institute of Standards and Technology, Gaithersburg, MD, United States, Tech. Rep. SP 800-94, 2007.

[159] B. Schneier, “Attack trees,” *Dr. Dobb’s journal*, vol. 24, no. 12, pp. 21–29, 1999. [Online]. Available: https://www.schneier.com/academic/archives/1999/12/attack_trees.html

[160] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, “Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),” Internet Request for Comments, pp. 1 – 55, November 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6775.txt

[161] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” Internet Request for Comments, pp. 1 – 112, June 2014. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7252.txt

[162] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet.* John Wiley & Sons, 2011, vol. 43.

[163] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, “Secure data uploading scheme for a smart home system,” *Information Sciences*, vol. 453, pp. 186 – 197, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518303001

[164] D. Shin, V. Sharma, J. Kim, S. Kwon, and I. You, "Secure and efficient protocol for route optimization in pmipv6-based smart home iot networks," *IEEE Access*, vol. 5, pp. 11 100–11 117, 2017.

[165] R. Simoni, V. Jamali, N. Zlatanov, R. Schober, L. Pierucci, and R. Fantacci, "Buffer-aided diamond relay network with block fading," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 1982–1987.

[166] W. Stallings, *Cryptography and network security: principles and practice.* Pearson, 2017.

[167] J. K. Strosnider, T. Marchok, and J. Lehoczky, "Advanced real-time scheduling using the ieee 802.5 token ring," in *Real-Time Systems Symposium, 1988., Proceedings.* IEEE, 1988, pp. 42–52.

[168] J. Sun, "Wireless local communities in mobile commerce," in *Encyclopedia of Portal Technologies and Applications.* IGI Global, 2007, pp. 1204–1209.

[169] X. Sun, Y. Zhang, and J. Li, "Wireless dynamic token protocol for manet," in *2007 International Conference on Parallel Processing Workshops (ICPPW 2007)*, Sept 2007, pp. 1–5.

[170] Y. Sun and E. M. Belding-Royer, "Dynamic address configuration in mobile ad hoc networks," University of California, Tech. Rep., 2003.

[171] ——, "A study of dynamic addressing techniques in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 4, no. 3, pp. 315–329, 2004.

[172] D. Tacconi, I. Carreras, D. Miorandi, I. Chlamtac, F. Chiti, and R. Fantacci, "Supporting the sink mobility: a case study for wireless sensor networks," in *2007 IEEE International Conference on Communications*, June 2007, pp. 3948–3953.

[173] D. Thaler, "Enabling Security/Privacy Addressing On 6LoWPAN Technologies," draft-thaler-6lo-privacy-addrs-00, Internet Engineering Task Force, Aug. 2015. [Online]. Available: https://tools.ietf.org/pdf/draft-thaler-6lo-privacy-addrs-00.pdf

[174] ——, "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms," Internet Request for Comments, pp. 1 – 10, February 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8065.txt

[175] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," Internet Request for Comments, pp. 1 – 30, September 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4862.txt

[176] A. Tønnesen, "Mobile ad-hoc networks," *Penguin pixmap (c) everaldo. com-802.11 illustrations by Lars Strand*, 2014.

[177] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: Applications and related technical issues," *IEEE communications surveys & tutorials*, vol. 10, no. 3, 2008.

[178] I. Turcanu, P. Salvo, A. Baiocchi, and F. Cuomo, "An integrated vanet-based data dissemination and collection protocol for complex urban scenarios," *Ad Hoc Networks*, vol. 52, pp. 28 – 38, 2016, modeling and Performance Evaluation of Wireless Ad Hoc Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870516301779

[179] R. A. Uzcategui, A. J. D. Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.

[180] W. Viriyasitavat, F. Bai, and O. K. Tonguz, "Uv-cast: An urban vehicular broadcast protocol," in *2010 IEEE Vehicular Networking Conference*, Dec 2010, pp. 25–32.

[181] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," *ACM Computing Surveys*, vol. pp, 2018. [Online]. Available: http://arxiv.org/abs/1806.06191

[182] P. Wang, D. S. Reeves, and P. Ning, "Secure address auto-configuration for mobile ad hoc networks," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on.* IEEE, 2005, pp. 519–521.

[183] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," Internet Request for Comments, pp. 1 – 157, March 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6550.txt

[184] F. Ye and R. Pan, "A survey of addressing algorithms for wireless sensor networks," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, Sept 2009, pp. 1–7.

[185] L. H. Yen and W. T. Tsai, "Flexible address configurations for tree-based ZigBee/IEEE 802.15.4 wireless networks," in *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*, March 2008, pp. 395–402.

[186] S. Yessad, F. Nait-Abdesselam, T. Taleb, and B. Bensaou, "R-mac: Reservation medium access control protocol for wireless sensor networks," in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, Oct 2007, pp. 719–724.

[187] S. Yi, Z. Qin, and Q. Li, “Security and privacy issues of fog computing: A survey,” in *International conference on wireless algorithms, systems, and applications.* Springer, 2015, pp. 685–695.

[188] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE communications surveys & tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[189] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. D. Silva, “Waveform and Numerology to Support 5G Services and Requirements,” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 90–98, November 2016.

[190] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[191] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in internet of things,” *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[192] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, “Vehicular ad hoc networks (vanets): status, results, and challenges,” *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[193] J. Zhang, K. . Liu, and X. Shen, “A novel overlay token ring protocol for inter-vehicle communication,” in *2008 IEEE International Conference on Communications*, May 2008, pp. 4904–4909.

[194] K. Zhang, X. Liang, R. Lu, and X. Shen, “Sybil attacks and their defenses in the internet of things,” *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, Oct 2014.

[195] W. Zhang, R. K. Mallik, and K. B. Letaief, “Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks,” *IEEE Transactions on wireless Communications*, vol. 8, no. 12, pp. 5761–5766, 2009.