Dissertação de Mestrado

Inatel
Instituto Nacional de Telecomunicações

Random-Access Accelerator (RAA): A Framework to Speed Up the Random-Access Procedure in 5G New Radio for IoT mMTC by Enabling Device-To-Device Communications

Author: Abel Rodriguez Medel
Advisor: Jose Marcos C. Brito

December/2020

RANDOM-ACCESS ACCELERATOR (RAA): A FRAMEWORK TO SPEED UP THE RANDOM-ACCESS PROCEDURE IN 5G NEW RADIO FOR IOT MMTC BY ENABLING D2D COMMUNICATIONS

ABEL RODRIGUEZ MEDEL

Dissertation presented to the National Institute of Telecommunications, as part of the requirements for obtaining a Master's Degree in Telecommunications Engineering.

ADVISOR: Prof. Dr. Jose Marcos C. Brito

Santa Rita do Sapucaí
2020

Medel, Abel Rodriguez
M488r
Random-Access Accelerator (RAA): A Framework to Speed Up the Random-Access Procedure in 5G New Radio for IoT mMTC by Enabling Device-To-Device Communications. / Abel Rodriguez Medel. – Santa Rita do Sapucaí, 2020.
71 p.

Orientador: Prof. Dr. Jose Marcos Camara Brito.
Dissertação de Mestrado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.
Inclui bibliografia.

1. IoT 2. mMTC 3. Random-Access procedure 4. D2D 5. NR 6. Mestrado em Telecomunicações. I. Brito, Jose Marcos Camara. II. Instituto Nacional de Telecomunicações – INATEL. III. Título.

CDU 621.39

APPROVAL SHEET

Dissertation defended and approved in 21 / 12 / 2020 ,
by the judging committee:

Antonio Alfredo Ferreira Loureiro / UFMG, Departamento de Ciência da Computação
(Name/Institution)

Antônio Marcos Alberti / INATEL
(Name/Institution)

_____
Master Course Coordinator

… it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!

Lewis Carroll

ACKNOWLEDGMENT

To Professor Jose Marcos C. Brito for his orientation and contribution during the completion of this work.

To Ms. Gisele Moreira dos Santos for all her dedication and help provided throughout the master's course.

To all colleagues, professors and employees of the Telecommunications Department of the National Institute of Telecommunications for the training and their friendship.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

64QAM - 64 Quadrature Amplitude Modulation

ACB - Access Class Barring

BLE - Bluetooth Low Energy

BPSK - Binary Phase-Shift Keying

BS - Base Station

BWP - Bandwidth Parts

CBW - Carrier Bandwidth

CORESET#0 - Control Resource Set Zero

CRC - Cyclic Redundancy Check

C-RNTI - Cell-Radio Network Temporary Identifier

C-UE - Cellular User Equipment

D2D - Device-To-Device

D2D-UE - Device-To-Device User Equipment

D2I - Device-to-Infrastructure

DCI - Downlink Control Information

Dst - Destiny

eMBB - enhanced Mobile Broadband

gNB - Next Generation NodeB

GID - Group ID

GSCN - Global Synchronization Raster Channel

ID - Identifier

IoT - Internet of Things

LSB - Least Significant Bit

LTE - Long-Term Evolution

MAC - Medium Access Control

MCF - Mesh Coordination Function

MIB - Master Information Block

mMTC - massive Machine Type Communication

MSB - Most Significant Bit

No. - Number

NR - New Radio

OFDM - Orthogonal Frequency-Division Multiplexing

PBCH - Physical Broadcast Channel

PCI - Physical-layer Cell Identities

PDSCH - Physical Downlink Shared Channel

PRACH - Physical Random-Access Channel

PSS - Primary Synchronization Signal

PUSCH - Physical Uplink Shared Channel

RAA - Random-Access Accelerator

RACH - Random-Access Channel

RAPID - Random-Access Preamble Identifier

RAR - Random-Access Response

RA-RNTI - Random-Access-Radio Network Temporary Identifier

RB - Resource Block

RBG - Resource Block Group

RRC - Radio Resource Control

SCS - Subcarrier Spacing

SIB1 - System Information Block 1

SID - Sector ID

Src - Source

SSB - Synchronization Signal/PBCH Block

SSS - Secondary Synchronization Signal

SUL - Supplementary Uplink

T/C-RNTI - Temporary/Cell-Radio Network Temporary Identifier

TTL - Time-To-Live

Type0-PDCCH - Type0-Physical Downlink Control Channel

URLLC - Ultra-Reliable Low Latency Communication

ZC - Zadoff–Chu (ZC)

LIST OF SYMBOLS

* - Assigned by the requester

** - Assigned by the gNB

$cellID_X$ - Cell ID of the cell the device X belongs to

f_{ID} - index of the carrier where was received the PRACH message

$freq_{2.4}$ - Frequency from the 2.4 GHz band

$freq_{max}$ - Number of frequencies from the 2.4 GHz band for D2D communication

g - Random number generated during the execution of the ACB algorithm

K_0 - Slot offset from the slot the DCI is received by the requester for downlink

K_2 - Slot offset from the slot the DCI is received by the requester for uplink

L_{DL} - Number of symbols for downlink

L_{UL} - Number of symbols for uplink

$N_{ID}^{(1)}$ - Physical cell identities' identifier

$N_{ID}^{(2)}$ - Neighbor cells identities' identifier

N_{ID}^{Cell} - Cell identifier

$numerology$ - Numerology index

$NoSc$ - Number of subcarriers

$NoSym_{DL}$ - Number of symbols the requester needs for downlink

$NoSym_{UL}$ - Number of symbols the requester needs for uplink

P_{ACB} - Barring rate used by the ACB algorithm

RB_{DL} - Number of Resources Blocks for downlink

RB_{UL} - Number of Resources Blocks for uplink

SC_{DL} - Number of subcarriers the requester needs for downlink

SC_{UL} - Number of subcarriers the requester needs for uplink

S_{DL} - Symbol index for the first symbol offered by the gNB for downlink

S_{UL} - Symbol index for the first symbol offered by the gNB for uplink

s_{ID} - Index of the first OFDM symbol where was received the PRACH message

T_{ACB} - Barring time used by the ACB algorithm

t_{ID} - Index of the first slot symbol where was received the PRACH message

t_{sym} - Symbol duration

$\mathcal{U}$ - Range

$ul_{carrier\ ID}$ - Uplink carrier used for the MSG1 transmission

RESUMO

As redes móveis têm um grande desafio atendendo aos esperados bilhões de dispositivos de *Internet of Things* (IoT) nos próximos anos. Devido ao acesso simultâneo limitado nas redes móveis, os dispositivos devem competir entre si pela alocação de recursos durante um procedimento de acesso aleatório. Esta contenção provoca um atraso não depreciável durante o registro dos dispositivos devido ao grande número de colisões experimentadas. Para contornar tal problema, um framework denominado *Random-Access Accelerator* (RAA) é proposto neste trabalho, a fim de agilizar o acesso à rede em *massive Machine Type Communication* (mMTC). O RAA explora as comunicações *Device-To-Device* (D2D), onde os dispositivos com recursos já atribuídos atuam como relés para o restante dos dispositivos que tentam obter acesso à rede. Os resultados da simulação mostram uma aceleração no procedimento de registro de 99%, e uma liberação de espaço do espectro alocado de até 68% em comparação com o procedimento de acesso aleatório convencional. Além disso, preserva quase o mesmo consumo de energia dos dispositivos em comparação com redes legadas usando *Bluetooth Low Energy* (BLE) como tecnologia sem fio para comunicações D2D. O framework proposto pode ser levado em consideração para a padronização do mMTC na *Fifth-Generation-New Radio* (5G NR).

Palavras-chave: IoT; mMTC; *Random-Access* procedure; D2D; NR.

ABSTRACT

Mobile networks have a great challenge by serving the expected billions of *Internet of Things* (IoT) devices in the upcoming years. Due to the limited simultaneous access in the mobile networks, the devices should compete one another for resource allocation during a *Random-Access* procedure. This contention provokes a non-depreciable delay during the devices' registration because of the great number of collisions experienced. To overcome such a problem, a framework called *Random-Access Accelerator* (RAA) is proposed in this work, in order to speed up network access in *massive Machine Type Communication* (mMTC). RAA exploits *Device-To-Device* (D2D) communications, where devices with already assigned resources act like relays for the rest of devices trying to gain access in the network. The simulation results show an acceleration in the registration procedure of 99%, and a freed space of the allocated spectrum until 68% in comparison with the conventional *Random-Access* procedure. Besides, it preserves almost the same devices' energy consumption compared with legacy networks by using *Bluetooth Low Energy* (BLE) as the wireless technology for D2D communications. The proposed framework can be taken into account for the standardization of mMTC in *Fifth-Generation-New Radio* (5G NR).

Keywords: IoT; mMTC; *Random-Access* procedure; D2D; NR.

CHAPTER 1: INTRODUCTION

For a device to share data with other devices, equipment, or infrastructure—which are not physically attached to it—wireless communication technologies have arisen to establish a connection among them by means of radiofrequency resources. Just as in wired communications, the wireless approach includes control data in every transmission to assure that the connection complies a required service quality or service level agreement, security, information integrity, authentication, and authorization. There are many protocols associated with the control data to dictate the way in which the communication is established and how the devices should behave to accomplish all the requirements. When a device transmits data to another device, the information data can be transmitted and received with minimal control data alongside. However, if a third device transmits data to the other two, their transmissions could collide delaying the communication, affecting real-time data processing and draining battery reserves faster in the case of portable devices. Therefore, the control algorithm must be enhanced to avoid two or more devices transmitting at the same time.

Medium Access Control (MAC) is one of the most important aspects of communication networks. It is in charge of the coordination of multiple entities sharing the same physical channel to avoid or reduce the collisions in it (Zhang, Peng 2008). In the case of mobile networks, the MAC method employed is the *Mesh Coordination Function* (MCF), which is based on a *Contention Channel Random-Access* procedure where limited management frames are reserved for future transmissions. The neighboring devices listen to the reservations and do not transmit during the reserved periods (Bensky 2019). Once the reserved period ends, the neighboring devices can each reserve a frame. Thus, it is possible that more than one device tries to make the same reservation at the same time provoking a collision and consequently not reserving a frame for either device. One more time, the devices should wait for the current reserved period to end. As the number of devices trying to reserve a frame increases, the probability of collisions also increases and the waiting time becomes longer (Larmo and Susitaival 2015). In addition, the current networks employ a *back-off* algorithm to randomly extend the waiting times of every device (Tello-Oquendo et al. 2018). That time can even be in the order of seconds, which helps avoid collisions, but increases the waiting time even more. Therefore, due to the exponential growth of connected stations, especially IoT devices (Alsaeedy and Chong 2019), the mobile networks will face a difficult challenge handling the medium access to reduce the access time of the devices in the upcoming years.

In addition, the 5G network will use higher frequencies for communication in order to expand the available bandwidth to support massive communication scenarios like mMTC (Zaidi et al. 2018). The use of such frequencies leads to more propagation

losses and, as a consequence, the communication distance should be shorter to avoid signal degradation ("5G Evolution Towards a Super-connected World" 2018). Therefore, the cell size will be smaller than in legacy networks, which increases the probability of a device crossing the limits of the cell. In the new cell where the device has camped, the device must perform the *Random-Access* procedure to register in the network again. This procedure is not efficient because there are limited vacancies to connect to the network. Only 64 vacancies (preambles) are periodically available for all the devices trying to gain access to the network (Schreiber and Tavares 2018). In mMTC, the number of devices is much higher than the number of available preambles. Many of these devices would select the same preamble. Every preamble is associated with a channel. The devices that select the same preamble will send back the selected preamble to the gNB (5G Base Station) in the same channel, thus provoking collision (Han et al. 2019). The more devices trying to gain access at the same time, the higher the collision probability, the higher the access delay (Choi 2020), and the higher the energy consumption.

The applications related to motion are the most affected by the discussed problem. The more mobility a device running these kinds of applications has, the more probability of crossing cell limits and a greater number of attempts for a device to have access. Some examples of those applications are factory automation, where a set of machines are in motion; autonomous driving, where cars are expected to cross the coverage area limits very often (Schulz et al. 2017); wearable devices that can be carried by a person along its way to notify the police about criminal incidents as rapes or assaults (Jatti et al. 2016); and fitness tracking devices that measure calories and heart rates along the athlete's trajectory (Zhou and Piramuthu 2014).

In 5G, the minimum unit of assigned resources per device is a *Resource Block* (RB). Every RB has 12 subcarriers (Lien et al. 2017) and each subcarrier has 14 symbols in *New Radio* (NR) (Parkvall et al. 2017). Therefore, there are $12 \times 14 = 168$ symbols within an RB. In the case of the IoT devices, most of them are sensors, and they require a few symbols to transmit/receive their data. For example, an IoT sensor that checks the room temperature in the wide range $0\ ^0$F to $100\ ^0$F will need 7 bits to cover the 101 possibilities. If the device uses *Binary Phase-Shift Keying* (BPSK) modulation where every symbol is composed of 2 bits, the device needs to transmit 4 symbols to send its temperature measurements. Therefore, 168 symbols are an excessive amount for the IoT sensor demands. Thus, the gNB will allocate 164 unused symbols to the IoT sensor.

The D2D communications allow that near devices exchange their unused resources to other devices. For a device to transmit part of its resources to another device, both devices need to agree on the channel they will use for the resource's exchange. The agreement procedure is influenced by the number of other agreements procedures that happen at the same time and in the same location. If more than a pair of devices agree on the same channel, their transmissions will collide. The D2D technologies were originally designed to avoid collisions, which means that collisions exist but the transmission/reception is based in reattempts until successfulness. In massive communications, the D2D procedures for the channel agreement and successful communication are prohibitively delayed due to the high device's concurrence. The delay implies more transmission attempts and consequently more energy consumption. Therefore, the current D2D technologies need to be readjusted

and assessed to select the ones that comply with both low latency channel agreement and low energy consumption during communication in the mMTC scenario.

In this work, RAA is proposed as a framework to improve the *Random-Access* procedure of the 5G network. The main goal is to reduce the waiting times during reserved periods by more efficiently handling the number of accessing attempts in the network. The procedure integrates not only the communication between the devices and the *Base Station* (BS), but also the D2D communications. The devices with already assigned resources by the network will serve as relays for new devices trying to access the network. Thus, the registered devices use their allocated resources to forward the pretending reservations of the new devices to the BS. Therefore, most collisions occur in D2D communications, and not in *Device-to-Infrastructure* (D2I) communication. The D2D technologies explored are Classic Bluetooth, BLE, and Wi-Fi. All of them were tested using an app and a simulation tool to verify their suitability for the proposed framework. In addition, RAA conserves the energy spent by the devices during the entire process in comparison with the traditional approach by using a D2D technology. Although the D2D communications allow new transmissions, they accelerate the *Random-Access* procedure enough to compensate the energy consumption employed in those transmissions. Furthermore, RAA allows that the network resources are shared more efficiently between all the devices than in legacy networks.

1.1 BACKGROUND AND MOTIVATION

In order to resolve the limitations raised before, some research propose mechanisms based on D2D communications as an alternative technology to assist the access demand. Vilgelm, Linares, and Kellerer 2017 introduced a new algorithm for contention resolution, called *Binary Countdown*. This algorithm is executed after receiving the resource allocation for the transmission of the *Connection Request* message, and before sending the *Connection Request* message to the gNB, which avoids unnecessary uplink communication (D2I) if a preamble collision has occurred. Every device will generate a random sequence of 0 and 1 with a variable length depending on the network load. The generated sequence is shared between nearby devices by a *sidelink* (D2D communication). Then, every device checks its sequence and compares it with the sequences arriving from nearby devices. If the current element from a device's sequence has the same priority as at least one element from the rest of the sequences, the device will continue with the contention resolution procedure. If the current element from a device's sequence has the highest priority and the rest of the elements from the other sequences has the lowest priority, the device wins the contention. However, due to the coverage restriction of the D2D technologies (Wang and Rouil 2018), all the devices in the cell do not communicate with each other. Then, it is possible that more than one device wins the contention in different groups and select the same preamble. Therefore, this procedure is based on the best effort to let devices gain access in the least possible time by reducing the

collision probability. The principal limitation of this solution is the high number of *sidelink* communications between the devices that want to transmit a preamble for resource allocation using the *traditional D2D wireless technologies*. This could drain the *device's battery* faster and could increase the number of collisions in a conflictive spectrum space like the 2.4 GHz band. Besides, the gNB must transmit an *extra message* periodically informing the devices of the sequence's length for the next contention, which increases the processing load of the network. Unlike the work presented by Vilgelm, Linares, and Kellerer 2017, with RAA there is no agreement between devices to select the device or devices that will execute the *Random-Access* procedure and the ones that will not do it. Therefore, RAA allows that all devices have the same priority during the access procedure.

Due to the minimum unit of assigned resources per device is an RB in 5G, the number of symbols a device needs could be less than the total of symbols an RB has. The rest of the symbols unused by the device could be delegated to another device that needs it. Thus, it is introduced the D2D communications for resource delegation by Soleymani et al. 2016. That work proposes a framework which is divided into two main algorithms, one of them is for the device that requests resources (called *Cellular User Equipment* C-UE), and the other one is for the device that offers resources (denominated *Device-To-Device User Equipment* D2D-UE). The approach consists of the gNB updating a list with all possible devices that need resources, and sharing the list with the providers (devices with already assigned resources). The list is different for every provider because it contains only devices in the vicinity of the provider. One of the constraints of this solution is the *extra memory* a D2D-UE needs to have to store exclusively all its C-UE neighbors. The D2D-UE also needs to communicate very frequently with the gNB to update the C-UE neighbor's list, which could drain the *device's battery* faster. In this scenario, the gNB must transmit an *extra message* periodically to update the identification of the nearby devices requesting resources in the connected devices list, which increases the processing load of the network. Contrasting with the approach exposed by Soleymani et al. 2016, RAA does not allow that nearby devices offer their unused resources, instead, only the gNB allocates the resources for each device. The devices with allocated resources are intermediaries between the devices looking for access and the gNB. The gNB receives the new devices' demands via the intermediary devices with allocated resources in the mobile network. Then, the resource allocation is done by the gNB over a trustable channel and not over the unlicensed spectrum channels, in order to avoid collisions and reduce the resource allocation time.

A D2D-based *Random-Access* technique is introduced by Han et al. 2019. It transfers the possible access congestion between the devices and the network to the *sidelink* communications between nearby devices. The access delay is significantly reduced, but it is not conceived the possibility of *sharing resources* in case some devices have more allocated resources than needed.

Vikhrova et al. 2019 considered various resource allocation strategies to more effectively handle the access in the mobile network for different slices: *enhanced Mobile Broadband* (eMBB), *Ultra-Reliable Low Latency Communication* (URLLC), and mMTC. Unlike the work presented by Vikhrova et al. 2019, the RAA procedure does not differently handle the types of slices. Thus, the eMBB, URLLC, and mMTC slices are treated with the same priority level. Choi 2020 introduces a 2-step *Random-*

Access approach instead of the 4-step conventional procedure by sending both control and information data in the same message. Unlike the work presented by Choi 2020, RAA is executed in parallel with the original traditional 4-step *Random-Access* approach in 5G. The approaches presented by Choi 2020 and Vikhrova et al. 2019 are focused only on the access blocking probability. They do not show the *elapsed time* for all devices' registration and there is no consideration of the *energy consumption* during the slicing procedure. Contrasting with those approaches, the RAA procedure presented in this work assesses the *elapsed time* for all devices' registration and considers the *energy consumption* of the devices and the gNB during the access procedure. In addition, the approach proposed by Choi 2020 can negatively influence the devices' energy consumption because—in case of collision—the devices spend both control and information data energy in infructuous transmissions.

RAA exploits the D2D communications to achieve low latency access to the network. In this work, BLE is proposed as the D2D technology used by RAA in order to avoid collisions and accelerate the discovery procedure between nearby devices. The devices that want to be registered in the mobile network, also called requesters, send their resources' demands to the nearby devices with allocated resources, named relays. The relays are forwarding devices; they retransmit the requesters' demands to the gNB. Then, the gNB handles two flows of resource requests: one from the traditional *Random-Access* procedure and the other from the RAA procedure. In the last case, the message containing the resources' demands from the new devices includes the number of subcarriers and symbols needed for the devices to transmit/receive data. With this information, the gNB fits the exact demand into the spectrum and time resources. Therefore, the gNB can manage resource allocation better than traditionally. As will be proofed in this work, with the proposed RRA, the elapsed time during registration is reduced by 99% in comparison to the traditional *Random-Access* approach. Due to the fast access experienced by the devices, the devices' energy consumption remains almost the same as in the traditional *Random-Access* procedure.

1.2 RESEARCH PROBLEM

In this work are considered the following research problems: The non-standardization of an algorithm for efficiently handle the *Random-Access* in mMTC scenarios as part of the 5G network; as well as the lack of a *Random-Access* procedure that agglomerates low access latency by avoiding access barring in the order of seconds, low energy consumption by giving the devices other communication alternatives like the D2D communications, and efficient resource allocation due to the traditional *Random-Access* procedure offers an entire RB to one device regardless of the device needs fewer resources than an RB.

1.3 HYPOTHESIS

If the *Random-Access* procedure introduces the BLE technology to handle the access between the devices trying to connect to the network and the devices with allocated resources, it is possible to achieve lower access latency and more efficient resource allocation than in the traditional approach.

1.4 OBJECTIVES

The main objective of this work is to propose a new *Random-Access* procedure to reduce the access latency and handle more efficiently the resource allocation in 5G for mMTC. To achieve the latter, the following specific objectives are considered:

- Assess the performance of three D2D technologies (Wi-Fi Direct, Classic Bluetooth, and BLE) in massive communications about elapsed time and energy consumption during the devices' discovery.
- Select the D2D technology that incurs in the lowest latency and the lowest power consumption during the devices' discovery, to be part of the new *Random-Access* procedure.
- Implement an algorithm for the communication between the devices looking for resources and the devices with allocated resources.
- Implement an algorithm for the communication between the devices with allocated resources and the gNB during the *Random-Access*.
- Establish how the communication between the gNB and the new devices happens.
- Use both the traditional *Random-Access* approach and the new *Random-Access* procedure (RAA) in parallel to accelerate even more access.
- Do not get rid of any original NR function in order to adapt the new *Random-Access* procedure faster and without incompatibility.

1.5 DOCUMENT STRUCTURE

This document is organized as follows. Chapter 2 describes the traditional *Random-Access* procedure and its limitations in detail. In Chapter 3, are introduced the possible D2D technologies to be used in the new *Random-Access* procedure. Chapter 3 also shows the assessment of the D2D technologies' performances during the devices' discovery in order to select the D2D technology with best performance in massive communications. Chapter 4 introduces the proposed framework to be used as part of a new *Random-Access* procedure (RAA) with all its features and a detailed example where some devices try to connect to the mobile network. Chapter 5 presents some simulation scenarios to compare the traditional *Random-Access* approach and the proposed RAA procedure performances. Last, the conclusions of this work and some recommendations are given.

CHAPTER 2: NR RANDOM-ACCESS PROCEDURE

This chapter covers the entire procedure for a device to gain access in the mobile network since the device enters an NR cell until it obtains downlink and uplink resources from the network. However, the focus is the NR *Random-Access* procedure. Then, it is exposed the principal limitations of the traditional *Random-Access* procedure as a result of the technologies and the built-in functions used.

2.1 SYNCHRONIZATION AND RANDOM-ACCESS

Cell search is the first thing a device does in either of the following scenarios: a device is turned ON within an NR cell; it moves from a *Long-Term Evolution* (LTE) cell to an NR cell and vice versa; or it does handover. The device needs to synchronize in frequency and time with the cell it has camped. Synchronization is acquired via the *Synchronization Signal/PBCH block* (SSB) sent by the gNB in 240 subcarriers (20 RBs) of the carrier associated with the SSB. SSB is transmitted with a periodicity of 5 ms, 10 ms, 20 ms, 40 ms, 80 ms, or 160 ms. A device can assume a default periodicity of 20 ms while cell searching. SSB includes the *Primary Synchronization Signal* (PSS), the *Secondary Synchronization Signal* (SSS), and the *Physical Broadcast Channel* (PBCH). SSB consists of 4 symbols in the time-domain (Chandramouli, Liebhart, and Pirskanen 2019). **Figure 1** shows SSB in frequency and time-domain.

One of many use-cases of NR is the transmission/reception in the mm-Wave band. This particular band has high path-loss. Therefore, antennas in NR must have high directional transmissions. A device that camps in a cell does not know in which direction to receive the gNB information, and it and the gNB perform beam-sweeping over different directions (Mazin, Elkourdi, and Gitlin 2018). This procedure is depicted in **Figure 2** for two different devices. From the figure, the x-axis represents the SSB signals sent by the gNB in different directions along the time. The top x-axis belongs to the device UE#1 while the bottom x-axis belongs to the device UE#2. The y-axis represents the SSB signal strength. In the case of the device UE#1, this device is in the direction of the orange beam of the gNB; therefore, UE#1 perceives that the strongest signal is the signal sent by the gNB in the direction of the orange beam. The same occurs for the device UE#2; in this case, UE#2 is in the same direction than the purple beam which is the strongest signal noticed by UE#2. Now, the devices know the exact direction to receive all the signals that will be send by the gNB.

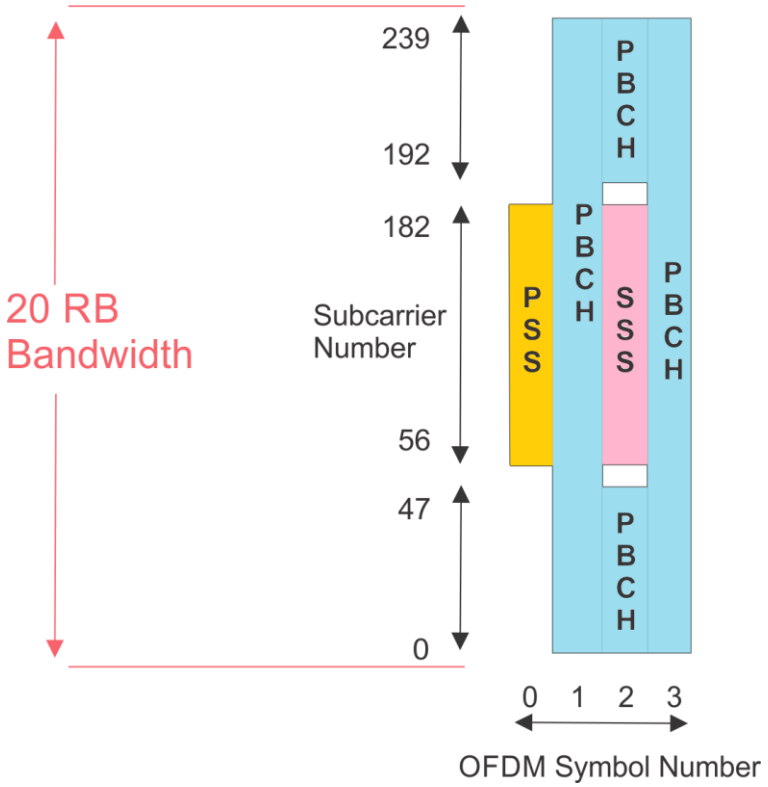


Figure 1. *SSB structure in frequency and time-domain.*

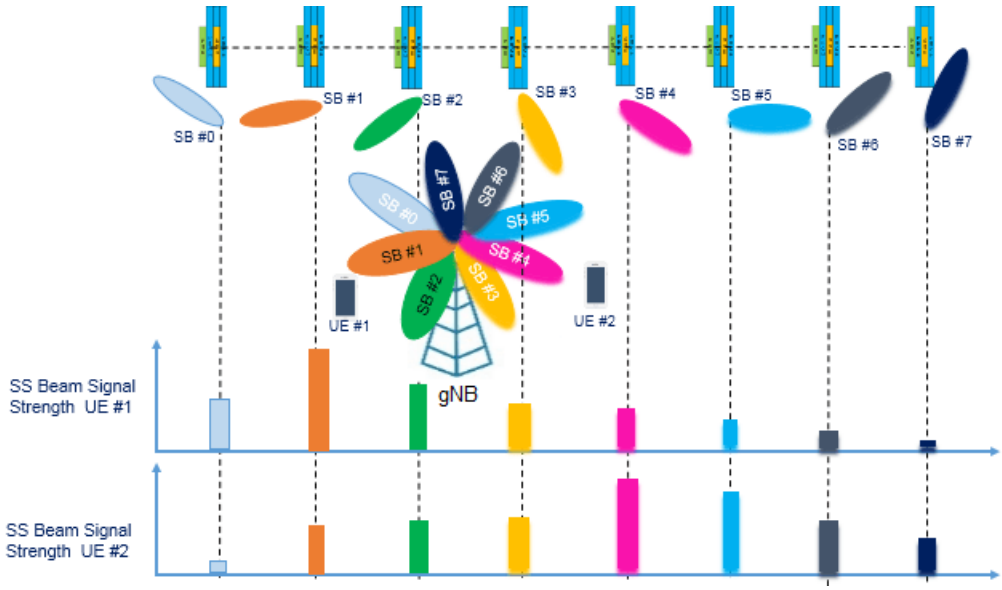


Figure 2. *Beam-sweeping between two devices and the gNB.*

2.1.1 FREQUENCY SYNCHRONIZATION

Frequency synchronization is carried out with a carrier polling by the device in every beam direction and band. This polling is executed by scanning the PSS signal in predefined carriers. The hops between one carrier and the next one while polling is called channel raster. The channel raster is proportional to the bandwidth. Greater bandwidth means greater channel raster. Otherwise, the terminal searching for synchronization has to iterate for a lot of possible frequencies to detect the carrier with the PSS signal, resulting in high energy consumption; also, it takes a too long time (Wei and Moulsley 2019). Channel raster is defined by the *Global Synchronization Raster Channel* (GSCN). The GSCN granularity is 50 or 150 or 250 kHz under 3 GHz, and it has the granularity of 1.44 MHz in above 3 GHz frequency range and below 24.25 GHz; the granularity jumps to 17.28 MHz if the frequency goes over 24.25 GHz. **Table 1** shows the GSCN parameters for the global frequency raster. GCSN is with reference to the central frequency of SSB based on the statement in the 3GPP TS 38.104 version 15.2.0 Release 15 (section 5.4.3.2) (3GPP 2018a).

Table 1. *GSCN parameters for the global frequency raster.*

Frequency range	**SS block frequency position SS_{REF}**	**GSCN**	**Range of GSCN**
0 – 3000 MHz	N×1200 kHz + M×50 kHz, N=1:2499, M ∈ {1,3,5}	3N + (M-3)/2	2 – 7498
3000 – 24250 MHz	3000 MHz + N×1.44 MHz, N=0:14756	7499 + N	7499 – 22255
24250 – 100000 MHz	24250.08 MHz + N×17.28 MHz, N=0:4383	22256 + N	22256 – 26639

In NR, the SSB is not transmitted at frequency 0 of the carrier, also called *point A*. Instead, the SSB is transmitted at different frequency offsets from carrier *point A* for every neighbor cell (Lei et al. 2019). This is useful to avoid synchronization signal interference between neighbor cells. In this case, the device search for the PSS signal in the channel raster defined in **Table 1**. With the PSS signal detection, the device identifies neighbor cells belonging to one of three different identities $N_{ID}^{(2)} = \{0, 1, 2\}$. Once it has found the PSS signal for a specific carrier, the device looks for the SSS signal in the same frequencies as PSS but one *Orthogonal Frequency-Division Multiplexing* (OFDM) symbol duration later. With the SSS signal detection, the cell identification finishes. The SSS signal identifies one of the 336 *Physical-layer Cell Identities* (PCI) $N_{ID}^{(1)} = \{0, 1, \dots, 335\}$. The cell identifier is calculated as: $N_{ID}^{Cell} = 3 * N_{ID}^{(1)} + N_{ID}^{(2)}$ (Kryukov, Pokamestov, and Rogozhnikov 2019). At this point, the device is synchronized with cell N_{ID}^{Cell} in frequency.

However, the device does not know the carrier start point associated with the discovered SSB. First, it looks for the offset from the SSB frequency to the subcarrier beginning where SSB was transmitted. The subcarrier offset is obtained from the *Master Information Block* (MIB) parameter *ssb-SubcarrierOffset* (MIB is part of the PBCH within the SSB and is transmitted periodically each 80 ms, and is repeated within that interval) (Chakrapani 2020). Then, the device needs to obtain the offset related to the carrier *point A*. This offset is called *offsetToPointA* and is part of the *System Information Block 1* (SIB1) (SONG and Yi 2020). **Figure 3** shows the offsets to *point A*.

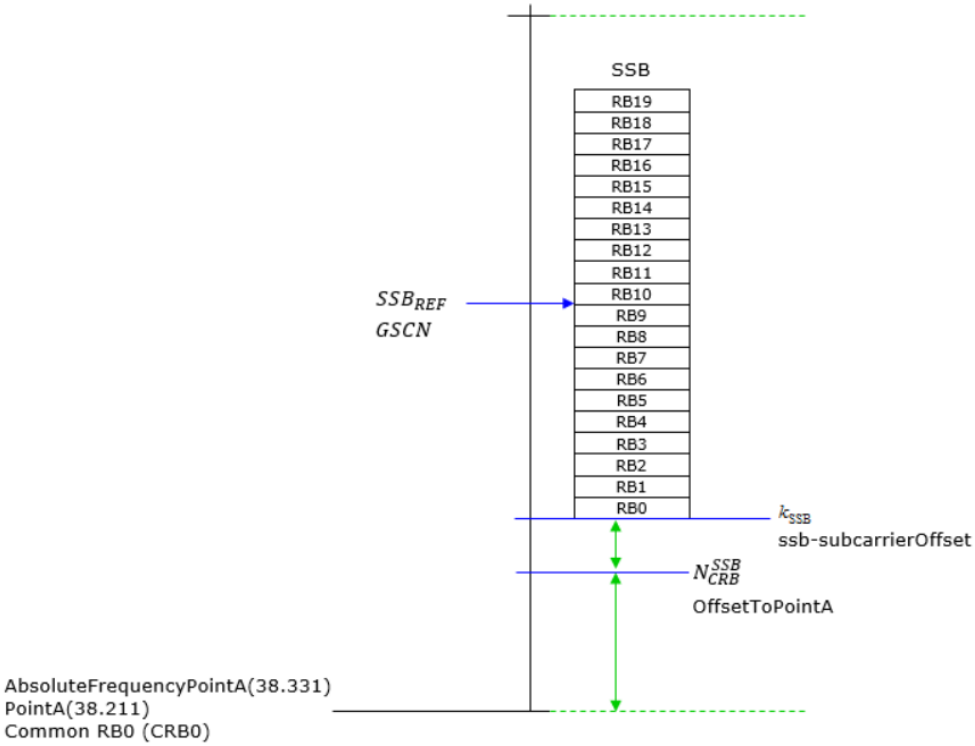


Figure 3. *SSB reference position, Point A, offsetToPointA, and ssb-SubcarrierOffset.*

2.1.2 TIME SYNCHRONIZATION

For time synchronization, instead of transmitting one SSB periodically, the gNB sends a set of SSB. This set is denominated *SS Burst Set*. Each SSB within the *SS Burst Set* matches a different beam. An *SS Burst Set* involves a variable number of SSB in concordance with the *Subcarrier Spacing* (SCS) used in the NR cell. The number of SSB within a *SS Burst Set* is proportional to the SCS. The *SS Burst Set* is transmitted completely within 5 ms (Chandramouli, Liebhart, and Pirskanen 2019). **Figure 4** depicts the *SS Burst Set* in the time-domain.

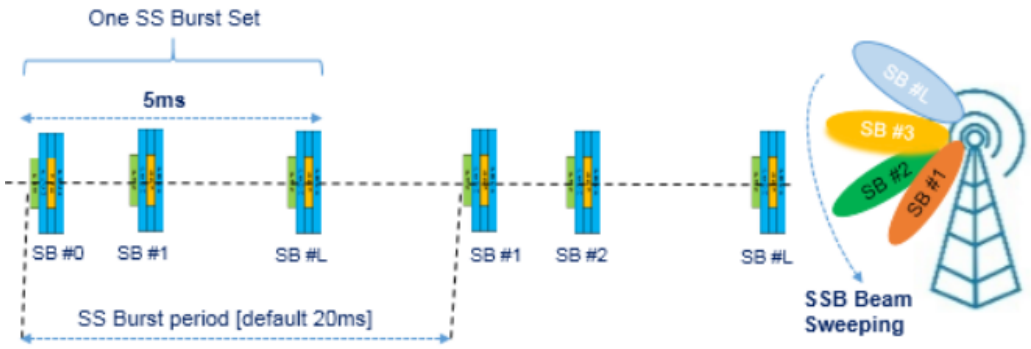


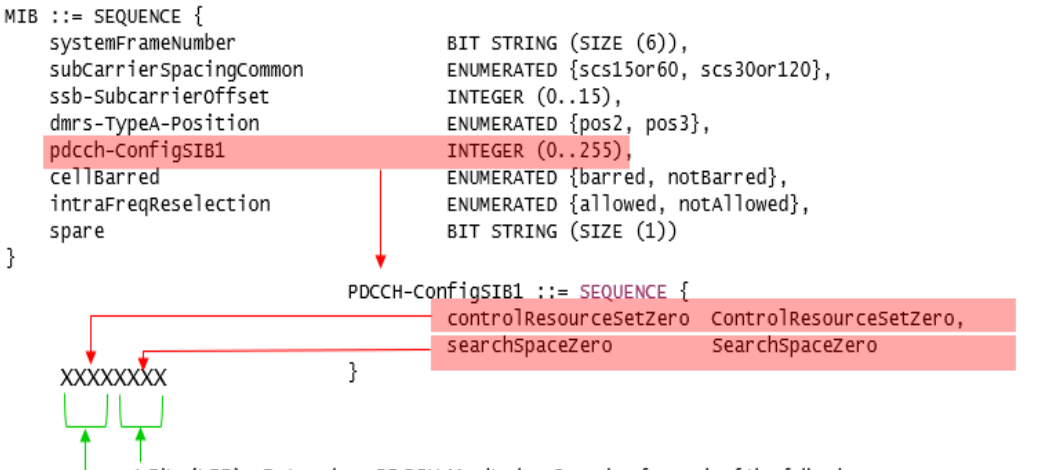
Figure 4. *SS Burst Set in time-domain while beam-sweeping.*

The PSS is transmitted in the first symbol of the SSB, PBCH is transmitted in the second and fourth symbol, and SSS is transmitted in the third symbol (Omri et al. 2019). A device searching a cell has three correlators for PSS, SSS, and PBCH respectively. If the correlators detect PSS, the device can calculate the time slot used for the current numerology by the duration of the symbol. This is possible because PSS is BPSK modulated and one bit corresponds to one symbol (Lin et al. 2019). Therefore, the duration of the power level of a PSS bit matches the duration of the OFDM symbol used by the gNB downlink transmission. The device knows the SSS timing by adding an OFDM signal duration at the end of the PSS symbol transmission. Now, the device is synchronized in time.

2.1.3 SIB1 DECODING

After synchronization, the device needs a *Physical Random-Access Channel* (PRACH) configuration. This information is carried out by SIB1. SIB1 is transmitted with a periodicity of 160 ms and variable transmission repetition within that interval (Lei et al. 2019). With the information offered by SIB1, the device is able to attempt to obtain resources from the gNB by performing the *Random-Access* procedure.

The device does not know how to acquire SIB1. MIB provides the device with parameters required to decode SIB1. In MIB, the field *subCarrierSpacingCommon* indicates the subcarrier spacing used for SIB1, and the field *pdcch-ConfigSIB1* indicates the frequency positions where the device may find SIB1 or the frequency range where the network does not provide SIB1. *pdcch-ConfigSIB1* carries the parameters indicating the location and resources for *Control Resource Set Zero* (CORESET#0) on the resource grid, where the device needs to search for the *Type0-Physical Downlink Control Channel (*Type0-PDCCH) Common Search Space to derive the SIB1 information (Xu et al. 2019). **Figure 5** shows the parameters offered by *pdcch-ConfigSIB1*.



4 Bits (LSB) : Determines PDCCH Monitoring Occasion for each of the following cases.

CORESET Multiplexing Pattern	Frequency Range	SSB SCS (Khz)	PDCCH SCS (Khz)	Table
Pattern 1	FR 1	N/A	N/A	38.213 - Table 13-11
Pattern 1	FR 2	N/A	N/A	38.213 - Table 13-12
Pattern 2	N/A	120	60	38.213 - Table 13-13
Pattern 2	N/A	240	120	38.213 - Table 13-14

4 Bits (MSB) : Defines a CORESET for each of the following cases. 38.101-1 Table 5.3.5-1

SSB SCS (Khz)	PDCCH SCS(Khz)	Min BW (Mhz)	Table
15	15	5	38.213 - Table 13-1
15	30	5	38.213 - Table 13-2
30	15	5 or 10	38.213 - Table 13-3
30	30	5 or 10	38.213 - Table 13-4
30	15	40	38.213 - Table 13-5
30	30	40	38.213 - Table 13-6
120	60	N/A	38.213 - Table 13-7
120	120	N/A	38.213 - Table 13-8
240	60	N/A	38.213 - Table 13-9
240	120	N/A	38.213 - Table 13-10

Figure 5. *pdcch-ConfigSIB1 and derivate parameters.*

From **Figure 5**, the first 4 bits (MSB) will determine the CORESET#0 Index, this will indicate the Number of RB/symbols used to determine the CORESET size of the Type0-PDCCH common search space. These parameters are offered by tables 38.213 – 13.1 to 38.213 – 13.10 from the 3GPP TS 38.213 version 15.3.0 Release 15 (section 13) (3GPP 2018c). Error! Reference source not found. depicts Table 38.213 – 13.1 as an example.

With the information given by the three last columns of those tables, the CORESET#0 can be mapped in frequency and time-domain. **Figure 6** shows the frequency domain location of the CORESET#0 and the associated SSB.

Table 2. *Set of resources blocks and slot symbols of control resource set for Type0-PDCCH search space when {SS/PBCH block, PDCCH} subcarrier spacing is {15, 15} kHz for frequency bands with minimum channel bandwidth 5 MHz or 10 MHz.*

Index	SS/PBCH block and control resource set multiplexing pattern	Number of RBs $N_{RB}^{CORESET}$	Number of symbols $N_{SYMB}^{CORESET}$	Offset (RBs)
0	1	24	2	0
1	1	24	2	2
2	1	24	2	4
3	1	24	3	0
4	1	24	3	2
5	1	24	3	4
6	1	48	1	12
7	1	48	1	16
8	1	48	2	12
9	1	48	2	16
10	1	48	3	12
11	1	48	3	16
12	1	96	1	38
13	1	96	2	38
14	1	96	3	38
15	Reserved			

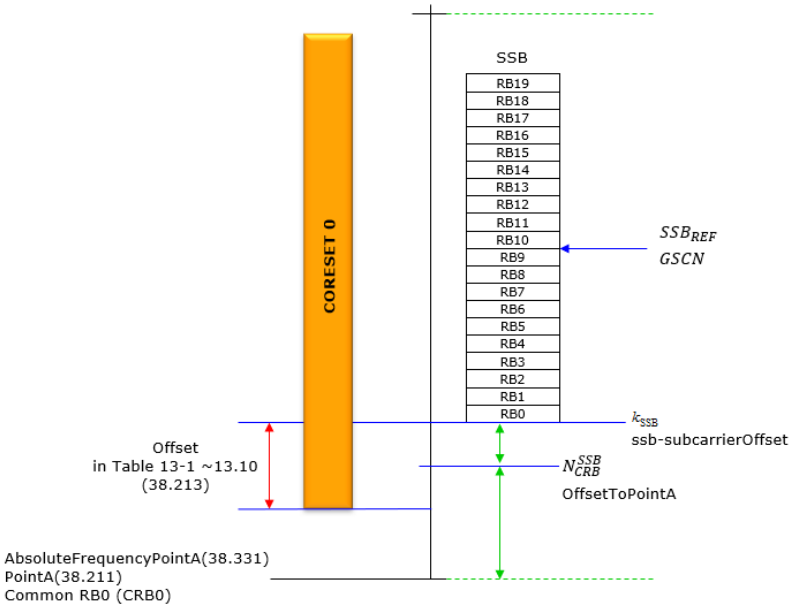


Figure 6. *CORESET#0 and associated SSB in the frequency domain.*

The second column in tables 38.213 – 13.1 to 38.213 – 13.10 gives the multiplexing pattern. It is used to select one of the tables 38.213 – 13.11 to 38.213 – 13.14, which offer the search space in the time-domain for Type0-PDCCH (3GPP 2018c).

Other CORESETs are mapped in the rest of Bandwidth Parts (BWP). BWP represents one of the ranges of frequencies within the whole Carrier Bandwidth (CBW) and is configured in SIB1. In NR, there are up to 4 BWP (Jeon 2018).

Unlike LTE, in NR the CORESETs are not distributed along the whole CBW because the bandwidth is longer than in LTE and the device can take a lot of time and energy to find the principal channels and system information blocks (Jeon 2018). **Figure 7** shows an example of CORESETs mapping in the CBW.

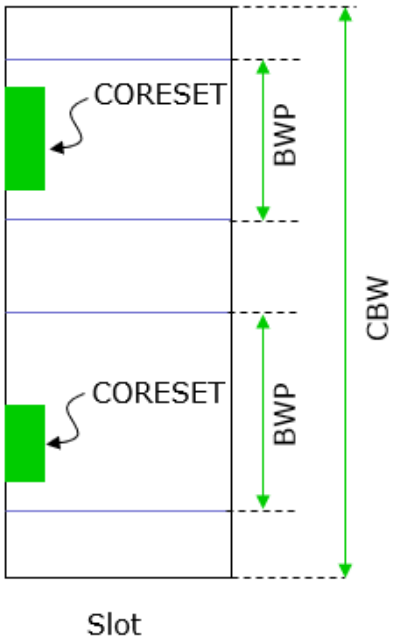


Figure 7. *CORESETs distributed within Bandwidth Parts (BWP) along the whole Carrier Bandwidth (CBW).*

2.1.4 CONTENTION-BASED RACH PROCEDURE

The contention-based *Random-Access* procedure or *Random-Access Channel* (RACH) procedure in the 5G NR network is summarized in **Figure 8**. After synchronization and SIB1 decoding, the device obtains the PRACH configuration which contains the available preambles and an access probability to execute the *Random-Access* procedure. The preambles are like tickets for the devices to obtain resources from the network. However, the number of preambles is limited to 64 and more than one device could select the same preamble or ticket. In order to reduce the number of devices that select the same preamble, the traditional *Random-Access*

procedure employs the *Access Class Barring* (ACB) algorithm. ACB consists of the device generating a random number (g) between 0 and 1 (Tello-Oquendo et al. 2018). If the generated number is equal or smaller than the access probability (P_{ACB}) sent by the gNB, the device can access the network.

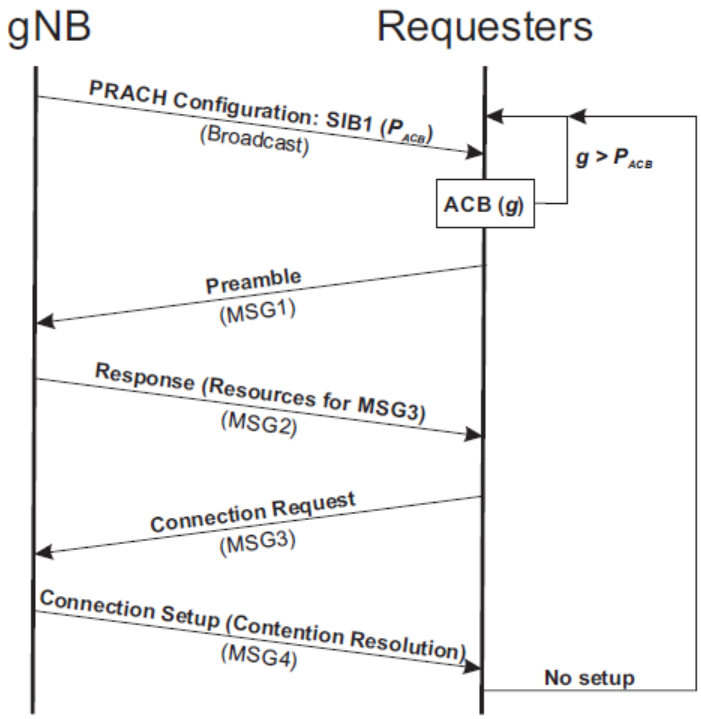


Figure 8. *New Radio Random-Access procedure.*

Once the device is allowed to access the network, it selects one of the preambles sent by the gNB in the PRACH configuration message (Vilgelm, Rueda Liñares, and Kellerer 2019). The device transmits the selected preamble to the gNB in a message called MSG1 or *Random-Access Request*. MSG1 is transmitted in the frequency domain location determined by the parameters *msg1-FDM* and *msg1-FrequencyStart* (KIM and YUN 2020), and in the time-domain location determined by the parameter *prach-ConfigurationIndex* (RYU et al. 2020). The MSG1 message includes a *Random-Access-Radio Network Temporary Identifier* (RA-RNTI). The RA-RNTI temporally identifies the device that is executing the *Random-Access* procedure. The device computes the RA-RNTI as follow:

$$RA - RNTI = 1 + s_{ID} + 14 \times t_{ID} + 14 \times 80 \times f_{ID} + 14 \times 80 \times 8 \times ul_{carrier\ ID}$$

where s_{ID} is the index of the first OFDM symbol where was received the PRACH message ($0 \leq s_{ID} < 14$) in the time-domain; t_{ID} is the index of the first slot symbol where was received the PRACH message ($0 \leq t_{ID} < 80$) in the time-domain; f_{ID} is the index of the carrier where was received the PRACH message in the frequency domain ($0 \leq f_{ID} < 8$); and $ul_{carrier\ ID}$ is the uplink carrier used for the MSG1

transmission (0 = normal carrier, 1 = *Supplementary Uplink* (SUL) carrier) (3GPP 2018b). Now, the device will listen to the Type1-PDCCH Common Search Space to receive MSG2 during an *ra-ResponseWindow* interval (Enescu 2020). If the *ra-ResponseWindow* interval expires and the device has not allocated resources yet, the device restarts the *Random-Access* procedure by listening to a new PRACH configuration after sending the next MSG1.

After the gNB receives MSG1, the gNB broadcasts a message (MSG2) called *Random-Access Response* (RAR) in response to the preamble transmission. This message contains resource allocation for the transmission of the *Radio Resource Control Connection Request* message (MSG3) (Chandramouli, Liebhart, and Pirskanen 2019). MSG2 is scrambled with the RA-RNTI sent in MSG1 by the device.

If the device successfully decodes the Type1-PDCCH, the device decodes the *Physical Downlink Shared Channel* (PDSCH) carrying the MSG2 data. The MSG2 packet is depicted in **Figure 9**. As it can be seen, the MSG2 message contains the *Random-Access Preamble Identifier* (RAPID) for the device to check that the message is a response to the requested preamble, the uplink grants (red) for the transmission of MSG3 and a new identifier for the device called *Temporary/Cell-Radio Network Temporary Identifier* (T/C-RNTI) (Enescu 2020).

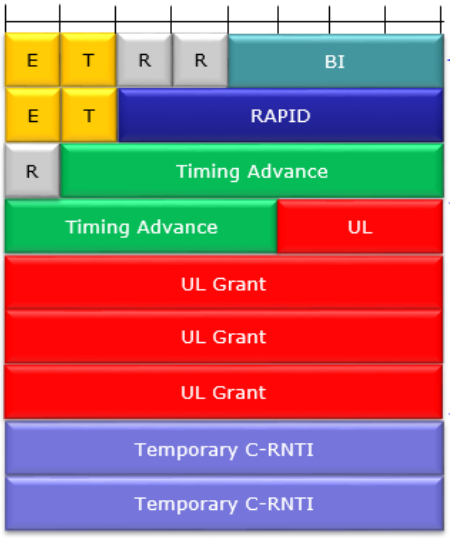


Figure 9. *RAR message (MSG2) sent by the gNB.*

After receiving MSG2, the device sends MSG3 to the gNB in the *Physical Uplink Shared Channel* (PUSCH). Then, the device will listen to the Type1-PDCCH Common Search Space to receive MSG4 during an *ra-ContentionResolutionTimer* interval (Enescu 2020).

When the gNB receives MSG3, the gNB sends a *Radio Resource Control Connection Setup* message (MSG4) in response to the connection requests including allocated resources for the device. MSG4 acts like a contention resolution message. It assigns a 40-bit Identifier (ID) to identify only one device from a group of devices that both selected the same preamble and transmitted at different times. In that case,

the gNB receives multiple requests for the same preamble but the gNB only replies to one device (Althumali and Othman 2018).

If the device successfully decodes MSG4 with the device's T/C-RNTI, the device sets the *Cell-Radio Network Temporary Identifier* (C-RNTI) equal to the T/C-RNTI. The C-RNTI is a permanent identifier that identifies the device in the cell it has camped. At this point, the *Random-Access* procedure is successfully completed for the device (Enescu 2020).

2.2 RANDOM-ACCESS PROCEDURE CONSTRAINTS

In 5G, there are only 64 available preambles to reserve resources. The limited preamble number is a consequence of the use of a *Zadoff–Chu* (ZC) signal generator. The ZC generates orthogonal preambles with zero correlation, which avoids inter-signal interference. However, the generation process is difficult to perform in real-time and requires a large amount of memory to store the sequences (Seo, Hong, and Choi 2019; Leyva-Mayorga et al. 2019). Therefore, the shortage of preambles and the great number of devices in massive communications derivate in very long periods of blackouts (no connection with the network) and many missed transmission opportunities.

After checking SIB1 information, devices know all preambles they can get for resource allocation. All devices select one of the preambles and send it to the gNB to request resources. Then, if more than one device selects the same preamble, they transmit in the frequency associated with the selected preamble. If both devices also transmit at the same time, their transmissions will collide because they are using the same channel (Hussain, Anpalagan, and Vannithamby 2017). However, collision is not detected yet. The devices that sent their preambles wait for a RAR message during an *ra-ResponseWindow* interval. If no responses arrive in the *ra-ResponseWindow* period, the devices know a collision has occurred (Chen, Shih, and Chou 2020).

The number of collisions is reduced when the ACB algorithm is executed. ACB limits the number of simultaneous access attempts from devices that want to connect to the network. In this case, the devices use two types of information sent by the gNB within the SIB1 message to execute the ACB procedure: Barring rates $P_{ACB} \in \{0.05, 0.1, ..., 0.3, 0.4, ..., 0.7, 0.75, 0.8, ..., 0.95\}$, and barring times $T_{ACB} \in \{4, 8, 16, \ldots, 512$ s$\}$. Then, every device determines its barring status. The devices pick their corresponding P_{ACB} and T_{ACB} from the lists above based on their classes (the class a device belongs is not important at this point). The devices generate a random number $g = \mathcal{U}[0, 1)$. If $g \leq P_{ACB}$, the devices transmit a selected preamble; otherwise, the devices wait for a random time (*back-off*) calculated as $T_{barring} = [0.7 + 0.6 \times \mathcal{U}[0, 1)] \times T_{ACB}$ (Tello-Oquendo et al. 2018).

It is not difficult to realize that there is not a negligible waste of time when devices do not meet $g \leq P_{ACB}$, and therefore a lot of data transmission opportunities are missed. Let us check the amount of data that could be transmitted in the *back-off* period.

If a device does not meet the requirement to transmit a preamble, and it gets the minimum values from P_{ACB} and T_{ACB} (best case), the $T_{barring}$ = [0.7 + 0.6 × 0.05] × 4 = 2.92 s. This time is equivalent to (2.92 s/66.67 μs) = 43,798 symbols for 15 kHz numerology (numerology with longer symbol duration). Considering BPSK modulation (modulation with a minimum of bits per symbol), the total amount of data could be transmitted in the blackout period is 43,798 (No. symbols) × 1 (No. bits – BPSK) = 43,789 bits. Considering 64 Quadrature Amplitude Modulation (64QAM) modulation, the total amount of data transmitted is 43,798 (No. symbols) × 6 (No. bits – 64QAM) = 262,734 bits.

For the worst case of the example above, $T_{barring}$ = [0.7 + 0.6 × 0.95] × 512 = 650.24 s. This time is equivalent to (650.24 s/66.67 μs) = 9,753,113 symbols for 15 kHz numerology. Considering BPSK modulation, the total amount of data could be transmitted in the blackout period is 9,753,113 (No. symbols) × 1 (No. bits – BPSK) = 9,753,113 bits. Considering 64QAM modulation, the total amount of data could be transmitted is 9,753,113 (No. symbols) × 6 (No. bits – 64QAM) = 58,518,678 bits.

When devices meet $g \leq P_{ACB}$, their preamble transmission still could collide because all the devices that want to get resources from the network will acquire one of only 64 available preambles in NR. Therefore, ACB only alleviates congestion, it does not remove it completely. Thus, ACB will cause a negligible effect by reducing the number of devices contending for resources in mMTC.

2.3 PARTIAL CONCLUSIONS

This chapter introduced an overview of the entire procedure executed by a device when it enters a cell coverage area until it obtains resources from the mobile network. It was described the synchronization process of the device with the gNB in order to communicate in time and frequency domains in a reduced space. Therefore, the device saves energy by not searching for signals from the gNB in the entire network bandwidth. After synchronization, the device executes the *Random-Access* procedure or RACH procedure. The chapter described the entire RACH procedure, and then it focused on the RACH limitations, especially in massive communications where many devices try to obtain access at the same time, causing a lot of collisions due to they get the same preambles and transmit the preambles back to the gNB in the same channel at the same time; and a great access delay because of the employment of the ACB algorithm in the order of seconds to delay some devices' access and avoid collisions.

CHAPTER 3: A COMPARISON AMONG WI-FI DIRECT, CLASSIC BLUETOOTH, AND BLE FOR ENABLING MMTC

The main goal of this chapter is to assess Wi-Fi Direct, Classic Bluetooth, and BLE performances in terms of number of collisions, energy consumption, and discovery latency, in order to check out the more suitable technology for mMTC scenarios. It also describes the discovery procedures of the principal D2D technologies. Then, the Classic Bluetooth and Wi-Fi Direct discovery performances are compared by analyzing the behavior of the two technologies with a developed app. Afterwards, the chapter describes the details of an additional *back-off* implemented for Classic Bluetooth to decrease collisions and reduce elapsed times during devices' discovery in mMTC scenarios. Next, the performances of Classic Bluetooth and BLE are compared by simulating their discovery procedures.

3.1 DISCOVERY PROCEDURES OF THE PRINCIPAL D2D TECHNOLOGIES

Since the discovery procedure of the D2D technologies concerns a lot of signaling, it captures all the attention, especially in massive communications. Therefore, this section describes the algorithms employed by every D2D technology during the discovery procedure.

In the case of Wi-Fi Direct, every device has two states: searching state and listening state. Both states have a random duration of N time units (102.4 ms), where $N \in [1, 2, 3, \ldots]$. **Figure 10** shows the discovery procedure for Wi-Fi Direct. In the searching state, devices broadcast probe requests (discovery requests) in one of the three social channels: 1, 6, 11 in the 2.4 GHz band (Camps-Mur, Garcia-Saavedra, and Serrano 2013). In the same searching interval, the device listens to probe request replies. In the listening state, devices only listen to probe requests in one of the social channels and send back responses in the corresponding cases. The selected channel in the listening state remains constant during the entire discovery process. In this state, devices do not listen to responses of their own past probe transmissions (Sun et al. 2016). In conclusion, a device only discovers a remote device when it receives probe request responses in the searching state.

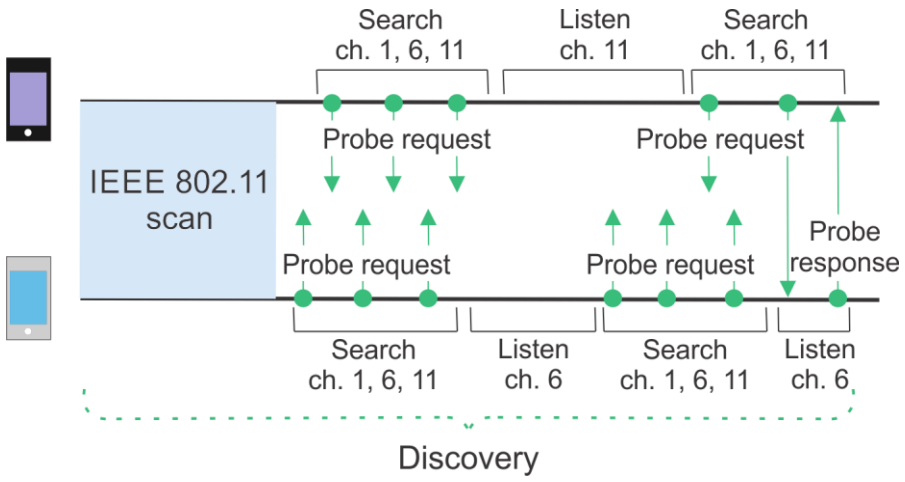


Figure 10. *Wi-Fi Direct discovery procedure between two devices. Redesigned from (Camps-Mur, Garcia-Saavedra, and Serrano 2013).*

For Bluetooth, the device that starts discovering nearby devices is called an inquiring device. It broadcasts inquiry packets in 32 of 79 possible frequencies. The 32 frequencies are previously agreed upon. The inquiring device keeps sending inquiry packets for two time-slots of 312.5 μs in two different frequencies generated by an internal 28-bit clock, as is shown in **Figure 11**. The device then listens in the next two subsequent time slots in the same frequencies it sent inquiry packets before. After the listening interval, if the inquiring device does not receive a reply to the inquiry packets, it starts the inquiry packets' broadcast again in two other frequencies. The scanning device uses the 28-bit clock to generate the frequency that the scanning device will use to listen to inquiry packets. The timing used by the scanning device is depicted in **Figure 12** (Duflot et al. 2006).

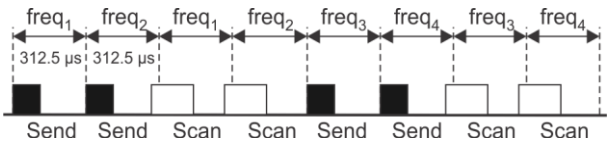


Figure 11. *Classic Bluetooth inquiring device's behavior. Redesigned from (Duflot et al. 2006).*

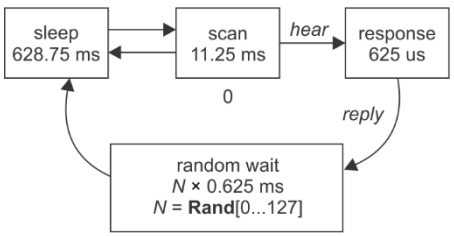


Figure 12. *Classic Bluetooth scanning device's behavior. Taken from (Duflot et al. 2006).*

In the case of BLE, there are only three channels used for the discovery procedure. These channels are the 37, 38, and 39. **Figure 13** (**a**, **b**) summarizes the BLE discovery procedure. On the one hand (**Figure 13a**), an advertising device sends advertising PDUs over the three channels during an Advertising Event. Between two consecutive advertising events, there is a variable T_a time, composed by a fixed *advInterval* and a pseudo-random *advDelay*. On the other hand (**Figure 13b**), a scanning device periodically scans the same three channels to look for advertising signals during a *scanWindow*, which is within a *scanInterval*. In every *scanWindow*, the scanning device scans a different channel from the three channels. The BLE standard states that the *advInterval* should be an integer multiple of 0.625 ms in the range of 20 ms to 10.24 s, the *advDelay* should be within the range of 0 ms to 10 ms, and the *scanInterval* and *scanWindow* shall be less than or equal to 10.24 s (Jia Liu, Canfeng Chen, and Yan Ma 2012) (Cho et al. 2016).

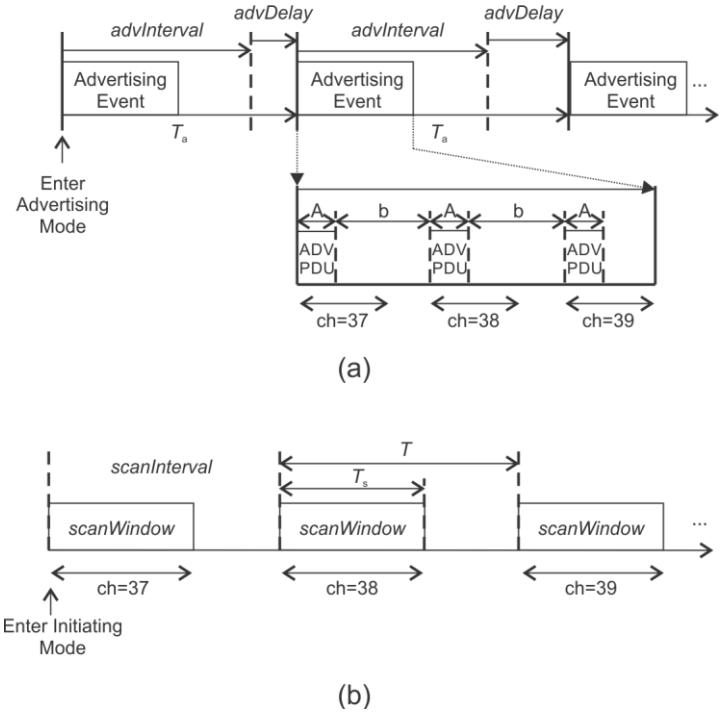


Figure 13. *BLE discovery procedure: (**a**) Advertising process and (**b**) Scanning process. Taken from (Jia Liu, Canfeng Chen, and Yan Ma 2012).*

3.2 CLASSIC BLUETOOTH AND WI-FI DIRECT DISCOVERY PERFORMANCES' COMPARISON IN TERMS OF DISCOVERY LATENCY

We developed two Android apps to test the Classic Bluetooth and Wi-Fi Direct elapsed times during the devices' discovery and incoming data processing. The

apps were tested on the same two devices: Samsung Galaxy S8+ cell phone and a Huawei P9 cell phone. After ten measurements, which is sufficient for the obvious difference in results, the average discovery time for Wi-Fi Direct was 8,164 ms. This result is almost five times greater than the Classic Bluetooth discovery time, as shown in **Table 3**. Therefore, Classic Bluetooth is best suitable to be employed as the D2D technology for enabling mMTC scenarios. However, the Classic Bluetooth discovery procedure can be applied to Wi-Fi to obtain a larger communication range. This means that the Classic Bluetooth discovery algorithm would be used but using the transmission power of Wi-Fi.

The two apps used for Classic Bluetooth and Wi-Fi Direct in this comparison were developed in Qt Creator and Android Studio using the Bluetooth and Wi-Fi Application Programming Interfaces (APIs), respectively. The apps can be found here https://github.com/Abel1027/D2D-Test-Apps.git.

Table 3. *Elapsed times in milliseconds for Classic Bluetooth and Wi-Fi Direct to find a remote device in short-range in 10 attempts.*

Attempts	Bluetooth (ms)	Wi-Fi Direct (ms)
1	649	8346
2	2659	8630
3	1600	6405
4	898	6335
5	2137	13176
6	4366	7329
7	530	6176
8	3196	9630
9	823	6000
10	373	9613
Mean	**1723.1**	**8164**
Standard deviation	**1269.16**	**2130.51**

3.3 PROPOSED MODIFICATIONS FOR THE CLASSIC BLUETOOTH DISCOVERY PROCEDURE

Since Classic Bluetooth has a shorter elapsed time when discovering nearby devices in comparison to Wi-Fi Direct, Bluetooth is studied for improvement to apply for the discovery procedure in mMTC scenarios. In Classic Bluetooth, if more than one device wants to discover other devices and perform the inquiring task and start inquiring with the same sequence of frequencies, their transmissions will collide in every attempt, and they never will find a remote device. Therefore, a *back-off* is proposed to be added every 11.25 ms (the time needed by a scanning device to listen to incoming discovery messages). The *back-off* is computed as a random value from

$[0, 1, \ldots, 10] \times 312.5\ \mu s$ where the maximum value is $10 \times 312.5\ \mu s = 3.125$ ms. The proposed *back-off* avoids transmissions at the same moment in the same frequency for a discrete simulated environment with a resolution of 31.25 µs and a communication latency of 0 ms. In real scenarios, the *back-off* needs to be calculated depending on the expected latency. In the next section, we use our own solution to simulate an environment where a set of devices (inquiring) tries to find another set of devices on discoverable mode (scanning).

3.4 PERFORMANCE COMPARISON BETWEEN CLASSIC BLUETOOTH AND BLE DURING DISCOVERY IN TERMS OF NUMBER OF COLLISIONS, POWER CONSUMPTION, AND DISCOVERY LATENCY

We employed *SimPy* (G. Müller, Vignaux, and Stefan Scherfke n.d.), a process-based discrete-event simulation Framework used through the *Python* programming language, to simulate the behavior of the Classic Bluetooth discovery algorithm with the modifications proposed in this work and the BLE discovery procedure. Why is *SimPy* selected for the simulation? Why is it not used as a continuous-time simulation tool? The answer is related to the computation capabilities of the computers where the simulation could be executed and the time resolution a programming language can offer. On one hand, the simulation involves hundreds of simultaneous processes or threads. This can reduce the overall performance of the simulation by delaying some processes more than others. In a real scenario, every device performs its functions and neither of their processes are affected by other devices' processes. On the other hand, the minimum time resolution of the programming languages is in the order of the milliseconds and this does not satisfy the proposed environment, where the minimum time resolution is in the order of the microseconds. To overcome those problems, *SimPy* was taken into account. *SimPy* waits that all simultaneous loops within different processes finish, and then it saves every output of those processes with the same timestamp. This means that it does not matter if one process is faster than others, *SimPy* always waits for the slowest process and assigns the same timestamp as the faster. The time resolution is solved too because *SimPy* only saves timestamps as a float value, and not as a real-time value. For example, if *SimPy* is set with a time resolution of 1, *SimPy* interprets it just as 1 and not as 1 µs, or 1 ms. The time resolution is interpreted by the application and not by *SimPy*, which is helpful for process synchronization.

The simulation was deployed using an MSI laptop with a Core i7 processor, a 16 GB RAM, and a 16 GB NVIDIA GeForce RTX 2070 video card. The scripts used for the simulation can be found here: https://github.com/Abel1027/Classic-Bluetooth-vs-BLE.git. The results of the simulation are shown below after computing the average of 10 simulations by variating the random seed in the range 0-9.

Figures **14-17** show the simulation when the number of inquiring devices augments from 10 to 100 in steps of 10 devices, and the number of scanning devices remains the same, in this case, 10 devices. Figures **18-21** show the simulation when the number of inquiring devices remains the same (10 devices), and the number of scanning devices augments from 10 to 100 in steps of 10 devices. The x-axis of the figures shows the number of inquiring/scanning devices. For example, 10/20 means that there are 10 inquiring devices and 20 scanning devices.

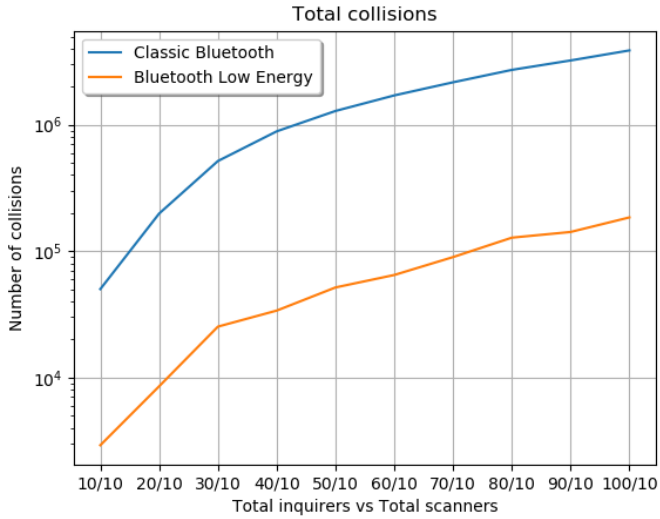


Figure 14. *Total number of collisions during the discovery procedure when the number of inquiring devices increases and the number of scanning devices remains the same.*

Figure 14 shows fewer collisions during the BLE discovery procedure than the Classic Bluetooth discovery procedure. Therefore, the energy consumption and the discovery time are expected to be lower in the case of BLE than in Classic Bluetooth.

The energy consumed by the devices is computed as:

$$E = P \times \frac{m}{R_b} \quad (1),$$

where E is the consumed energy in every transmission given in J, P is the transmission power in W, m is the number of bits of the transmitted message, and R_b is the bit rate given in bits/seconds.

The energy consumed by the inquiring devices is calculated from Eq.(1) with $P = 6.31\ mW$, $m = 68\ bits$, and $R_b = 1\ Mb/s$ for Classic Bluetooth; and with $P = 6.31\ mW$, $m = 128\ bits$ => 8 bits (Preamble) + 32 bits (Access Address) + 64 bits (Connectable Undirected Advertising packet) + 24 bits (*Cyclic Redundancy Check* - CRC), and $R_b = 1\ Mb/s$ for BLE. The energy consumed by the scanning devices is calculated from Eq.(1) with $P = 6.31\ mW$, $m = 286\ bits$ => 72 bits (Access Code) + 54 bits (Header) + 144 bits (Payload) + 16 bits (CRC), and $R_b = 1\ Mb/s$ for Classic Bluetooth; and with $P = 6.31\ mW$, $m = 176\ bits$ => 8 bits (Preamble) + 32 bits (Access Address) + 112 bits (Connectable Directed Advertising packet) + 24 bits (CRC), and $R_b = 1\ Mb/s$ for BLE.

From **Figure** 15, the energy consumed by the inquiring devices is less for the BLE case than for the Classic Bluetooth case. The same occurs for the energy consumption of the scanning devices (see **Figure 16**).

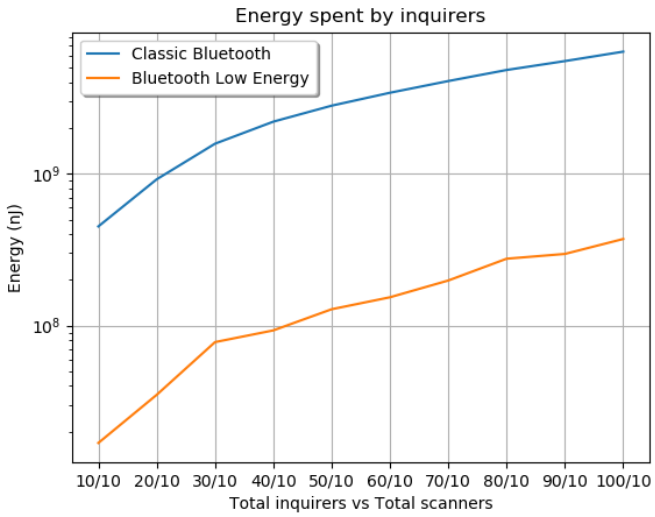


Figure 15. *Total energy spent by the inquiring devices during the discovery procedure when the number of inquiring devices increases and the number of scanning devices remains the same.*

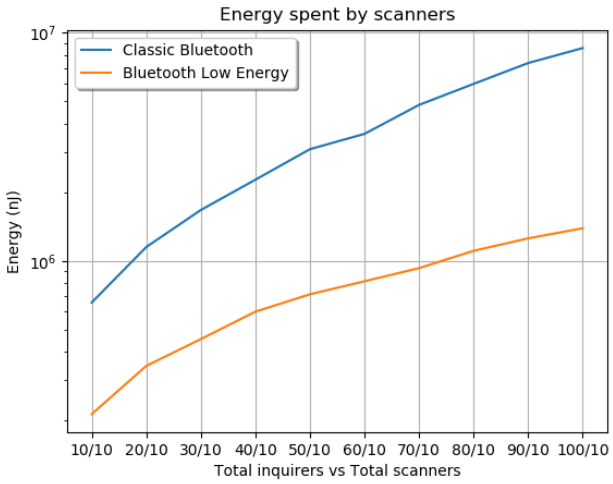


Figure 16. *Total energy spent by the scanning devices during the discovery procedure when the number of inquiring devices increases and the number of scanning devices remains the same.*

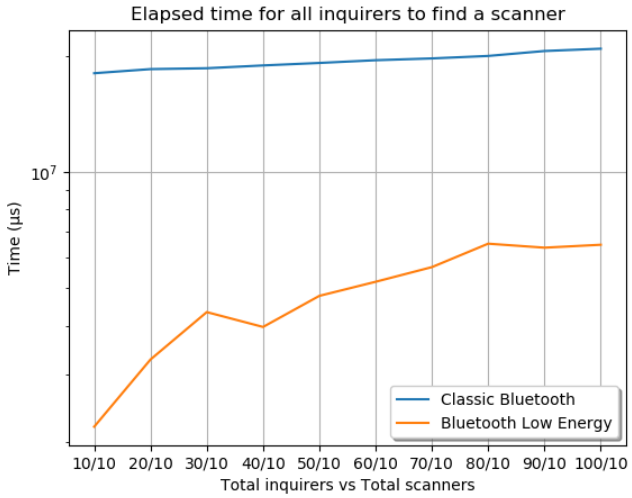


Figure 17. *Total elapsed time for all inquiring devices to find a scanning device during the discovery procedure when the number of inquiring devices increases and the number of scanning devices remains the same.*

Figure 17 shows that the total time elapsed during the discovery procedure is less for the BLE case than for the Classic Bluetooth case. In the BLE case, the discovery time is below 10 seconds while the discovery time for the Classic Bluetooth case is above 10 seconds.

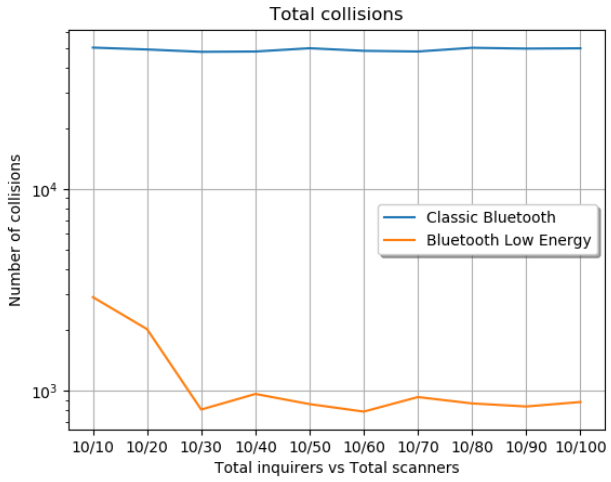


Figure 18. *Total number of collisions during the discovery procedure when the number of inquiring devices remains the same and the number of scanning devices increases.*

From **Figure 18**, there are fewer collisions during the BLE discovery procedure than the Classic Bluetooth discovery procedure. Therefore, the energy consumption and the discovery time are expected to be lower in the case of BLE than in Classic Bluetooth.

The energy consumed by the inquiring devices is less for the BLE case than for the Classic Bluetooth case, as it is depicted in **Figure 19**. The same occurs for the energy consumption of the scanning devices (see **Figure 20**).

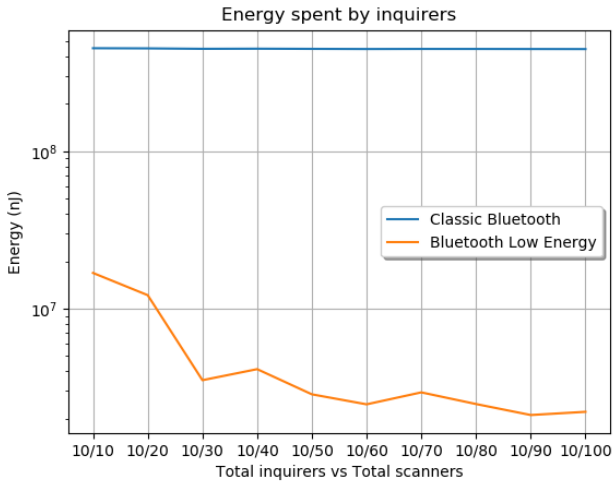


Figure 19. *Total energy units spent by the inquiring devices during the discovery procedure when the number of inquiring devices remains the same and the number of scanning devices increases.*

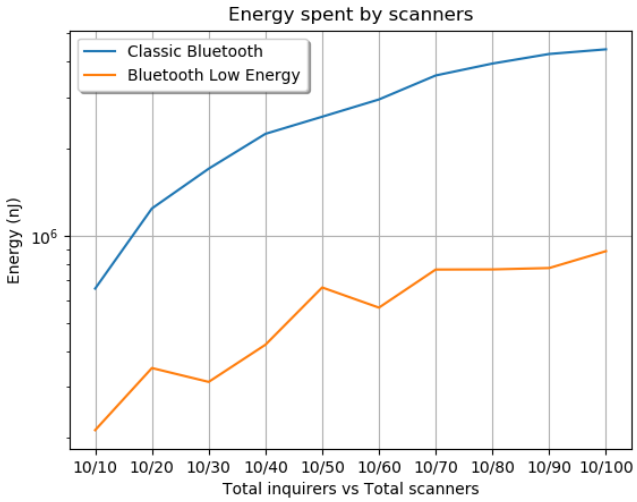


Figure 20. *Total energy units spent by the scanning devices during the discovery procedure when the number of inquiring devices remains the same and the number of scanning devices increases.*

Figure 21 shows that the total time elapsed during the discovery procedure is less for the BLE case than for the Classic Bluetooth case. In the BLE case, the discovery time is below 1 second when the number of scanning devices is greater than 20 while the discovery time for the Classic Bluetooth case is always above 10 seconds.

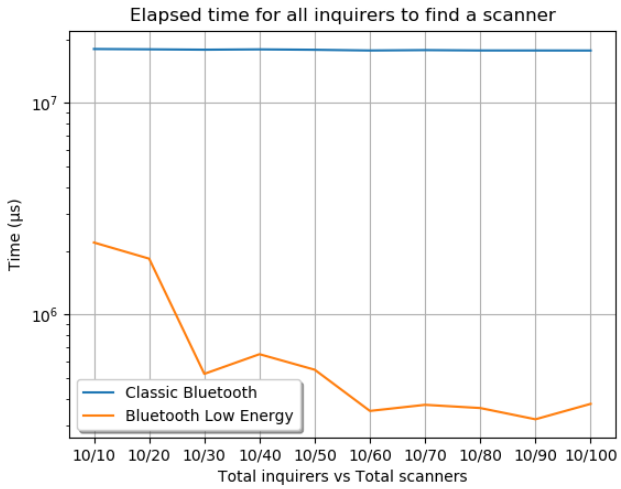


Figure 21. *Total elapsed time for all inquiring devices to find a scanning device during the discovery procedure when the number of inquiring devices remains the same and the number of scanning devices increases.*

3.5 PARTIAL CONCLUSIONS

In this chapter we compared the discovery procedures' performances of Wi-Fi Direct and Classic Bluetooth. Then, we compared Classic Bluetooth with BLE. The results show that Classic Bluetooth was 5 times faster than Wi-Fi Direct during the devices' discovery. However, Classic Bluetooth is slower than BLE. BLE also experiences fewer collisions and consumes less energy than Classic Bluetooth during the devices' discovery. BLE allows that around 100 inquiring devices find at least 1 scanning device out of 10 scanning devices in less than 10 seconds. BLE also allows that around 10 inquiring devices discover at least 1 scanning device out of 100 scanning devices in less than 1 second, in contrast with the 10 seconds the Classic Bluetooth takes to do the same task. Therefore, BLE is faster and consumes less energy than Bluetooth, and Wi-Fi by transitivity (BLE > Bluetooth > Wi-Fi in terms of energy consumption). For these reasons, BLE is one of the most promising D2D communication technologies—out of the three analyzed in this work—for enabling mMTC in the *Next-Generation* networks. However, BLE is a very short-range D2D technology, and it has some security vulnerabilities during the pairing procedure, which can be overcome by other D2D technologies like LTE-Direct.

CHAPTER 4: RAA FRAMEWORK

This chapter describes the proposed framework: RAA. It embraces the functions related to the devices looking for access, the devices with allocated resources, and the gNB. The chapter also includes an example where some devices try to gain access in the mobile network by using the RAA approach.

4.1 RAA DETAILS

The main purpose of RAA is that new devices (requesters) entering the cell coverage area discover at least one nearby peer with uplink grants that serve as a bridge between them and the gNB. If the new devices find another device that is registered in the cell that they want to have access to, the registered device can act as a relay for them to forward their resource requirements directly to the gNB. This procedure means that the new devices do not need to wait for the transmission of SIB1 by the gNB to start the traditional *Random-Access* procedure. The behavior of RAA is summarized in **Figure 22**.

Figure 22a shows the requester's behavior. When the requester enters a cell coverage area, it tries to synchronize with the gNB downlink. To do that, the requester waits to receive the SSB, which is a message broadcasted periodically by the gNB. SSB contains the PSS and the SSS signals. The requester extracts the *Sector ID* (SID) of the cell from PSS, and the *Group ID* (GID) of the cell from SSS (M.R., Dama, and Kuchi 2017). Then, the requester computes the *cell ID* with both SID and GID and starts performing the proposed RAA procedure.

The requester also extracts the MIB from SSB. MIB provides the bandwidth of the downlink, the frame numbers, and the SIB1 location in frequency and time-domain (Chandramouli, Liebhart, and Pirskanen 2019). Then, the device scans for SIB1. Once the requester receives SIB1, it extracts the configuration parameters from the message to perform the *Random-Access* procedure.

At this point, the requester executes two procedures in parallel, the traditional *Random-Access* procedure, and the proposed RAA procedure. This is possible because the requester transmits in different channels for every procedure. In the RAA procedure, the device starts looking for nearby devices that have already allocated resources—also called relays—in the same cell the requester is. For the discovery process, the requester broadcasts a discovery message via D2D communication, which contains the *cell ID* of the cell the requester has camped, the number of

subcarriers the requester needs for downlink (SC_{DL}), the number of subcarriers the requester needs for uplink (SC_{UL}), the number of symbols the requester needs for downlink ($NoSym_{DL}$), and the number of symbols the requester needs for uplink ($NoSym_{UL}$). The requester sends the discovery message periodically during an inquiry interval in the 2.4 GHz channel, which does not interfere with the communications in the mobile network band. The requester stops the discovery if it receives either an acknowledge message from a nearby relay or a RAR from the gNB. If neither of those messages arrives, the requester continues discovering relays until the inquiry interval ends. After that, the requester applies a random *back-off* to avoid collision with other requesters that could be transmitting discovery messages at the same time. Once the *back-off* expires, the requester starts discovering relays again.

In case the requester receives an acknowledge message from a relay, the requester waits for a RAR from the gNB. Once the requester receives the RAR due to a previous relay discovery or as part of the traditional *Random-Access* procedure, the requester extracts from the RAR the information with frequency and time-domain to transmit the *Radio Resource Control* (RRC) *Connection Request* message. Then, the requester waits for receiving the *RRC Connection Setup* message with the allocated resources. Now, the requester becomes a relay.

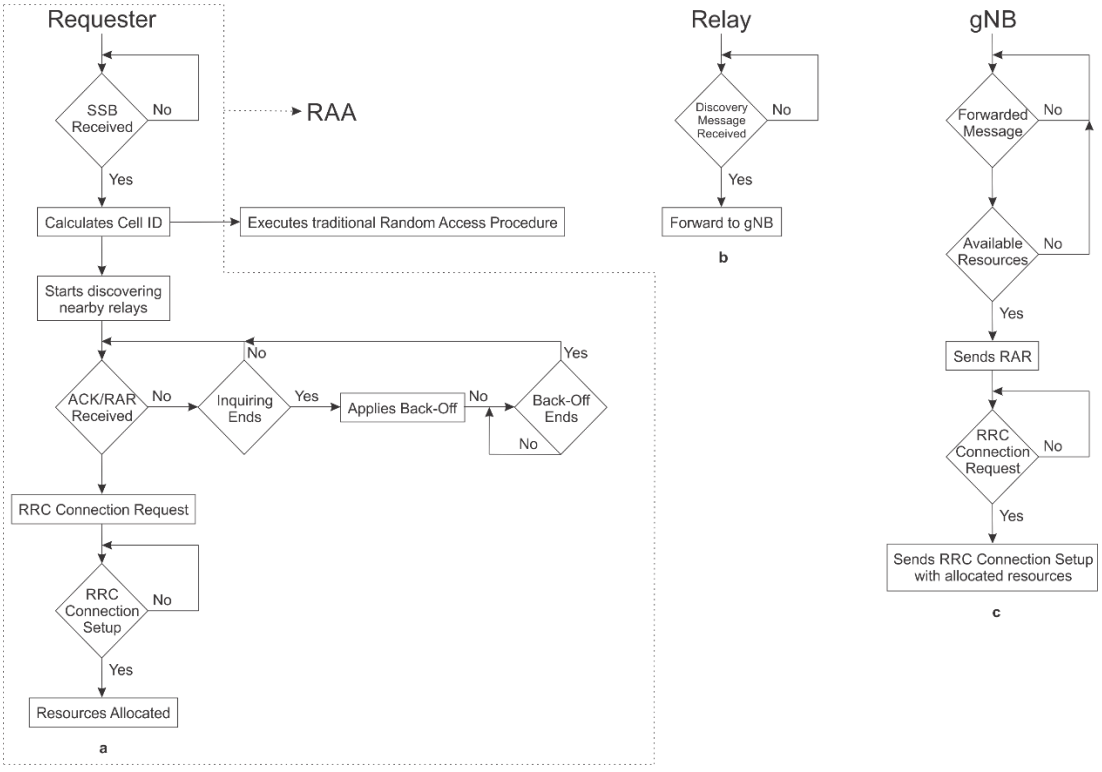


Figure 22. *(a) Requester, (b) Relay, and (c) gNB summarized behavior pseudocodes.*

In **Figure 22b** is depicted the relay behavior. A device could act as a relay after accessing the network or when commanded by the gNB. In this work, we assume that the devices become relays after accessing the network. From the figure, the relay

listens to discovery messages during a scanning interval. If the scanning interval ends, the relay waits a long time (seconds) to start scanning again. If during the scanning interval the relay receives a discovery message from a nearby requester, it will send an acknowledge message to the requester to notify that a relay has been found. The relay also forwards the requester message with the requester resource requirements to the gNB in the relay allocated uplink resources.

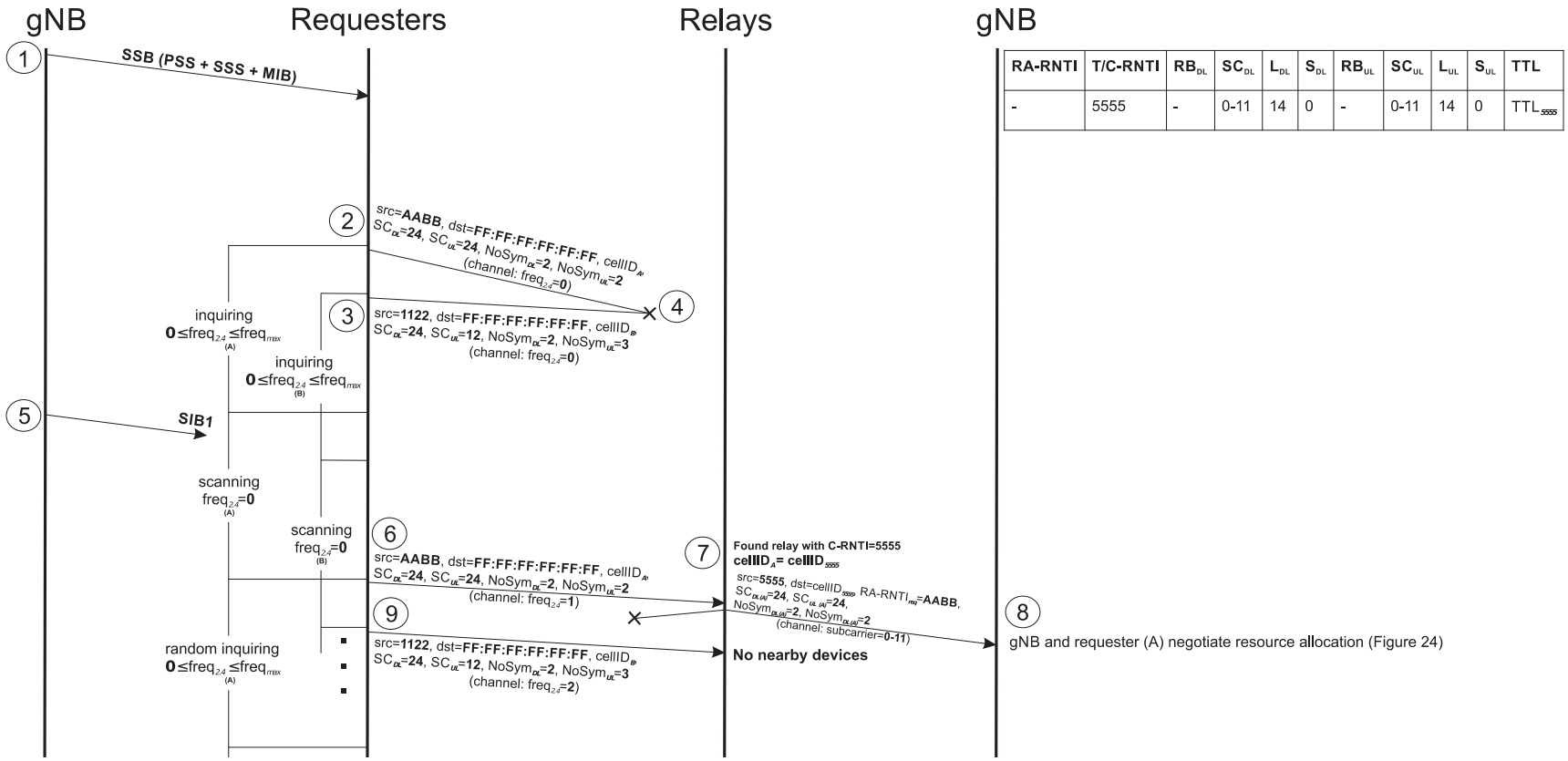
Figure 22c shows the gNB functionalities during the proposed RAA procedure. The gNB receives the information with the required resources for the requester via relay. Then, the gNB looks in a resource allocation table if there are available resources for the requester. If there are resources, the gNB informs the requester of resource availability via RAR. Otherwise, the gNB ignores the forwarded message.

4.2 MEDIUM ACCESS CONTROL

The D2D communication between the requesters and the relays is performed through wireless communication. In this kind of communication, the medium access control cannot remove collision when more than one device transmits at the same time in the same frequency. Instead, wireless technologies are focused on collision avoidance. Thus, collisions could exist but not permanently. RAA was conceived to be based on BLE as the wireless technologies for D2D communication due to BLE is the technology—out of the three analyzed in this work—with best performance during the discovery procedure about access latency and energy consumption, as it was proved in Chapter 3. In this case, requesters and relays select one of 3 available frequencies from the 2.4 GHz band to transmit the discovery messages and to listen to the discovery messages, respectively. If two or more nearby requesters select the same frequency and transmit their discovery messages at the same time, their transmissions will collide. Therefore, neither of the requesters will find a relay. However, the requesters select periodically new frequencies for the next transmissions within an inquiring interval. The only way that more than one requester selects the same frequency all the time is if they use the same frequency sequence. Fortunately, BLE adds a *back-off* after an inquiring/advertising interval. This increases the probability that the next time the requesters start the discovery, they use different frequencies for their transmissions.

4.3 RAA PROCEDURE EXAMPLE

This section shows some examples of message exchange between some requesters, relays, and the gNB during the RAA procedure. **Figure 23** shows such interactions, and below is explained every step.



RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}

Figure 23. *Message exchange between requester devices, relay devices, and the gNB.*

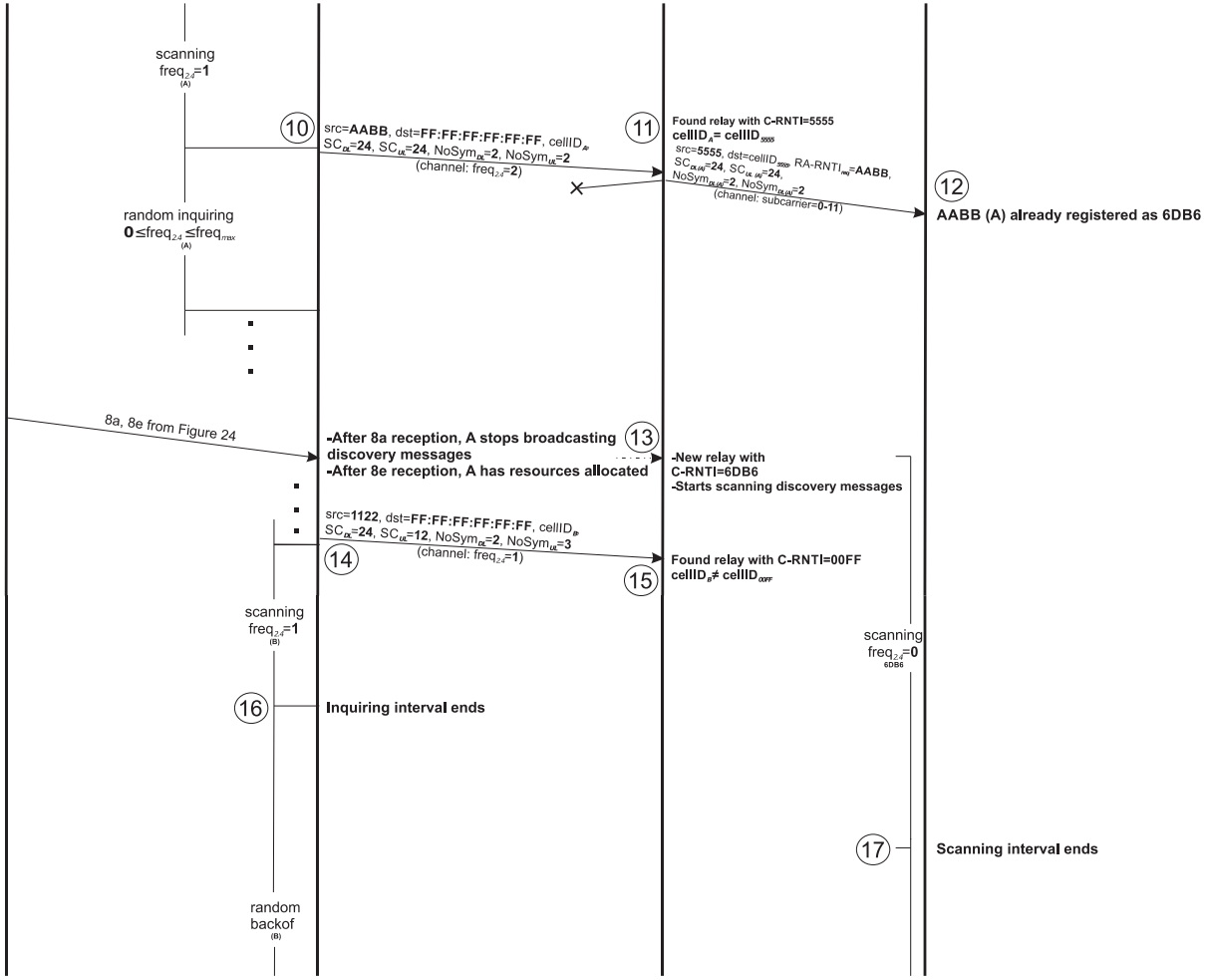
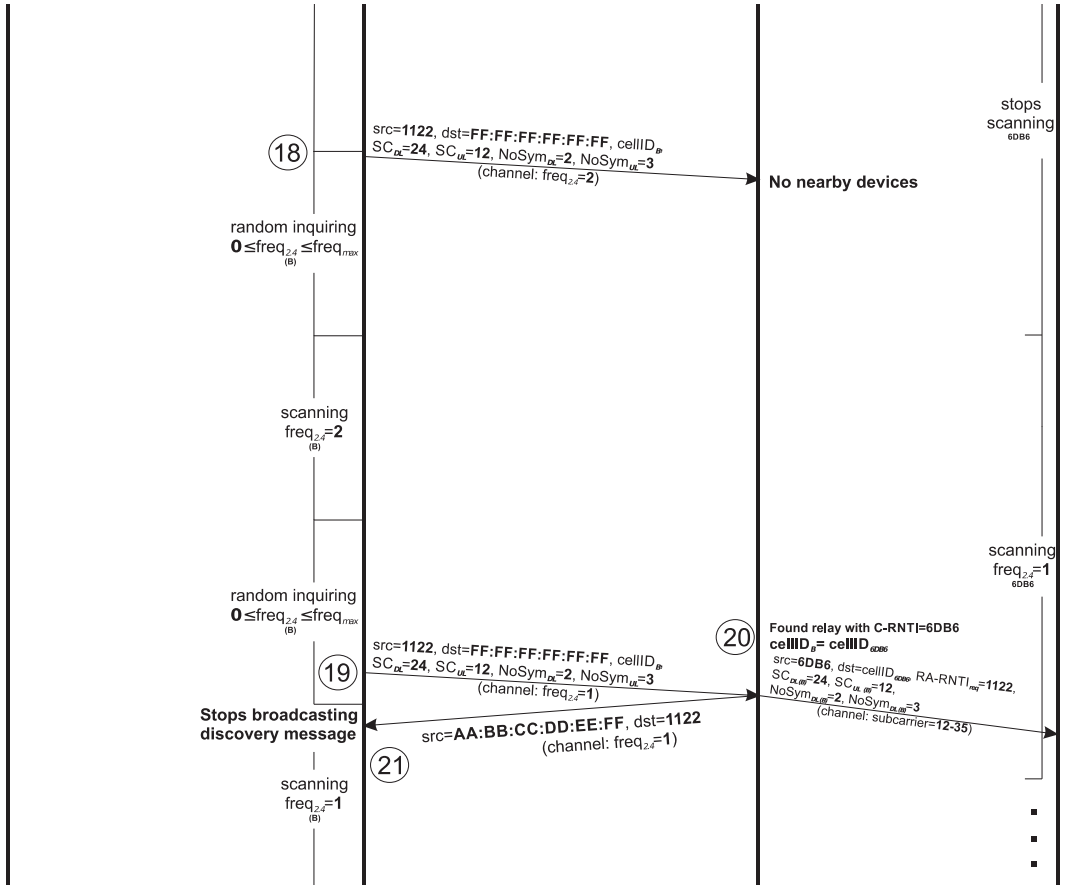
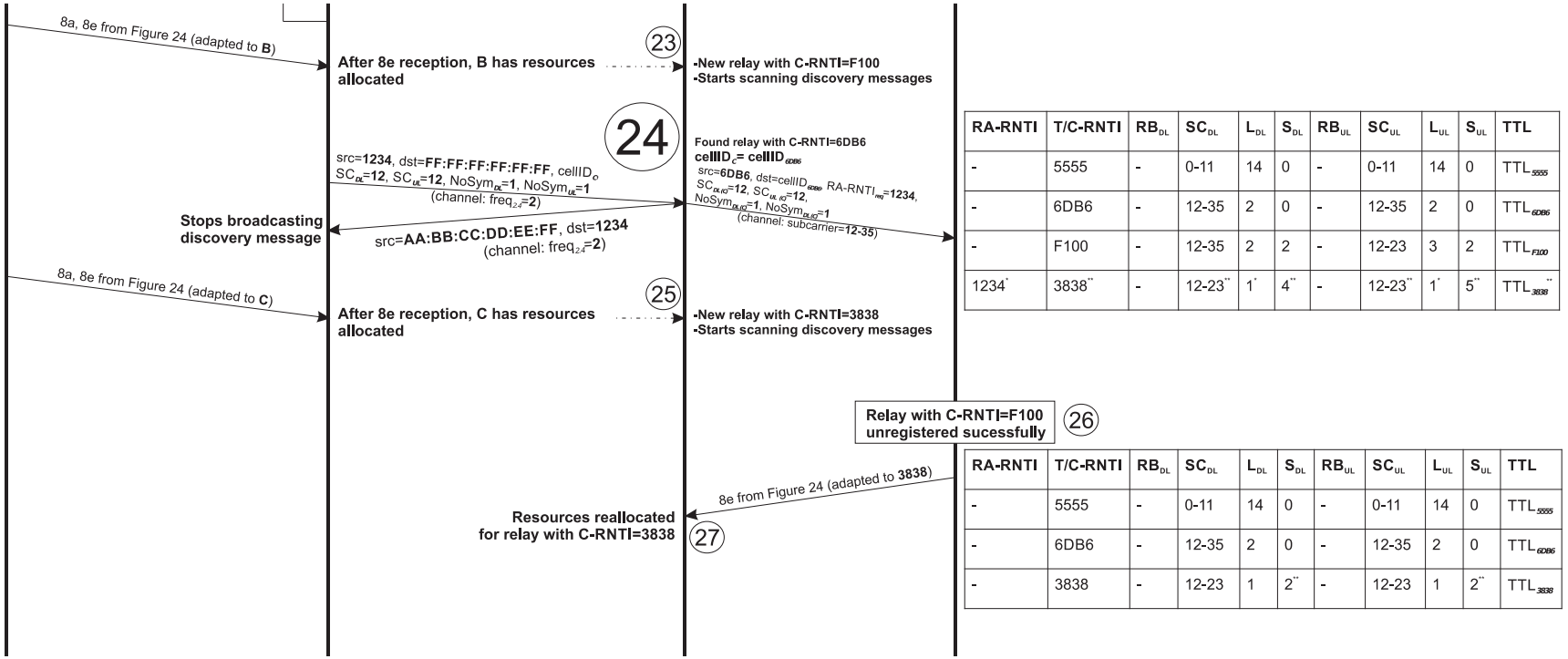


Figure 23. *Message exchange between requester devices, relay devices, and the gNB.*



RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}
AABB	6DB6	-	12-35	2	0	-	12-35	2	0	TTL_{6DB6}
1122*	F100**	-	12-35**	2*	2**	-	12-23**	3*	2**	TTL_{F100}**

Figure 23. *Message exchange between requester devices, relay devices, and the gNB.*



RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}
-	6DB6	-	12-35	2	0	-	12-35	2	0	TTL_{6DB6}
-	F100	-	12-35	2	2	-	12-23	3	2	TTL_{F100}
1234*	3838**	-	12-23**	1*	4**	-	12-23**	1*	5**	TTL_{3838}**

RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}
-	6DB6	-	12-35	2	0	-	12-35	2	0	TTL_{6DB6}
-	3838	-	12-23	1	2**	-	12-23	1	2**	TTL_{3838}

Figure 23. *Message exchange between requester devices, relay devices, and the gNB.*

Legend:

- RB_{DL}/RB_{UL}: Number of Resources Blocks for downlink/uplink
- K_0/K_2: Slot offset from the slot the *Downlink Control Information* (DCI) is received by the requester for downlink/uplink
- L_{DL}/L_{UL}: Number of symbols for downlink/uplink (gNB)
- S_{DL}/S_{UL}: Symbol index for the first symbol offered by the gNB for downlink/uplink
- TTL: Time-To-Live
- src: Source
- dst: Destiny
- $cellID_X$: *Cell ID* of the cell the device **X** belongs to
- $freq_{2.4}$: Frequency from the 2.4 GHz band
- $freq_{max}$: Number of frequencies from the 2.4 GHz band for D2D communication
- *: Assigned by the requester
- **: Assigned by the gNB

Step 1: Synchronization with the gNB and PRACH configuration acquisition

The device needs to synchronize in frequency and time with the cell it has camped. Synchronization and the *cell ID* are acquired via the SSB. After synchronization, the device needs the PRACH configuration to start the *Random-Access* procedure. This information is carried out by SIB1. The device does not know how to acquire SIB1. MIB provides the device with parameters required to decode SIB1. Therefore, the device receives at first the MIB.

Step 2: Device **A** executes the *Random-Access* procedure and looks for relays

Device **A** starts the traditional *Random-Access* procedure and the relays discovery at the same time. The device sends a broadcast message to find a relay that belongs to the same cell. This message includes a device identifier randomly selected in the range **0x0001 - 0xFFF3** to not overlap with other network binary sequences that are not oriented to device identification, as it can be seen in **Table 4** (3GPP 2017). In this case, the selected value is **0xAABB**. The message also includes the *cell ID* (N_{ID}^{Cell}), the number of subcarriers the device needs for downlink (SC_{DL}), the number of subcarriers the device needs for uplink (SC_{UL}), the number of symbols the device

needs for downlink ($NoSym_{DL}$), and the number of symbols the device needs for uplink ($NoSym_{UL}$). To make this example simpler, the number of subcarriers and symbols is included in the message like if the requester knows the numerology that will be offered by the gNB to it. However, at this point, the requester does not know the numerology it will use for downlink and uplink communications. Therefore, the requester sends a number of subcarriers and symbols like it will communicate with the greater numerology (see note at the end of this step).

Table 4. *RNTI values.*

Value (hexa-decimal)	RNTI
0000	No assigned
0001 – 0960	RA-RNTI, C-RNTI, Semi-Persistent Scheduling C-RNTI, TC-RNTI, eIMTA-RNTI, TPC-PUCCH-RNTI, TPC-PUSCH-RNTI, SL-RNTI, G-RNTI, SL-V-RNTI, UL Semi-Persistent Scheduling V-RNTI, SL Semi-Persistent Scheduling V-RNTI, and SRS-TPC-RNTI
0961 – FFF3	C-RNTI, Semi-Persistent Scheduling C-RNTI, TC-RNTI, eIMTA-RNTI, TPC-PUCCH-RNTI, TPC-PUSCH-RNTI, SL-RNTI, G-RNTI, SL-V-RNTI, UL Semi-Persistent Scheduling V-RNTI, SL Semi-Persistent Scheduling V-RNTI, and SRS-TPC-RNTI
FFF4	SI-RNTI
FFF5 – FFF9	Reserved for future use
FFFA	SC-N-RNTI
FFFB	SC-RNTI
FFFC	CC-RNTI
FFFD	M-RNTI
FFFE	P-RNTI
FFFF	SI-RNTI

The message (discovery message) is broadcasted in a 2.4 GHz channel during an inquiring slot. The 2.4 channel selected by the device follows the BLE procedure. The device selects the first frequency (or randomly) of the $freq_{max}$ possible frequencies for transmission. After the inquiring slot, the device scans for relays responses in the same frequency it sent the discovery message during a scanning interval. This process is cyclically repeated until the device uses all the frequencies for inquiring of the $freq_{max}$ possible frequencies. Then, the device executes a random *back-off* to avoid collisions with other requester transmissions. After *back-off*, the device starts the process again. The procedure is executed until at least one relay is discovered or the gNB allocates resources for the device. In this case, **A** selects a channel ($freq_{2.4}$ = 1st frequency (0) from $freq_{max}$ possible frequencies) for the discovery message transmission.

A uses a binary sequence equal to the random identifier selected above (**0xAABB**) to descramble the PDCCH that contains the RAR. This sequence is used as RA-RNTI. Then, the device starts a blind search for RAR in Type1-PDCCH.

Note: The number of subcarriers and symbols required by a requester is calculated by the device with a 240 kHz SCS (greater SCS) and BPSK modulation (smaller modulation) base. This is necessary because the requester does not know the numerology and neither the modulation used in the BWP that will be assigned to it. The resource allocation, numerology, and modulation are informed via RRC signaling. At this point, the requester does not receive RRC setup yet. When the gNB receives the information with the number of subcarriers and symbols required by a requester, the gNB converts them in the number of subcarriers and symbols supported by the numerology that will be assigned to the requester.

For example, if a requester tells the gNB that it needs 2 subcarriers (SCS = 240 kHz) and 16 symbols = 16 bits (BPSK), the gNB assumes the requester needs a transmission/reception of 16 bits in a period of time:

$$\frac{NoSym\ (240\ kHz)_{BPSK}}{NoSc\ (240\ khz)} \times t_{sym\ (240\ kHz)} = \frac{16}{2} * 4.17\ \mu s = 33.36\ \mu s \quad (2)$$

where $NoSym\ (240\ kHz)$ is the number of symbols the requester needs for SCS = 240 kHz, $NoSc\ (240\ khz)$ is the number of subcarriers the requester needs to achieve 33.36 µs during transmission/reception, with a symbol duration of $t_{sym\ (240\ kHz)}$. Then, the gNB will allocate resources (number of subcarriers in this case) for the requester by the following formula:

$$NoSc\ (numerology) = \frac{2^{numerology(240\ kHz) - numerology} \times NoSc\ (240\ kHz)}{\log_2 modulation} \quad (3)$$

where $numerology$ is the numerology index {numerology = 0 (SCS = 15 kHz), numerology = 1 (SCS = 30 kHz), …, numerology = 4 (SCS = 240 kHz)}. Evaluating the requester parameters in the formula above and considering that the numerology that will be offered by the gNB to the requester is SCS = 30 kHz and the modulation is QPSK, the number of subcarriers allocated for the requester is:

$$NoSc\ (1) = \frac{2^{4-1} \times 2}{\log_2 4} = 8$$

Let's check the above result by evaluating in Eq.(2), the number of subcarriers offered by the gNB to the requester for 30 kHz numerology $NoSc\ (30\ kHz) = 8$, the number of symbols supported by the modulation $NoSym\ (30\ kHz)_{QPSK} = \frac{NoSym\ (240\ kHz)_{BPSK}}{\log_2 4} = \frac{16}{2} = 8$, and the symbol duration for 30 kHz numerology $t_{sym\ (30\ kHz)} = 33.36\ \mu s$:

$$\frac{NoSym\ (30\ kHz)_{QPSK}}{NoSc\ (30\ khz)} \times t_{sym\ (30\ kHz)} = \frac{8}{8} * 33.3\ \mu s = 33.33\ \mu s$$

As can be seen, the time for transmission/reception for this configuration (SCS = 30 kHz) is the same for the requester configuration (SCS = 240 kHz). In case the gNB has not enough resources to offer 8 subcarriers (part of one RB) to the requester for 30 kHz numerology, the gNB will assign the resources to the requester as the gNB could do it.

Step 3: Device **B** executes the *Random-Access* procedure and looks for relays

B starts the traditional *Random-Access* procedure and the relays discovery at the same time. The device sends a broadcast message to find a relay that belongs to the same cell. This message includes a random device identifier equal to **0x1122**, the *cell ID* (N_{ID}^{Cell}) of the cell where the device belongs to, the number of subcarriers the device needs for downlink (SC_{DL}), the number of subcarriers the device needs for uplink (SC_{UL}), the number of symbols the device needs for downlink ($NoSym_{DL}$), and the number of symbols the device needs for uplink ($NoSym_{UL}$).

The message is broadcasted in a 2.4 GHz channel during an inquiring slot. **B** selects a channel ($freq_{2.4}$ = 1st frequency (0) from $freq_{max}$ possible frequencies) for the discovery message transmission.

0x1122 is the binary sequence used to descramble the PDCCH that contains the RAR. This sequence is used as RA-RNTI. Then, the device starts a blind search for RAR in Type1-PDCCH.

Step 4: Collision between **A** and **B** discovery messages

Both **A** and **B** selected the same channel to transmit their discovery messages. Therefore, those messages collide and neither of the two devices can find a relay.

Step 5: Search space for MSG2 and MSG4 via SIB1

Once SIB1 is found for the devices, they need to know the search space for RAR in case the devices sent a *Random-Access Request* as part of the traditional *Random-Access* procedure for resource allocation. The RAR is part of the PDSCH. The SIB1 message defines the search space for the Type1-PDCCH that carries the RAR information (MSG2) and MSG4.

Step 6: **A** continues looking for relays after scanning responses

After the **A** scanning interval, the device performances the inquiring procedure again. **A** selects a new 2.4 frequency ($freq_{2.4}$ = 2nd frequency (1) from $freq_{max}$ possible frequencies) to broadcast a discovery message. The device keeps doing a blind search for RAR in Type1-PDCCH.

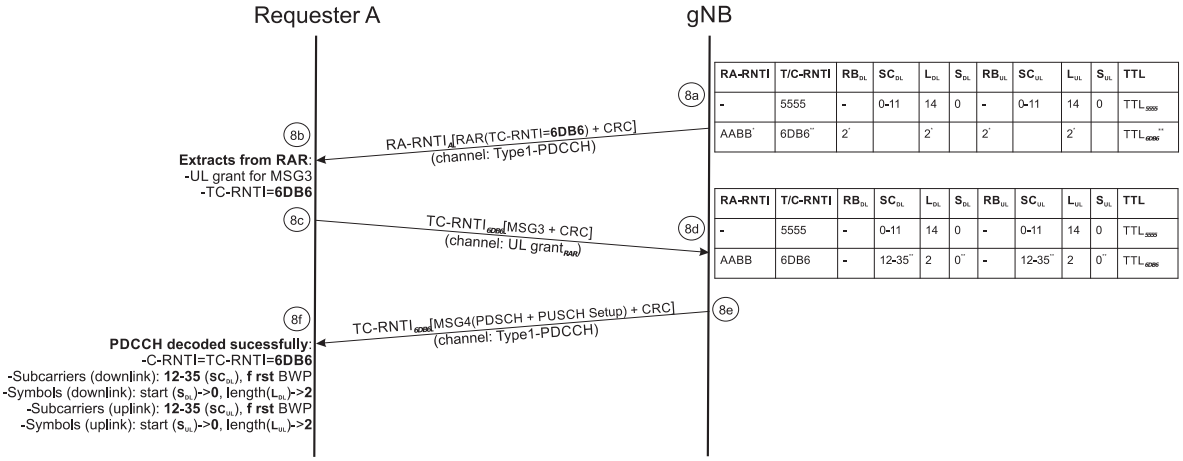
Step 7: **0x5555** relay is found

A relay with C-RNTI **0x5555** is scanning incoming discovery messages. The relay scans a discovery message from **A**. It checks that the *cell ID* transmitted in the message matches with its own *cell ID*. In this case, the *cell IDs* are equal to N_{ID}^{Cell} and the device acts like a relay between **A** and the gNB.

The relay sends the information about the number of symbols required by **A** for both downlink and uplink communications to the gNB. It also sends the RA-RNTI of **A** to the gNB. The message is sent in the uplink channel already assigned by the gNB to the relay in the past.

The relay also sends an acknowledge message to the requester (**A**) to notify a relay has been found. In this case, the acknowledge message collides with other transmissions in the 2.4 GHz band.

Step 8: gNB and requester **A** negotiate resource allocation (**Figure 24**)



RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}
AABB`	6DB6``	2`		2`		2`		2`		TTL_{6DB6}``

RA-RNTI	T/C-RNTI	RB_{DL}	SC_{DL}	L_{DL}	S_{DL}	RB_{UL}	SC_{UL}	L_{UL}	S_{UL}	TTL
-	5555	-	0-11	14	0	-	0-11	14	0	TTL_{5555}
AABB	6DB6	-	12-35``	2	0``	-	12-35``	2	0``	TTL_{6DB6}

Figure 24. *Requester registration with the gNB.*

8a. The gNB receives the forwarded message via **0x5555** relay in the relay uplink channel (subcarriers = 0-11). The gNB stores the requester RA-RNTI (**0xAABB**), both the number of subcarriers for downlink (2 RBs) and uplink (2 RBs), and both the number of symbols for downlink and uplink required by the requester in a resource allocation table. The gNB assigns a TC-RNTI to the requester: **0x6DB6**.

If the requester RA-RNTI matches an RA-RNTI selected by another device during the traditional *Random-Access* procedure, the gNB sends a RAR with the bit R (reserved bit set to 0) set to 1, indicating that the current RAR is for a requester asking for resources via D2D communication. This is used to avoid misinterpretation in MSG3 during the *Random-Access* procedure.

The gNB sends the RAR and appends a 16 bits CRC to the RAR. The RAR and the CRC are scrambled together with the requester RA-RNTI (**0xAABB**).

8b. During blind search, the requester (**A**) tries to descramble the Type1-PDCCH. It descrambles the RAR and the 16 bits CRC from PDCCH with its own RA-RNTI (**0xAABB**). Due to no CRC error is detected in the descrambled binary data, the requester determines that PDCCH carries its own control information, in this case, its own RAR.

The requester checks the uplink grants offered by the gNB for MSG3 transmission from the RAR message. It also extracts the TC-RNTI. This RNTI is used to descramble MSG4.

8c. The requester sends an *RRC Connection Request* to the gNB in MSG3 for resource allocation. This message is transmitted in the UL grant offered by the gNB in RAR and is scrambled with the requester TC-RNTI.

8d. The gNB receives the *RRC Connection Request* in the UL grant sent to the requester in RAR (*UL grant*$_{RAR}$). The gNB descrambles the *RRC Connection Request* with the TC-RNTI associated with the channel it receives the RRC message. Due to no CRC error is detected in the descrambled binary data, the gNB determines that requester (**0x6DB6** -> **A**) is requesting for *RRC Connection Setup*.

Downlink setup for frequency-domain resource assignment:

In NR, *Downlink Control Information* (DCI) formats 1_0, 1_1, and 1_2 are used to dynamically allocate frequency domain resources for PDSCH. NR uses two types of resource allocation in the downlink scheme; type 0 and type 1. When RRC is used to set up a device, type 0 is always used. The downlink resource allocation type 0 is a bitmap-based resource allocation scheme. The RB assignment information includes a bitmap indicating the *Resource Block Groups* (RBGs) that are allocated to the scheduled device

The gNB knew the number of subcarriers the requester needs by receiving the relay message in **8a**. Then, the gNB makes a match between the RBG and the BWP (see **Table 5**) (3GPP 2018d). From **Table 5** and with RBG size = 2 (two RBs = 24 subcarriers required by requester **A**), the gNB assigns the BWP 1-36 to the requester (**A**).

Table 5. *Relationship between the RGB size configuration and the Bandwidth Part (BWP) size.*

Bandwidth Part size	**RGB size**	
	Configuration 1	**Configuration 2**
1 – 36	2	4
37 – 72	4	8
73 – 144	8	16
145 – 275	16	16

The gNB has already assigned all symbols (14) of the first RB (subcarriers: 0-11th) in BWP 1-36 to relay **0x5555** for downlink. Therefore, the gNB assigns the next

two RBs (subcarriers: 12th-35th) of the BWP 1-36 to requester **A** with TC-RNTI **0x6DB6** for downlink communication.

Downlink setup for time-domain resource assignment:

In NR, DCI formats 1_0 and 1_1 are used to dynamically allocate time-domain resources for PDSCH. DCI formats 1_0 and 1_1 carry a 4-bit field named *time-domain resource assignment* which points to one of the rows of a look-up table. Each row in the look-up table provides the parameters shown in **Table 6**.

Table 6. *Parameters of the look-up table for time-domain resources' assigment.*

Parameter	Explanation
K_0	Slot offset from the slot where DCI is received
S	First symbol in the slot in which PDSCH will be received
L	Allocation length in number of OFDM symbols

In this case, the number of symbols required by the requester (**A**) for the downlink is 2. Then, the time-domain resource assignment is $K_0 = 0$, $S = 0$, $L = 2$.

Uplink setup for frequency-domain resource assignment:

The same procedure as in the downlink frequency-domain resource assignment is followed for uplink. In this case, the gNB has already assigned all symbols (14) of the first RB (subcarriers: 0-11th) in BWP 1-36 to relay **0x5555** for uplink. Therefore, the gNB assigns the next two RBs (subcarriers: 12th-35th) of the BWP 1-36 to requester **A** with TC-RNTI **0x6DB6** for uplink communication. Note that the same subcarriers are assigned for downlink and uplink communications, this is done just for a simpler explanation. In a real scenario, the downlink and uplink subcarriers given to the device should not be the same to allow full-duplex communication.

Uplink setup for time-domain resource assignment:

The same procedure as in the downlink time-domain resource assignment is followed for uplink. In this case, the number of symbols required by the requester (**A**) for uplink is 2. Then, the time domain resource assignment is $K_2 = 0$, $S = 0$, $L = 2$.

8e. All this information is carried out by the fields *PDSCH-Config* and *PUSCH-Config* within MSG4 for downlink and uplink configuration, respectively. MSG4 is sent in Type1-PDCCH by the gNB and is scrambled with the requester TC-RNTI.

8f. The requester (**A**) monitors the Type1-PDCCH search space to find the resource allocation information as part of the MSG4. The requester descrambles all possible PDCCHs in the search space with its own TC-RNTI. Due to no CRC error is detected in the descrambled binary data, the requester determines that PDCCH carries its own control information, in this case, its own *RRC Connection Setup*. If

PDCCH is successfully decoded, the requester sets to itself a C-RNTI equal to the TC-RNTI given to it by the gNB previously.

The requester obtains the start RB and the number of RBs from the field *PDSCH-Config* within RRC message. Now, the requester knows where are located its frequency resources for downlink communication (subcarriers: 12th-35th, in the first BWP).

From *PDSCH-Config*, the requester also obtains the *pdsch-TimeDomainAllocationList*. This is the look-up table to find the time domain resource assignment. In this case, the requester gets from the look-up table the slot offset, the start symbol, and the number of symbols. Now, the requester knows where are located its time-domain resources for downlink communication ($K_0 = 0$, $S = 0$, $L = 2$).

The requester finds frequency and time domain resource assignments for uplink communication the same way it found the resource assignments for downlink communication. This time, the requester gets that information from the field *PUSCH-Config* within RRC message. Now, the requester knows where are located its frequency-domain resources for uplink communication (subcarriers: 12th-35th, in the first BWP) and its time-domain resources for uplink communication ($K_2 = 0$, $S = 0$, $L = 2$).

At this point, requester **A** is able to send and receive its data directly to the gNB. The requester becomes a relay for the next requesters.

Step 9: **B** continues looking for relays

The **B** inquiring slot starts again. Therefore, the device can continue transmitting discovery messages. It selects channel $freq_{2.4}$ = 3rd frequency (2) from $freq_{max}$ possible frequencies to transmit the discovery message. In this case, the message does not reach any relays. Then, the device needs to keep looking for relays in the next inquiring slots. The device keeps doing a blind search for RAR in Type1-PDCCH.

Step 10: **A** continues looking for relays after scanning responses

After **A** scanning slot, the device performances the inquiring procedure again because it did not receive any acknowledge message from a relay and neither a RAR from the gNB. **A** selects channel $freq_{2.4}$ = 3rd frequency (2) from $freq_{max}$ possible frequencies to transmit the discovery message. The device keeps doing a blind search for RAR in Type1-PDCCH.

Step 11: **0x5555** relay is found

A relay with C-RNTI **0x5555** is scanning incoming discovery messages. The relay scans a discovery message from **A**. It checks that the *cell ID* transmitted in the message matches its own *cell ID*. In this case, the *cell IDs* are equal to N_{ID}^{Cell} and the device acts like a relay between **A** and the gNB.

The relay sends the information about the number of symbols required by **A** for both downlink and uplink communications to the gNB. It also sends the RA-RNTI

for **A** to the gNB. The message is sent in the uplink channel already assigned by the gNB to the relay in the past.

The relay also sends an acknowledge message to the requester (**A**) to notify a relay has been found. In this case, the acknowledge message collides with other transmissions in the 2.4 GHz band.

Step 12: **A** is already registered in gNB

The gNB receives the forwarded message via **0x5555** relay in the relay uplink channel (subcarriers = 0-11). The gNB extracts the requester RA-RNTI (**0xAABB**) and checks its resource allocation table. The gNB discovers that the requester (**0xAABB** -> **A**) is already registered in the current cell. If the TTL field in the resource allocation table, associated with this requester (TTL_{AABB}), has timed up the gNB starts the same procedure executed in step **8** (see **Figure 24**). The reception of the same message by the gNB means that the requester did not receive RAR from the gNB or the requester sends another unnecessary message probably because it did not receive an acknowledge message from its relay. In this case, the TTL has no timed up and the gNB does nothing.

Step 13: RAR (8a) and MSG4 (8e) reception by **A**

The requester **A** receives the RAR message from the gNB and stops broadcasting discovery messages to avoid collisions with other requester transmissions. Then, the requester receives MSG4 with allocated resources for it.

After resource allocation, the requester becomes a new relay in its cell. Now, it is able to forward discovery message information from other requesters to the gNB.

The new relay **A** starts scanning devices as a relay. It uses $freq_{2.4}$ = 1st frequency (0) from $freq_{max}$ possible frequencies to listen to incoming discovery messages. The selection of this specific frequency could be informed by the gNB to the relay.

Step 14: **B** continues looking for relays

The **B** inquiring slot starts again. Therefore, the device can continue transmitting discovery messages. It selects channel $freq_{2.4}$ = 2nd frequency (1) from $freq_{max}$ possible frequencies to transmit the discovery message. The device keeps doing a blind search for RAR in Type1-PDCCH.

Step 15: **0x00FF** relay is found

A relay with C-RNTI **0x00FF** is scanning incoming discovery messages. The relay scans a discovery message from **B**. It checks that the *cell ID* transmitted in the message matches its own *cell ID*. In this case, the *cell IDs* are different and the relay does nothing.

Step 16: No relay found by **B**

After the inquiring interval, the requester **B** stops transmitting discovery messages. The requester computes a random *back-off* to keep inactive for a while. After the *back-off* expires, the requester can start the discovery procedure again. This is helpful to avoid battery drain and collisions.

Step 17: Relay **0x6DB6** (**A**) stops scanning discovery messages

After a scanning interval (it also could be defined by the gNB depending on the rate of registration requests), the relay **0x6DB6** (**A**) stops scanning discovery messages to avoid battery drain. The relay does not scan incoming discovery messages during a stop interval that could be also defined by the gNB.

Step 18: **B** continues looking for relays

The **B** inquiring slot starts again. Therefore, the device can continue transmitting discovery messages. It selects channel $freq_{2.4}$ = 3rd frequency (2) from $freq_{max}$ possible frequencies to transmit the discovery message. The device keeps doing a blind search for RAR in Type1-PDCCH.

This requester is in the coverage area of relay **0x6DB6** (**A**). However, the relay has stopped scanning discovery messages. Therefore, the relay does not listen to the discovery message sent by **B**. The same occurs if **0x6DB6** is actively listening to discovery messages in a different frequency from the frequency the requester is transmitting.

Step 19: **B** continues looking for relays

Requester **B** continues transmitting discovery messages, now in a new inquiring slot. It selects channel $freq_{2.4}$ = 2nd frequency (1) from $freq_{max}$ possible frequencies. The device keeps doing a blind search for RAR in Type1-PDCCH.

This requester is in the coverage area of relay **0x6DB6** (**A**). The relay is scanning in channel $freq_{2.4}$ = 1 like the requester. Therefore, the relay listens to the discovery message sent by **B**.

Step 20: **0x6DB6** (**A**) relay is found

The relay with C-RNTI **0x6DB6** is scanning incoming discovery messages. The relay scans a discovery message from **B**. It checks that the *cell ID* transmitted in the message matches its own *cell ID*. In this case, the *cell IDs* are equal to N_{ID}^{Cell} and the device acts like a relay between **B** and the gNB.

The relay sends the information about the number of symbols required by **B** for both downlink and uplink communications to the gNB. It also sends the RA-RNTI for **B** to the gNB. The message is sent in the uplink channel already assigned by the gNB to the relay in the past.

The relay also sends an acknowledge message to the requester (**B**) to notify a relay has been found. The acknowledge message is sent back in the same channel from the 2.4 GHz band: $freq_{2.4} = 1$.

Step 21: Relay **0x6DB6** (**A**) sends an acknowledge message to requester **B**

The acknowledge message is simple. It only contains the requester RA-RNTI: **0x1122** as the destiny. The requester scans this message in its scanning interval due to the message is sent in the same frequency the requester is listening to. Then, the requester stops broadcasting discovery messages and stops performing the traditional *Random-Access* procedure too.

Step 22: The gNB allocates resources for requester **B**

The gNB receives the forwarded message via relay **0x6DB6** (**A**). Then, the gNB and the requester negotiate resource allocation as in **8**. In this case, the requester needs 2 RBs = 24 subcarriers and 2 symbols for downlink and 1 RB = 12 subcarriers and 3 symbols for uplink communications. The gNB checks its resource allocation table and realizes that it has enough resources for the requester in subcarriers 12th-35th, in the first BWP. The gNB assigns subcarriers 12th-35th, in the first BWP, and the next two symbols after device **0x6DB6** (**A**) symbols: 3rd-4th ($K_0 = 0$, $S = 2$, $L = 2$), for downlink. The gNB also assigns subcarriers 12th-23th, in the first BWP, and the next three symbols after device **0x6DB6** (**A**) symbols: 3rd-5th ($K_2 = 0$, $S = 2$, $L = 3$), for uplink.

Step 23: MSG4 (**8e**) reception by **B**

The requester **B** receives MSG4 with allocated resources from the gNB. Then, the requester becomes a new relay in its cell. Now, it is able to forward discovery message information from other requesters to the gNB. It starts scanning devices as a relay.

Step 24: A new requester (**C**) executes the *Random-Access* procedure and looks for relays

C starts the *Random-Access* procedure and the relays' discovery at the same time. The device sends a broadcast message to find a relay that belongs to the same cell. This message includes a random device identifier equal to **0x1234**, the *cell ID* (N_{ID}^{Cell}), the number of subcarriers the device needs for downlink (SC_{DL}), the number of subcarriers the device needs for uplink (SC_{UL}), the number of symbols the device needs for downlink ($NoSym_{DL}$), and the number of symbols the device needs for uplink ($NoSym_{UL}$).

The message is broadcasted in a 2.4 GHz channel during an inquiring slot. **C** selects a channel ($freq_{2.4}$ = 3rd frequency (2) from $freq_{max}$) for the discovery message transmission.

0x1234 is the binary sequence used to descramble PDCCH with RAR. This sequence is used as RA-RNTI. Then, the device starts blind search for RAR in Type1-PDCCH.

Relay **0x6DB6** (**A**) scans an incoming discovery message from **C** and forwards it to the gNB with the information about the number of resources blocks and symbols required by **C** for both downlink and uplink communications. The relay also sends an acknowledge message to the requester to notify a relay has been found.

The gNB receives the forwarded message. Then, the gNB and the requester negotiate resource allocation as in step **8**. In this case, the requester needs 1 RB = 12 subcarriers and 1 symbol for downlink and 1 RB = 12 subcarriers and 1 symbol for uplink communications. The gNB checks its resource allocation table and realizes that it has enough resources for the requester in subcarriers 12th-23th, in the first BWP. The gNB assigns subcarriers 12th-23th, in the first BWP, and the next symbol after device **0xF100** (**B**) symbols: 5th ($K_0 = 0$, $S = 4$, $L = 1$), for downlink. The gNB also assigns subcarriers 12th-23th, in the first BWP, and the next symbol after device **0xF100** (**B**) symbols: 6th ($K_2 = 0$, $S = 5$, $L = 1$), for uplink.

Step 25: MSG4 (**8e**) reception by **C**

The requester **C** (registered in the network as **0x3838**) receives MSG4 with allocated resources from the gNB. Then, the requester becomes a new relay in its cell. Now, it is able to forward discovery message information from other requesters to the gNB. It starts scanning devices as a relay.

Step 26: Relay **0xF100** (**B**) unregistered successfully from the mobile network

The relay with C-RNTI **0xF100** (**B**) unregisters from the network for unknown reasons. The gNB takes off the allocated resources for this relay and updates its resource allocation table. The gNB fills the empty resources that left the unregistered relay. To do this, the gNB reallocates the resources associated with devices that had resources allocated next on the right (in time, at the same frequencies) of the unregistered device resources. In this case, the device with C-RNTI **0x3838** (**C**) has resources allocated next on the right of the unregistered device **0xF100**. Therefore, the gNB shifts the **0x3838** (**C**) resources in time domain to the left. Thus, the resource reallocation for **0x3838** (**C**) results in subcarriers 12th-23th, in the first BWP, and the next symbol after device **0x6DB6** (**A**) symbols: 3rd ($K_0 = 0$, $S = 2$, $L = 1$), for downlink; and subcarriers 12th-23th, in the first BWP, and the next symbol after device **0x6DB6** (**A**) symbols: 3rd ($K_2 = 0$, $S = 2$, $L = 1$), for uplink.

Step 27: The gNB informs device **0x3838** (**C**) of resource reallocation

The gNB informs relay **0x3838** (**C**) about resource reallocation via RRC signaling. The reallocation is informed the same way it was done in MSG4 (**8e**).

4.4 PARTIAL CONCLUSIONS

This chapter described the proposed framework to accelerate the traditional *Random-Access* procedure: RAA. The chapter introduced the new functions added to the devices looking for access, the devices with allocated resources, and the gNB as part of the RAA procedure. Then, it is depicted an example where some devices try to gain access in the mobile network. Once the devices obtain resources from the network they serve as a bridge between the new devices requesting resources and the gNB in order to reduce the access latency. It was shown how the gNB allocate resources differently from the traditional approach by fitting the resources' demands in the empty spaces of the time and frequency domains. The new resource allocation strategy could save bandwidth, especially when the devices accessing the network are IoT-based. Therefore, the RAA procedure offers low access latency and efficient resource allocation. In the next chapter, the RAA procedure is simulated and the results show that the RAA algorithms behave according to the description provided in this chapter.

CHAPTER 5: SIMULATION

In this chapter is simulated the behavior of the RAA procedure for a variable number of requesters and relays. The chapter compares some RAA alternatives with the traditional *Random-Access* approach. A time-discrete library is used for the simulation. Simulation results can be reproduced by downloading the code from the links provided in the following sections and running them.

5.1 SIMULATION CHARACTERISTICS

RAA has been simulated using the *Python* programming language and the *SimPy* module. The simulation involves the functions performed by the gNB for devices' registration, the procedures executed by the devices to connect to the mobile network, and the relay functionality when the devices obtain resources. All these functionalities are summarized in **Table 7**.

Table 7. *Functionalities of the gNB, the requester, and the relay.*

gNB	▪ Sending the SIB1 with the access probability for ACB. ▪ Scanning incoming *Random-Access Requests*. ▪ Scanning incoming messages from relays with the resource information of the devices that want to connect to the mobile network. ▪ Registering RA-RNTIs and assigning T/C-RNTIs when *Random-Access Requests* arrive. ▪ Sending back RACH responses. ▪ Checking the TTL of every device. ▪ Scanning incoming *RRC Connection Request* messages. ▪ Allocating resources. ▪ Sending back *RRC Connection Setup* messages.
Requester	▪ Scanning incoming SIB1. ▪ Executing the ACB procedure. ▪ Performing the *Random-Access* procedure. ▪ Sending discovery messages to nearby relays.

	▪ Generating the discovery frequencies used for the D2D communication. ▪ Stopping the *Random-Access* procedure and the device discovery when an acknowledge message from nearby relays or a RACH response arrives from the gNB. ▪ Starting again the *Random-Access* and the proposed RAA procedures when the *ra-ResponseWindow* expires. ▪ Sending *RRC Connection Request* message for resource allocation when RACH response arrives. ▪ Applying *back-off* when there is not RACH response after *ra-ResponseWindow*. ▪ Scanning incoming *RRC Connection Setup* message.
Relay	▪ Scanning incoming discovery messages. ▪ Generating the frequencies used for the discovery message scanning. ▪ Sending back a response for the discovery messages. ▪ Forwarding the discovery messages to the gNB in its mobile network resources.

5.2 SIMULATION PARAMETERS

The gNB provides a set of parameters for device's registration within MIB and SIB1. These parameters are not fixed and can vary in dependence of many factors such as the mobile network capacity, the number of devices attempting to connect to the network, the number of devices unregistered successfully from the network, and many others. Simulating all the different parameters is a very complex task. Thus, the simulation is based on the basic parameters offered by the network. For example, if all possible values from the *ra-ResponseWindow* are {sl1, ..., sl80}, the simulation only selects the first one (sl1). **Table 8** and **Table 9** summarize the parameters used in the simulation by the gNB and the devices, respectively. Most of these parameters are the same as those used by a real mobile network or real devices. However, the periodicity of SIB1 and the interval at which the devices are turned on are chosen to stress the network and verify the behavior of the RAA procedure in a chaotic scenario. **Table 10** describes the metrics used in the simulation.

Table 8. *Parameters used by the gNB in the simulation.*

Parameter	Value
Access probability sent by gNB	Real range [0.2, 0.8]
SIB1 periodicity	5 ms
Numerology	SCS = 15 kHz, Time slot = 66.67 µs

TTL	sl1 = 66.67 µs
RRC interval (wait for RRC request)	sl1 = 66.67 µs
Resource allocation capacity	∞
2.4 GHz slices (Groups of D2D)	52 (BLE) or 39 (Wi-Fi)
gNB transmission power	24 dBm
T_{ACB}	4 s
$T_{barring}$	$[0.7 + 0.6 \times \mathcal{U}[0, 1)] \times T_{ACB}$
Simulation time resolution	31.25 µs

Table 9. *Parameters used by the devices in the simulation.*

Parameter	Value
RBs requested by devices	Integer range [1 (12 subcarriers), 3 (36 subcarriers)]
Maximum number of symbols requested by devices	14
Transmission power	8 dBm (BLE) and 20 dBm (Wi-Fi)
Coverage area	BLE radius = 50 m, Wi-Fi radius = 100 m
D2D frequency used for inquiring	BLE frequency generator or random (3 frequencies)
D2D frequency used for scanning	BLE frequency generator
Interval where devices turn on	Real range [0 ms, 15 ms] (random)
Simulation time resolution	31.25 µs

Table 10. *Metrics used in the simulation.*

Metric	Description
Energy consumption	Calculated from Eq.(1) for every of the following cases: ▪ **Message from the requester to the relay**: $P = 6.31\,mW$ (BLE) or $P = 100\,mW$ (Wi-Fi), $m = 376\,bits$ => 8 bits (Preamble) + 32 bits (Access Address) + 312 bits (BLE Advertising Channel Protocol Data Unit) + 24 bits (CRC), and $R_b = 1\,Mb/s$. ▪ **Message from the relay to the gNB and from the gNB to the requester/relay**: $P = 251.19\,mW$, $m = 72\,bits$ (Assuming the length of the RAR message from **Figure 9**), and $R_b = 100\,Mb/s$.

Collisions	Number of collisions experienced: Number of transmissions in the same channel and at the same time. This is a dimensionless quantity.
Time	Total time for devices' registration in the mobile network. Given in milliseconds.

5.3 SIMULATION RESULTS

In this section, we present the results for five simulation scenarios where a number of devices try to connect to the network and there are no devices registered yet. The first scenario simulates the traditional *Random-Access* procedure without using the proposed RAA procedure. The other four scenarios were simulated using a customized *Random-Access* procedure and RAA at the same time. The new *Random-Access* procedure consists of the devices expanding the limits of a list where a new random value is selected every time a device is expecting a RAR and the *ra-ResponseWindow* expires. For example, a requester device sends a *Random-Access Request* and it waits for a response during *ra-ResponseWindow*. If that interval expires and there are no received responses, the device selects a random value from the list [0, 1]. This value represents the number of subsequent SIB1s that the device will not listen to. After that, the device can listen to incoming SIB1s and executes the *Random-Access* procedure again. If the device does not receive a RAR again, the list becomes [0, 1, 2] to ensure that the device could be delayed another SIB1 period to avoid collisions in a chaotic scenario.

The difference between the last four scenarios is the wireless technology used and the frequency generation in the 2.4 GHz band for the discovery message transmissions. The second and third scenarios involve RAA using two BLE algorithms: One using the traditional frequency generator for the transmission of the discovery messages, and the other generating the frequency randomly. The fourth and fifth scenarios simulate RAA using a customized version of Wi-Fi for both the traditional frequency generator and the random frequency generator. Wi-Fi is referred to as the classic Wi-Fi with its respective transmission power but it incorporates the BLE algorithms.

The last four simulations commented above were tested in another four circumstances. In the first one, the requester devices start the RAA functions after receiving the SIB1, and in the second one the requester devices start these functions since the beginning (before receiving SIB1) when the devices want to connect to the network. The last two circumstances are a variation of the second and third circumstances where the gNB selects the frequency that every registered device (relay) will use to listen to incoming discovery messages from remote requesters.

Figure 25 shows the four circumstances for the total energy spent by a specific number of devices that want to register in the mobile network. The figure is related to the scenario where is used Wi-Fi as wireless technology and the random frequency generator. The simulation shows the results for 100, 200, 300, 400, 500, 600, 700,

800, 900, and 1000 devices attempting to acquire resources from the network. Looking at the figure, the best circumstance—less energy consumption during the devices' registration—is the RAA procedure starting after the reception of the first SIB1 and using the traditional frequency generator to listen for incoming discovery messages (relay's function). The best circumstance is also selected in the rest of the scenarios. Once the best circumstances from every scenario are selected, they are compared with each other in terms of number of collisions, energy, and elapsed time during registration.

Note that the results of the simulations are the average of ten independent simulations. For each one of the individual simulations, it was employed a different seed for randomization. The seeds are in the integer range [0-9] for each simulation, respectively.

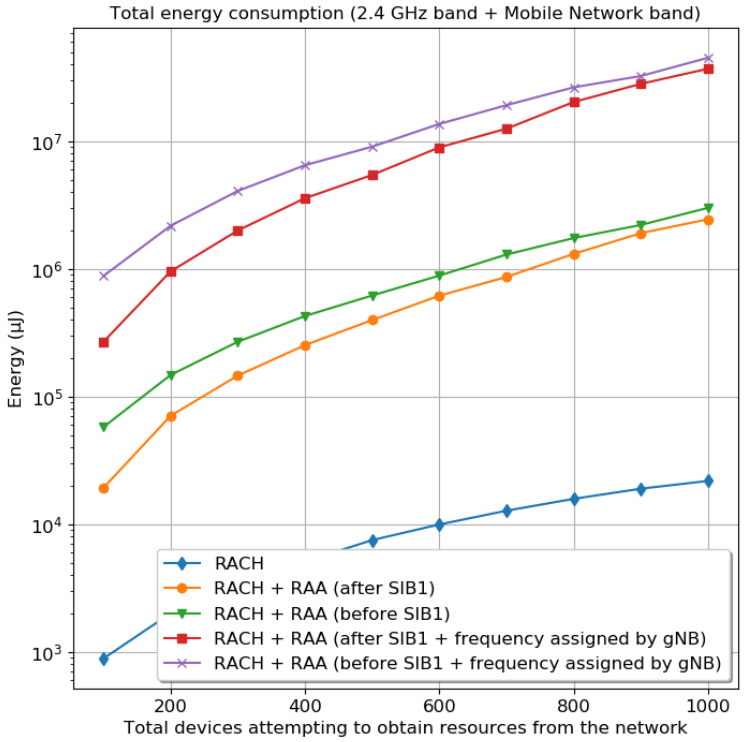


Figure 25. *Total energy spent by the devices (requesters and relays) attempting to obtain resources from the network in Wi-Fi using the random frequency generator.*

5.3.1 COLLISION ANALYSIS

When RAA is used, it is expected that the total number of collisions in the 2.4 GHz band is higher for Wi-Fi than for BLE because of the wider coverage range in Wi-Fi. However, the simulation results demonstrated that in the Wi-Fi scenario, where the random frequency generator is used, fewer collisions occur than in the BLE

scenario where is used the traditional frequency generator. This is because this Wi-Fi-based scenario registers all the devices in a shorter period of time compared with the others. The other Wi-Fi case (traditional frequency generator) is not fast enough during devices' registration and cannot reach a smaller number of collisions. In the mobile network band, the two Wi-Fi cases experiment fewer collisions than BLE. This happens for the same reason that was discussed before in the 2.4 GHz band. The Wi-Fi cases are faster registering devices than the collision per time unit rate in this band. **Figure 26** shows the total collisions in the two bands: the 2.4 GHz band and the mobile network band for the two BLE and Wi-Fi scenarios. From the figure, the BLE case that uses the random frequency generator is the best case. It experiences a smaller number of collisions in comparison with the other RAA scenarios. However, this best case involves higher number of collisions compared with the classic *Random-Access* procedure.

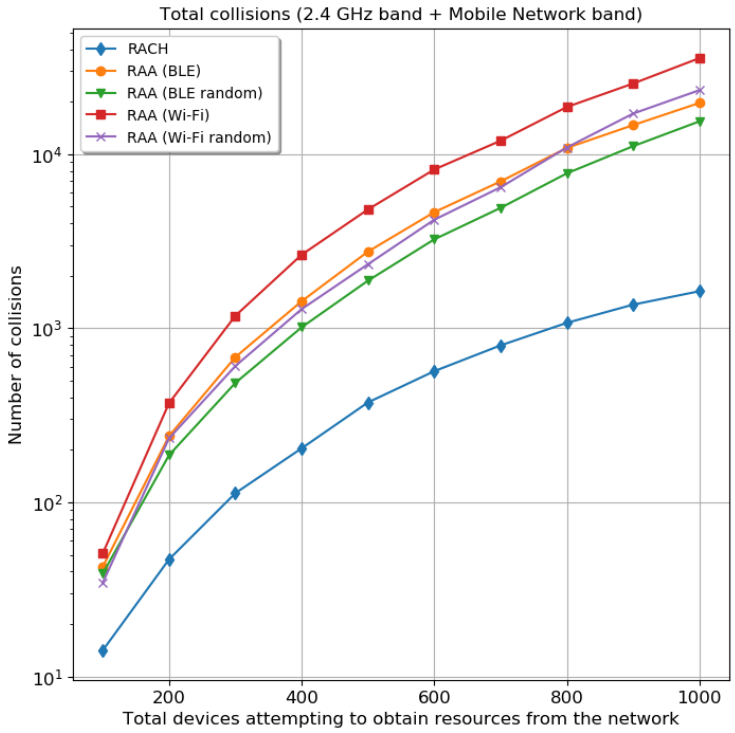


Figure 26. *Total number of collisions in all bands for the classic Random-Access Channel (RACH) procedure and four different Random-Access Accelerator (RAA) procedures.*

5.3.2 ENERGY ANALYSIS

The total energy spent in the 2.4 GHz band is higher in the Wi-Fi cases as expected because of the higher transmission power associated with this wireless technology. The energy spent in the Wi-Fi cases is 16 times the energy spent by BLE. Unlike the energy spent in the 2.4 GHz band, the energy consumption in the mobile network band for the Wi-Fi cases is similar to the energy spent by the BLE cases. The similitude is due to the same power transmission used in all RAA procedures when the devices are communicating with the gNB in the mobile network band. In this case, the devices use the same power regardless of the D2D technology. **Figure 27** shows the total energy spent by all devices during registration in the two bands: the 2.4 GHz band and the mobile network band for the two Wi-Fi and BLE scenarios. From this figure, the BLE cases experience lower energy consumption. Although the Wi-Fi cases are faster for devices' registration, the energy spent in every transmission far exceeds (16 times) the energy that the BLE approach consumes.

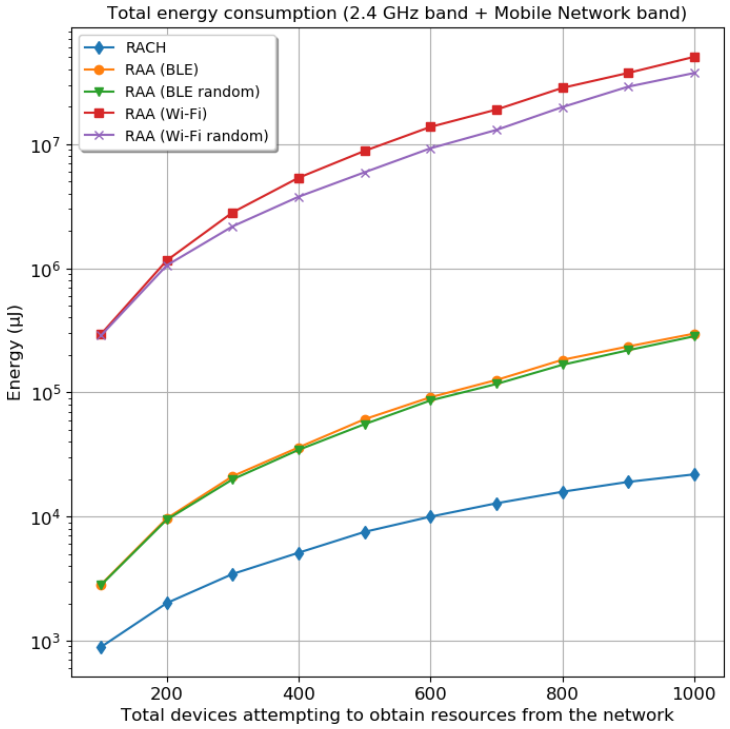


Figure 27. *Total energy spent in all bands by the devices (requesters and relays) for the classic RACH and four different RAA procedures.*

Figure 28 shows the energy spent by the gNB in every scenario. This figure infers that RAA is much faster than the classic *Random-Access* procedure because it is expected that the energy spent by the gNB remains almost constant for every procedure. Then, if there is a great difference between the two procedures about

energy consumption by the gNB, is because of the great difference about the elapsed time for devices' registration in every procedure. In all scenarios, the gNB transmits periodically the same amount of information. However, there is a small variation for these scenarios because of the number of responses the gNB sends to the requester devices. The energy consumption of the gNB depends on the number of collisions and the elapsed time for all devices' registration. More collisions and more delays in devices' registration mean that the gNB will receive more *Random-Access Requests* and the gNB will send more RAR, which incurs in more energy consumption.

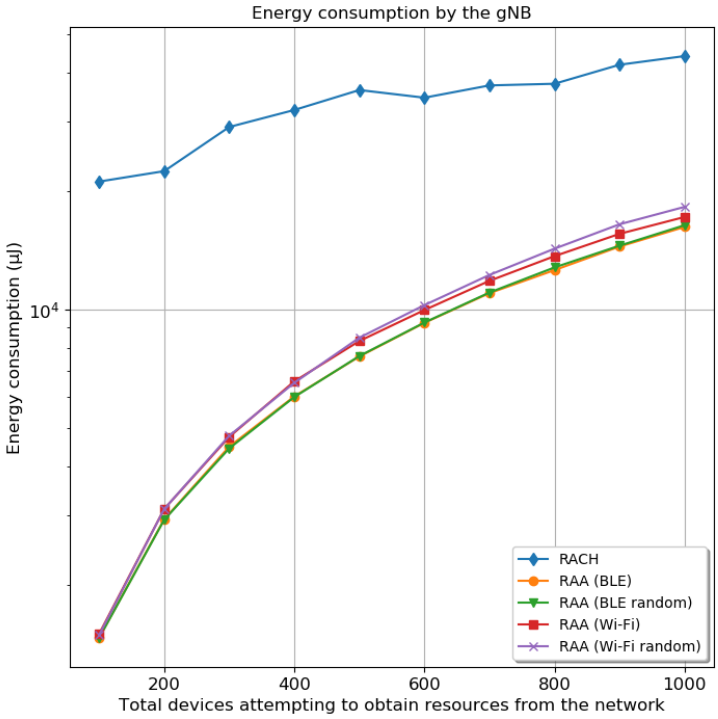


Figure 28. *Total energy spent by the gNB for the classic RACH and four different RAA procedures.*

5.3.3 DELAY ANALYSIS

From **Figure 29**, it can be seen that the elapsed time for all devices registration when are used the RAA procedures overcomes the classic *Random-Access* procedure by far. The elapsed time for all the RAA scenarios is always around 100 ms or less; meanwhile, the elapsed time for the classic *Random-Access* procedure is always above 10 s. This great difference is because of the ACB algorithm used in the classic *Random-Access* procedure. On one hand, the ACB back-offs many attempts of devices' registration in the order of seconds, causing that many of these devices register in the network in very distanced intervals, as it is depicted in **Figure 30a**. On the other hand, the RAA approach redistributes the devices' registration more

regularly in time, meaning that there are no large periods of isolation between groups of devices, see **Figure 30b**.

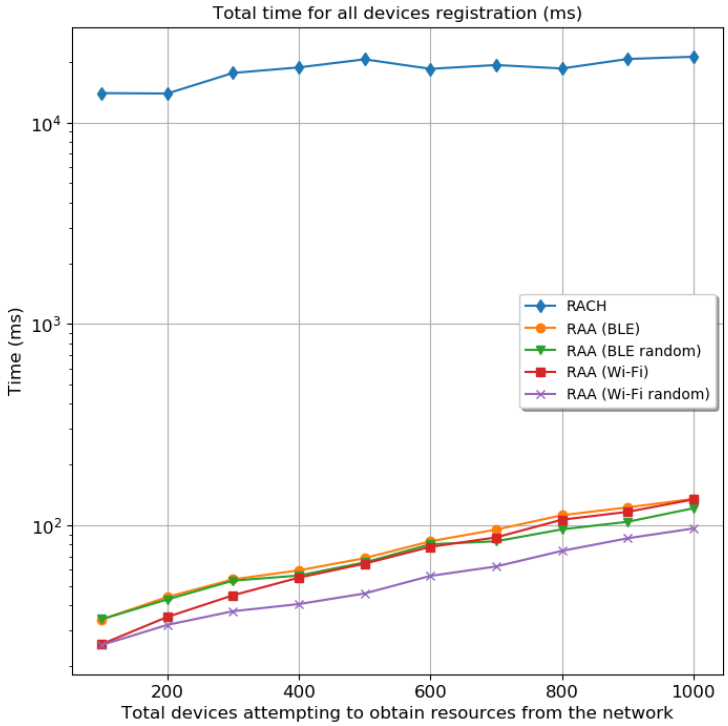


Figure 29. *Elapsed time for all devices registration for the classic RACH and four different RAA procedures.*

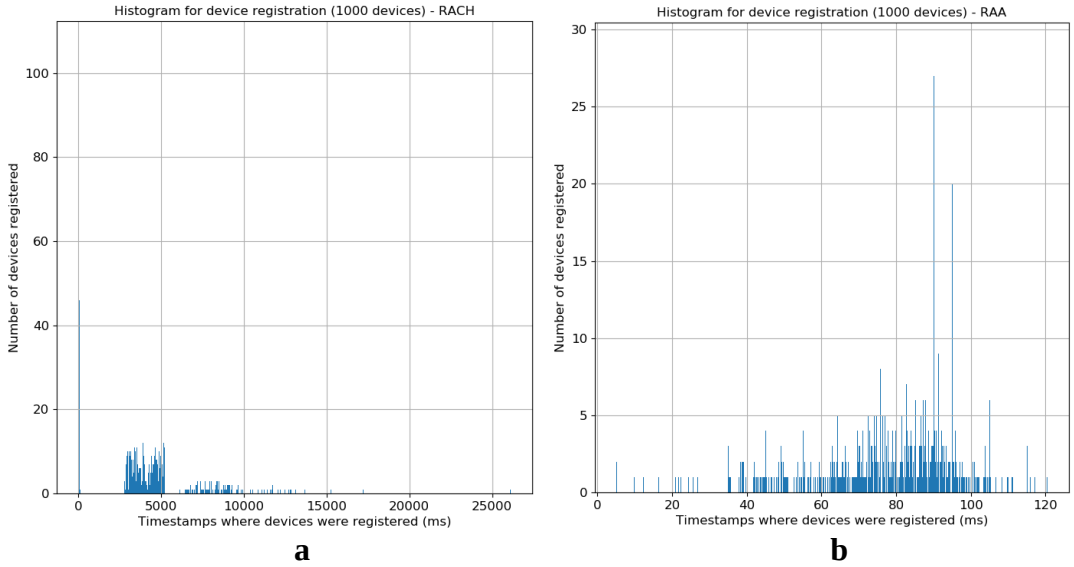


Figure 30. *Number of devices registered per time unit for the (**a**) classic Random-Access procedure and the (**b**) RAA procedure.*

5.3.4 RESOURCE ALLOCATION

Figure 31 and **Figure 32** show the number of allocated resources for the classic *Random-Access* procedure and the RAA approach. In the x-axis are depicted the resources' demands for 9 cases. For example, the first one (12/5) means that all the devices request 12 subcarriers and 5 symbols for their data transmission/reception. The y-axis represents the total number of subcarriers offered by the gNB. The allocation process is done by the gNB, placing the resources offered to every device alongside other devices' resource until the 14 symbols of a subcarrier are occupied. From the figure, it can be seen that the gNB allocates the requested resources with more flexibility when it is used the proposed RAA procedure. In the RAA case, the devices only request the resources they need. In the classic *Random-Access* procedure, the devices request resources but they do not notify the exact number of subcarriers and symbols they need; therefore, the gNB assigns an entire RB to every device. The resource allocation made by the use of RAA overcomes the classic procedure especially when the devices that want resources from the mobile network are IoT devices. These kinds of devices require a small number of symbols for their transmissions. They are expected to request between 1 and 5 symbols, and less than 12 subcarriers. In that case, the RAA approach overcomes the classic resource allocation procedure in 22% for the simulation results of 100 devices requesting resources (**Figure 31a**, **b**). When 1000 devices attempt to obtain resources, the resource allocation procedure using RAA overcomes the classic procedure in 68% (**Figure 32a**, **b**).

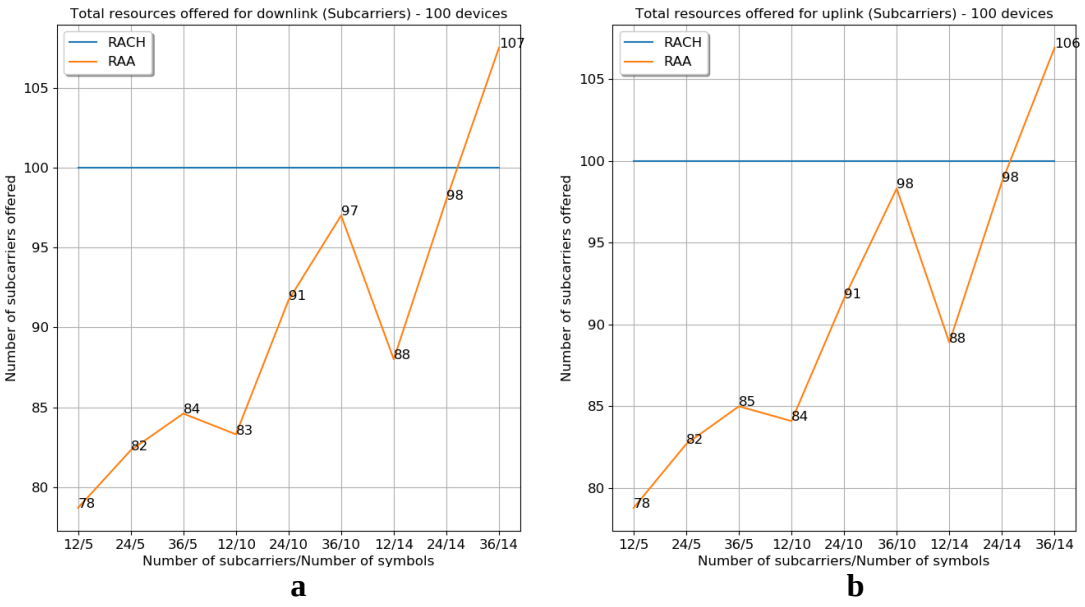


Figure 31. *Resource allocation (number of allocated subcarriers) in the (**a**) downlink and the (**b**) uplink when there are 100 devices registered in the mobile network by using the classic resource allocation procedure (the result of RACH execution) and the RAA approach.*

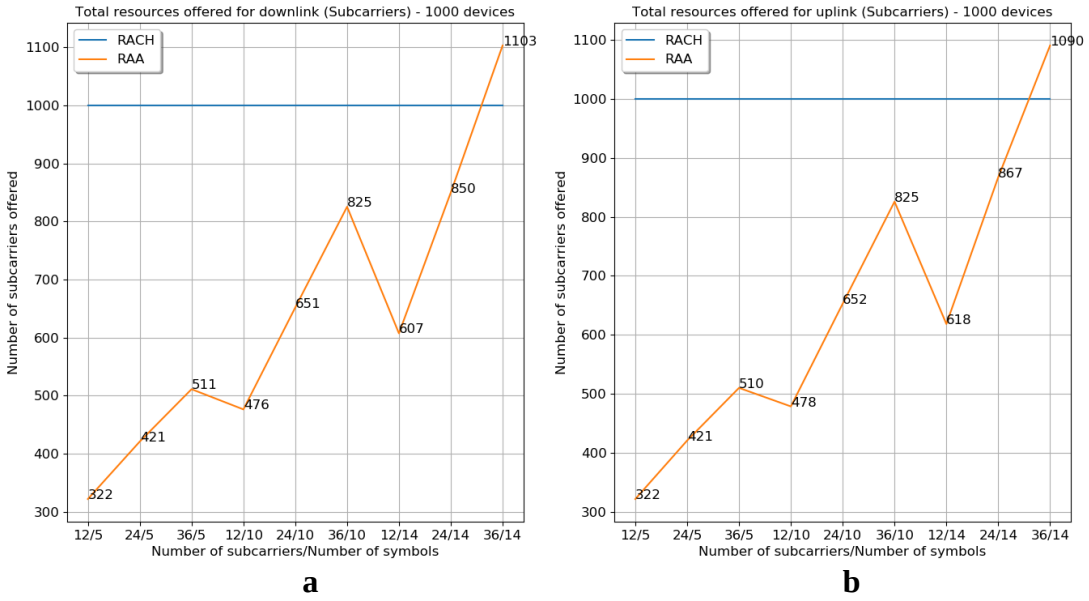


Figure 32. *Resource allocation (number of allocated subcarriers) in the (**a**) downlink and the (**b**) uplink when there are 1000 devices registered in the mobile network by using the classic resource allocation procedure (the result of RACH execution) and the RAA approach.*

5.3.5 A MORE REALISTIC SCENARIO

All the above simulations were made exclusively in this work, and they are available on GitHub: https://github.com/Abel1027/Framework-To-Speed-Up-RACH-BLE/tree/main/RAA. As they represent the scenario where there are no registered devices in the mobile network at the time all devices arrive, it is interesting to simulate a more real scenario. A more realistic situation concerns a group of devices connected to the gNB and another set of devices, frequently smaller than the connected ones, trying to connect to the network. In the simulation, the same five cases discussed before were used (RACH and the best four cases of the proposed RAA procedure) with a group of 1000 connected devices. This means that, when the first requester device attempting to connect to the network is turned on (gets into the mobile network), there are 1000 devices that can serve as a relay for it. However, the requester device will camp in the coverage area of a subgroup of connected devices because they are distributed randomly along the cell coverage area. **Figure 33** and **Figure 34** show the results for the total energy consumption and the elapsed time for all devices' registration in every scenario, respectively. These simulations are found on GitHub too: https://github.com/Abel1027/Framework-To-Speed-Up-RACH-BLE/tree/main/RAA-1000-Relays-Connected.

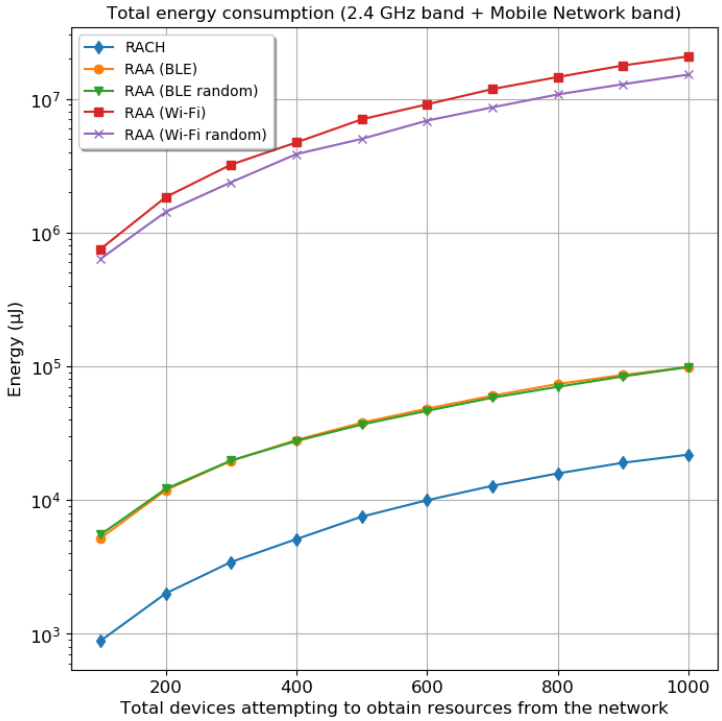


Figure 33. *Total energy spent by the devices (requesters and relays) in all bands when there are 1000 connected devices to the mobile network before the new requesters start attempting to obtain resources from the network.*

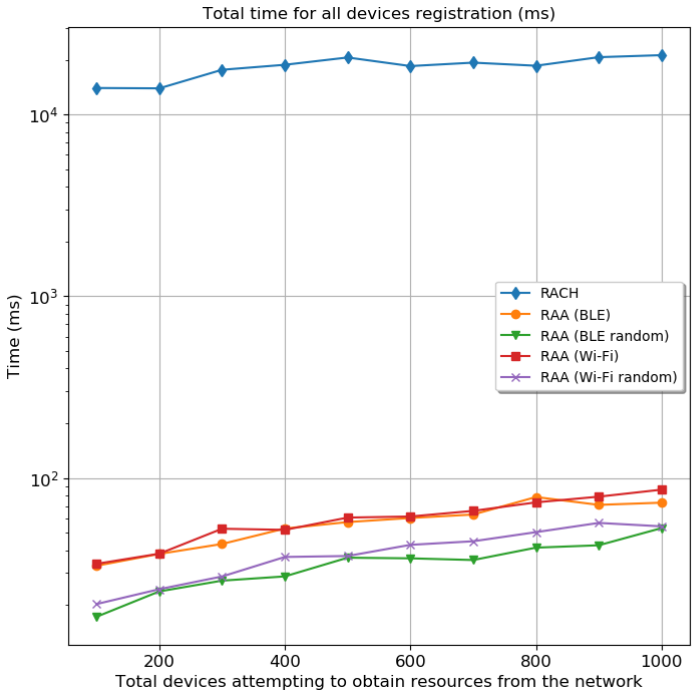


Figure 34. *Elapsed time for all requesting devices in all bands when there are 1000 connected devices to the mobile network before the new requesters start attempting to obtain resources from the network.*

In this case, the total energy consumption decreases from 400 devices requesting access. The RAA case where is used BLE with the default frequency generator, the energy consumption is reduced by a few energy units compared with the simulation where there are no registered devices in the network and only 400 new devices try to gain access. However, the energy consumption decreases by doing the same comparison when 1000 devices try to gain access. The total time for the devices' registration decreases too. The elapsed time is always bellow 100 ms.

5.4 PARTIAL CONCLUSIONS

In this chapter it was simulated the traditional *Random-Access* procedure and the RAA procedure. Four different customized technologies were used for device discovery and data transmission/reception: BLE using the classic frequency generator, BLE using a random frequency generator, and the same two approaches but using the Wi-Fi transmission power. The four technologies are integrated into the RAA procedure resulting in four RAA alternatives. All the RAA alternatives' performances were compared with the traditional *Random-Access* procedure for two cases: first, a bunch of devices starts looking for access to the mobile network when there are not connected devices yet, and second, the same number of devices try to gain access to the network but there are already 1000 devices with allocated resources. In the first case, the number of collisions experienced by the devices in the unlicensed band and the mobile network band was higher for the four RAA procedures than the number of collisions experienced by the devices in the same bands using the traditional *Random-Access* procedure, as it is depicted in **Figure 26**. The energy spent by the devices was also a bit higher for the RAA procedures (**Figure 27**). However, the energy consumption of the gNB is lower for the RAA procedures than the traditional *Random-Access* procedure (**Figure 28**). In the second case, where 1000 devices are acting as relays, the energy consumption of the devices is still a bit higher than the traditional *Random-Access* procedure when the two RAA procedures with the Wi-Fi transmission power are used. However, if the RAA procedures use the BLE transmission power, the energy consumption of the devices is in the same order than the traditional *Random-Access* procedure, as it is depicted in **Figure 33**. In both cases, the four RAA procedures reduce the elapsed time for the devices' registration by 99% in comparison to the traditional *Random-Access* approach (**Figure 29** and **Figure 34**). The comparison results also show that the number of devices registered by time unit is more regular when the four RRA procedures are used than the traditional *Random-Access* procedure. In the traditional *Random-Access* approach, most of the devices register at the beginning of the access procedure, and then the number of devices registered per time unit is reduced proportionally to the elapsed time (**Figure 30**). The RAA procedures also overcome the traditional *Random-Access* procedure during resource allocation because RAA places every resource's demand

in the first empty space it fits from the resources grid. In this aspect, RAA overcomes the traditional procedure in 68% if the registered devices are IoT based (**Figure 32**).

The energy consumption related to the individual processing of each device has not been studied. It will be analyzed in future work. However, it is expected that the processing energy spent on every device will be low enough to preserve the required life cycle of the device battery. It was checked that with only 1000 connected devices acting like relays and distributed randomly, RAA incurs in almost the same energy consumption for the BLE case like in the *Random-Access* procedure. Therefore, if the gNB commands that 1000 different devices act like relays every pre-established time in a super-populated NR cell, the low energy requirement for IoT devices will be fulfilled because just a very small number of devices is processing incoming data from nearby devices.

CHAPTER 6: CONCLUSIONS

In NR, every time a device camps inside the coverage area of a cell or a device moves on from an LTE cell to an NR cell, it has to perform a *Random-Access* procedure to obtain resources from the network. During the execution, the device competes with other devices that are also requesting resources. This fight becomes harder when the number of devices contending for resource allocation is high because there are limited access opportunities, especially in mMTC scenarios. Therefore, the total time for the devices' registration increases too much.

As the first step, in this work was studied the traditional *Random-Access* procedure. This allowed understanding the behavior of the access process to search for solutions in order to minimize access latency. The most important aspects studied were the ACB algorithm due to the high access latency that it compromises and the resource allocation strategy that allocates unnecessary resources in many cases, especially those that involve IoT devices.

D2D communication was introduced as one of the enablers for mMTC. Therefore, the principal D2D technologies such as Wi-Fi Direct, Bluetooth, and BLE were analyzed to select the technology with the best performance about energy consumption and discovery latency. In this case, BLE spends less energy than the other D2D technologies and BLE is also faster during the discovery procedure. Therefore, BLE was selected to be part of a new *Random-Access* procedure.

It was proposed the framework RAA, which uses BLE for D2D communication to transmit and receive resource requirements messages between the devices that want to be registered in the mobile network and the devices that have already allocated resources. The registered devices act as relays and forward all the resources necessities of the no-connected devices to the gNB. Then, the gNB directly transmits back the requested resources to the devices looking for access.

In this work it was simulated the traditional *Random-Access* procedure and the RAA approach. The results indicated that the RAA procedure is 99% faster than the conventional *Random-Access* procedure. RAA also manages resource allocation more efficiently than the *Random-Access* approach. The energy consumed by the devices during the RAA procedure is a bit higher than the traditional *Random-Access* approach; however, the energy consumption is in the same order for both procedures.

RECOMMENDATIONS

For future works we recommend:

- Test the RAA procedure in a real scenario with real devices and a real gNB.

- Investigate and design new procedures for the devices' discovery in massive communications to decrease discovery latency and energy consumption during the discovery procedure.

- Check the results of statistical algorithms that perform a probabilistic analysis of access attempts at peak times to manage transmissions more efficiently. The gNB could inform the devices of the level of concurrency on the network in order to back-off some access attempts.

REFERENCES

3GPP. 2017. “LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) Protocol Specification (3GPP TS 36.321 Version 13.5.0 Release 13).” https://www.etsi.org/deliver/etsi_ts/136300_136399/136321/13.05.00_60/ts_136321v130500p.pdf.

———. 2018a. “5G; NR; Base Station (BS) Radio Transmission and Reception (3GPP TS 38.104 Version 15.2.0 Release 15).” https://www.etsi.org/deliver/etsi_ts/138100_138199/138104/15.02.00_60/ts_138104v150200p.pdf.

———. 2018b. “5G; NR; Medium Access Control (MAC) Protocol Specification (3GPP TS 38.321 Version 15.3.0 Release 15).” https://www.etsi.org/deliver/etsi_ts/138300_138399/138321/15.03.00_60/ts_138321v150300p.pdf.

———. 2018c. “5G; NR; Physical Layer Procedures for Control (3GPP TS 38.213 Version 15.3.0 Release 15).” https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/15.03.00_60/ts_138213v150300p.pdf.

———. 2018d. “5G; NR; Physical Layer Procedures for Data (3GPP TS 38.214 Version 15.3.0 Release 15).” https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/15.03.00_60/ts_138214v150300p.pdf.

“5G Evolution Towards a Super-connected World.” 2018. In *Practical Guide to LTE-A, VoLTE and IoT: Paving the Way towards 5G*, 382–443. Wiley. https://doi.org/10.1002/9781119063407.ch8.

Alsaeedy, Alaa A. R., and Edwin K. P. Chong. 2019. “Mobility Management for 5G IoT Devices: Improving Power Consumption With Lightweight Signaling Overhead.” *IEEE Internet of Things Journal* 6 (5): 8237–47. https://doi.org/10.1109/JIOT.2019.2920628.

Althumali, Huda, and Mohamed Othman. 2018. “A Survey of Random Access Control Techniques for Machine-to-Machine Communications in LTE/LTE-A Networks.” *IEEE Access* 6: 74961–83. https://doi.org/10.1109/ACCESS.2018.2883440.

Bensky, Alan. 2019. *Short-Range Wireless Communication*. Newnes.

Camps-Mur, Daniel, Andres Garcia-Saavedra, and Pablo Serrano. 2013. "Device-to-Device Communications with Wi-Fi Direct: Overview and Experimentation." *IEEE Wireless Communications* 20 (3): 96–104. https://doi.org/10.1109/MWC.2013.6549288.

Chakrapani, Arvind. 2020. "On the Design Details of SS/PBCH, Signal Generation and PRACH in 5G-NR." *IEEE Access* 8: 136617–37. https://doi.org/10.1109/ACCESS.2020.3010500.

Chandramouli, Devaki, Rainer Liebhart, and Juho Pirskanen. 2019. *5G for the Connected World*. John Wiley & Sons.

Chen, Hung-Chen, Mei-Ju Shih, and Chie-Ming Chou. 2020. Random access procedure in next generation wireless networks. United States US20200100294A1, filed September 17, 2019, and issued March 26, 2020. https://patents.google.com/patent/US20200100294A1/en.

Cho, Keuchul, Gisu Park, Wooseong Cho, Jihun Seo, and Kijun Han. 2016. "Performance Analysis of Device Discovery of Bluetooth Low Energy (BLE) Networks." *Computer Communications* 81 (May): 72–85. https://doi.org/10.1016/j.comcom.2015.10.008.

Choi, Jinho. 2020. "On Fast Retrial for Two-Step Random Access in MTC." *IEEE Internet of Things Journal*, 1–1. https://doi.org/10.1109/JIOT.2020.3012449.

Duflot, Marie, Marta Kwiatkowska, Gethin Norman, and David Parker. 2006. "A Formal Analysis of Bluetooth Device Discovery." *International Journal on Software Tools for Technology Transfer* 8 (6): 621–32. https://doi.org/10.1007/s10009-006-0014-x.

Enescu, Mihai. 2020. *5G New Radio: A Beam-Based Air Interface*. John Wiley & Sons.

G. Müller, Klaus, Tony Vignaux, and Ontje Lünsdorf Stefan Scherfke. n.d. *Simpy: A Process-Based Discrete-Event Simulation Framework Based on Standard Python*. Accessed April 15, 2020. https://simpy.readthedocs.io/en/latest/.

Han, Bin, Vincenzo Sciancalepore, Oliver Holland, Mischa Dohler, and Hans D. Schotten. 2019. "D2D-Based Grouped Random Access to Mitigate Mobile Access Congestion in 5G Sensor Networks." *IEEE Communications Magazine* 57 (9): 93–99. https://doi.org/10.1109/MCOM.001.1701032.

Hussain, Fatima, Alagan Anpalagan, and Rath Vannithamby. 2017. "Medium Access Control Techniques in M2M Communication: Survey and Critical Review." *Transactions on Emerging Telecommunications Technologies* 28 (1): e2869. https://doi.org/10.1002/ett.2869.

Jatti, Anand, Madhvi Kannan, R M Alisha, P Vijayalakshmi, and Shrestha Sinha. 2016. "Design and Development of an IOT Based Wearable Device for the Safety and

Security of Women and Girl Children." In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 1108–12. https://doi.org/10.1109/RTEICT.2016.7808003.

Jeon, Jeongho. 2018. "NR Wide Bandwidth Operations." *IEEE Communications Magazine* 56 (3): 42–46. https://doi.org/10.1109/MCOM.2018.1700736.

Jia Liu, Canfeng Chen, and Yan Ma. 2012. "Modeling and Performance Analysis of Device Discovery in Bluetooth Low Energy Networks." In *2012 IEEE Global Communications Conference (GLOBECOM)*, 1538–43. https://doi.org/10.1109/GLOCOM.2012.6503332.

KIM, Ji Hyung, and Mi Young YUN. 2020. Method and apparatus for signal configuration for mobile base station. United States US20200153500A1, filed November 12, 2019, and issued May 14, 2020. https://patents.google.com/patent/US20200153500A1/en.

Kryukov, Yakov, Dmitriy Pokamestov, and Eugeniy Rogozhnikov. 2019. "Cell Search and Synchronization in 5G NR." *ITM Web of Conferences* 30: 04007. https://doi.org/10.1051/itmconf/20193004007.

Larmo, Anna, and Riikka Susitaival. 2015. Mtc rach procedure. United States US20150312938A1, filed November 26, 2013, and issued October 29, 2015. https://patents.google.com/patent/US20150312938A1/en.

Lei, Wan, Anthony C. K. Soong, Liu Jianghua, Wu Yong, Brian Classon, Weimin Xiao, David Mazzarese, Zhao Yang, and Tony Saboorian. 2019. *5G System Design: An End to End Perspective*. Springer Nature.

Leyva-Mayorga, Israel, Cedomir Stefanovic, Petar Popovski, Vicent Pla, and Jorge Martinez-Bauset. 2019. "Random Access for Machine-Type Communications." In *Wiley 5G Ref*, 1–21. American Cancer Society. https://doi.org/10.1002/9781119471509.w5GRef031.

Lien, Shao-Yu, Shin-Lin Shieh, Yenming Huang, Borching Su, Yung-Lin Hsu, and Hung-Yu Wei. 2017. "5G New Radio: Waveform, Frame Structure, Multiple Access, and Initial Access." *IEEE Communications Magazine* 55 (6): 64–71. https://doi.org/10.1109/MCOM.2017.1601107.

Lin, Xingqin, Jingya Li, Robert Baldemair, Jung-Fu Thomas Cheng, Stefan Parkvall, Daniel Chen Larsson, Havish Koorapaty, et al. 2019. "5G New Radio: Unveiling the Essentials of the Next Generation Wireless Access Technology." *IEEE Communications Standards Magazine* 3 (3): 30–37. https://doi.org/10.1109/MCOMSTD.001.1800036.

Mazin, Asim, Mohamed Elkourdi, and Richard D. Gitlin. 2018. "Accelerating Beam Sweeping in MmWave Standalone 5G New Radios Using Recurrent Neural

Networks." In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 1–4. https://doi.org/10.1109/VTCFall.2018.8690810.

M.R., Sriharsha, Sreekanth Dama, and Kiran Kuchi. 2017. "A Complete Cell Search and Synchronization in LTE." *EURASIP Journal on Wireless Communications and Networking* 2017 (1): 101. https://doi.org/10.1186/s13638-017-0886-3.

Omri, Aymen, Mohammed Shaqfeh, Abdelmohsen Ali, and Hussein Alnuweiri. 2019. "Synchronization Procedure in 5G NR Systems." *IEEE Access* 7: 41286–95. https://doi.org/10.1109/ACCESS.2019.2907970.

Parkvall, Stefan, Erik Dahlman, Anders Furuskar, and Mattias Frenne. 2017. "NR: The New 5G Radio Access Technology." *IEEE Communications Standards Magazine* 1 (4): 24–30. https://doi.org/10.1109/MCOMSTD.2017.1700042.

RYU, Hyunseok, Yongseok Kim, Peng XUE, Hyunkyu Yu, Sangwon Choi, and Kuyeon WHANG. 2020. Method and apparatus for uplink transmission in wireless communication system. United States US20200280868A1, filed May 19, 2020, and issued September 3, 2020. https://patents.google.com/patent/US20200280868A1/en.

Schreiber, Gerhard, and Marcos Tavares. 2018. "5G New Radio Physical Random Access Preamble Design." In *2018 IEEE 5G World Forum (5GWF)*, 215–20. https://doi.org/10.1109/5GWF.2018.8517052.

Schulz, Philipp, Maximilian Matthe, Henrik Klessig, Meryem Simsek, Gerhard Fettweis, Junaid Ansari, Shehzad Ali Ashraf, et al. 2017. "Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture." *IEEE Communications Magazine* 55 (2): 70–78. https://doi.org/10.1109/MCOM.2017.1600435CM.

Seo, Hyowoon, Jun-Pyo Hong, and Wan Choi. 2019. "Low Latency Random Access for Sporadic MTC Devices in Internet of Things." *IEEE Internet of Things Journal* 6 (3): 5108–18. https://doi.org/10.1109/JIOT.2019.2896620.

Soleymani, Dariush M., André Puschmann, Elke Roth-Mandutz, Jens Mueckenheim, and Andreas Mitschele-Thiel. 2016. "A Hierarchical Radio Resource Management Scheme for next Generation Cellular Networks." In *2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 416–20. https://doi.org/10.1109/WCNCW.2016.7552735.

SONG, Huayue, and Yunjung Yi. 2020. Method and device for transmitting or receiving data in wireless communication system. United States US20200236638A1, filed October 2, 2018, and issued July 23, 2020. https://patents.google.com/patent/US20200236638A1/en.

Sun, Weiping, Changmok Yang, Sunggeun Jin, and Sunghyun Choi. 2016. "Listen Channel Randomization for Faster Wi-Fi Direct Device Discovery." In *IEEE INFOCOM 2016*

- *The 35th Annual IEEE International Conference on Computer Communications*, 1–9. https://doi.org/10.1109/INFOCOM.2016.7524342.

Tello-Oquendo, Luis, José-Ramón Vidal, Vicent Pla, and Luis Guijarro. 2018. "Dynamic Access Class barring Parameter Tuning in LTE-A Networks with Massive M2M Traffic." In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 1–8. https://doi.org/10.23919/MedHocNet.2018.8407086.

Vikhrova, Olga, Chiara Suraci, Angelo Tropeano, Sara Pizzi, Konstantin Samouylov, and Giuseppe Araniti. 2019. "Enhanced Radio Access Procedure in Sliced 5G Networks." In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 1–6. https://doi.org/10.1109/ICUMT48472.2019.8970776.

Vilgelm, Mikhail, Sergio Rueda Linares, and Wolfgang Kellerer. 2017. "Enhancing Cellular M2M Random Access with Binary Countdown Contention Resolution." In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 1–6. https://doi.org/10.1109/PIMRC.2017.8292435.

Vilgelm, Mikhail, Sergio Rueda Liñares, and Wolfgang Kellerer. 2019. "Dynamic Binary Countdown for Massive IoT Random Access in Dense 5G Networks." *IEEE Internet of Things Journal* 6 (4): 6896–6908. https://doi.org/10.1109/JIOT.2019.2912424.

Wang, Jian, and Richard Rouil. 2018. "Assessing Coverage and Throughput for D2D Communication." In *2018 IEEE International Conference on Communications (ICC)*, 1–6. https://doi.org/10.1109/ICC.2018.8422099.

Wei, Xusheng, and Timothy Moulsley. 2019. Cell Search and Synchronization in 5G. United States US20190253959A1, filed April 25, 2019, and issued August 15, 2019. https://patents.google.com/patent/US20190253959A1/en.

Xu, Zetao, Yang Zhang, Ao Shen, Bao Guo, Yuehua Han, and Yi Liu. 2019. "Initial Analysis of the Cell Selection Progress in SA of 5G NR." In *Signal and Information Processing, Networking and Computers*, edited by Songlin Sun, Meixia Fu, and Lexi Xu, 495–504. Lecture Notes in Electrical Engineering. Singapore: Springer. https://doi.org/10.1007/978-981-13-7123-3_58.

Zaidi, Ali A., Robert Baldemair, Vicent Moles-Cases, Ning He, Karl Werner, and Andreas Cedergren. 2018. "OFDM Numerology Design for 5G New Radio to Support IoT, EMBB, and MBSFN." *IEEE Communications Standards Magazine* 2 (2): 78–83. https://doi.org/10.1109/MCOMSTD.2018.1700021.

Zhang, Peng. 2008. "Data Communications in Distributed Control System." *Industrial Control Technology: A Handbook for Engineers and Researchers*, 675–774.

Zhou, Wei, and Selwyn Piramuthu. 2014. “Security/Privacy of Wearable Fitness Tracking IoT Devices.” In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–5. https://doi.org/10.1109/CISTI.2014.6877073.