THE UNIVERSITY
of EDINBURGH

Towards Efficient Support for Massive Internet of Things over Cellular Networks

Galini Tsoukaneri

Doctor of Philosophy

Institute of Computing Systems Architecture

School of Informatics

University of Edinburgh

2019

Abstract

The usage of Internet of Things (IoT) devices over cellular networks is seeing tremendous growth in recent years, and that growth in only expected to increase in the near future. While existing 4G and 5G cellular networks offer several desirable features for this type of applications, their design has historically focused on accommodating traditional mobile devices (e.g. smartphones). As IoT devices have very different characteristics and use cases, they create a range of problems to current networks which often struggle to accommodate them at scale. Although newer cellular network technologies, such as Narrowband-IoT (NB-IoT), were designed to focus on the IoT characteristics, they were extensively based on 4G and 5G networks to preserve interoperability, and decrease their deployment cost. As such, several inefficiencies of 4G/5G were also carried over to the newer technologies.

This thesis focuses on identifying the core issues that hinder the large scale deployment of IoT over cellular networks, and proposes novel protocols to largely alleviate them. We find that the most significant challenges arise mainly in three distinct areas: connection establishment, network resource utilisation and device energy efficiency. Specifically, we make the following contributions. First, we focus on the connection establishment process and argue that the current procedures, when used by IoT devices, result in increased numbers of collisions, network outages and a signalling overhead that is disproportionate to the size of the data transmitted, and the connection duration of IoT devices. Therefore, we propose two mechanisms to alleviate these inefficiencies. Our first mechanism, named ASPIS, focuses on both the number of collisions and the signalling overhead simultaneously, and provides enhancements to increase the number of successful IoT connections, without disrupting existing background traffic. Our second mechanism focuses specifically on the collisions at the connection establishment process, and used a novel approach with Reinforcement Learning, to decrease their number and allow a larger number of IoT devices to access the network with fewer attempts.

Second, we propose a new multicasting mechanism to reduce network resource utilisation in NB-IoT networks, by delivering common content (e.g. firmware updates) to multiple similar devices simultaneously. Notably, our mechanism is both more efficient during multicast data transmission, but also frees up resources that would otherwise be perpetually reserved for multicast signalling under the existing scheme.

Finally, we focus on energy efficiency and propose novel protocols that are designed for the unique usage characteristics of NB-IoT devices, in order to reduce the

device power consumption. Towards this end, we perform a detailed energy consumption analysis, which we use as a basis to develop an energy consumption model for realistic energy consumption assessment. We then take the insights from our analysis, and propose optimisations to significantly reduce the energy consumption of IoT devices, and assess their performance.

Acknowledgements

I would like to extend my gratitude to my supervisor, Dr. Mahesh K. Marina for his guidance and patience during these past four years. I also am grateful to Dr. Franscisco Gracia and Dr. Massimo Condoluci, whose help was invaluable in my research.

I am fortunate to have worked along side some amazing collegues, Dr Paul Patras, Dr Lito Kriara, Dr Saravana Rathinakumar, Dr Yota Katsikouli, Dr Xenofon Foukas, Dr Praveen Tammana, Dr Ursula Challita, Mohammed Kaseem, Alex Dawson and Rajkarn Singh.

I am very thankful to my family, Dimitris, Nikoletta and George, who were always there for me when I needed it the most, with endless understanding, their unconditional love, their unlimited support for my dreams no matter how crazy, and for never stop believing in me.

Last, but certainly not least, my most special thanks to my husband, Anestis, whose understanding and encouragement helped me from the beginning till the end of this amazing journey. He has always been there to celebrate my successes and to help back on my feet during the difficulties. I could not have done it without him.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. Some of the material used in this thesis has been published in the following papers:

- Galini Tsoukaneri and Xenofon Foukas and Mahesh K. Marina. *"ASPIS: A Holistic and Practical Mechanism for Efficient MTC Support over Mobile Networks"*. In Proc. IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Orlando, Florida, Oct. 2017

- Luca Feltrin and Galini Tsoukaneri and Massimo Condoluci and Chiara Buratti and Toktam Mahmoodi and Mischa Dohler and Roberto Verdone. *"Narrowband-IoT : A survey on downlink and uplink perspectives"*. IEEE Wireless Communications Magazine, Feb. 2019.

- Galini Tsoukaneri and Massimo Condoluci and Toktam Mahmoodi and Mischa Dohler and Mahesh K. Marina. *"Group Communications in Narrowband-IoT: Architecture, Procedures, and Evaluation"*. IEEE Internet of Things Journal, vol. 5, no. 3, pp. 15391549, Jun. 2018.

- Galini Tsoukaneri and Mahesh K. Marina. *"On Device Grouping for Efficient Multicast Communications in Narrowband-IoT"*. In Proc. IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, Jul 2018.

- Galini Tsoukaneri and Yue Wang and Shangbin Wu. *"Probabilistic Preamble Selection with Reinforcement Learning for massive Machine Type Communication (MTC) devices"*. Accepted for publication at the 30th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, Sep 2019.

- Galini Tsoukaneri and Franscisco Garcia and Mahesh K. Marina. *"Narrowband-IoT Energy Consumption Characterization and Optimizations"*. Under submission at the 27th IEEE International Conference on Network Protocols (ICNP), Chicago, Illinois, Oct 2019.

Table of Contents

List of Figures

List of Tables

Acronyms

3GPP	3rd Generation Partnership Project
AF	Application Function
AKA	Authentication & Key Agreement
AMF	Access and Mobility Management function
APN	Access Point Name
AR	Augmented Reality
ARQ	Automatic Repeat Request
AS	Access Stratum
AUSF	Authentication Server Function
BCCH	Broadcast Control Channel
BCH	Broadcast Channel
Bps	Bits per second
BPSK	Binary Phase Shift Keying
BS	Base Station
C-DRX	Connected mode DRX
C-RNTI	Cell Radio Network Temporary Identifier
CBC	Cell Broadcast Centre
CCCH	Common Control Channel
CDMA	Code Division Multiple Access
CN	Core Network
CP	Control Plane
CP	Cyclic Prefix
CSI	Channel State Information
D2D	Device-to-Device
DCCH	Dedicated Control Channel
DCI	Downlink Control Information
DL	Downlink

DLSCH	Downlink Shared Channel
DoS	Denial of Service
DRB	Data Radio Bearer
DRX	Discontinuous Reception
DTCH	Dedicated Traffic Channel
eDRX	extended Discontinuous Reception
EIR	Equipment Identity Register
eMBMS	evolved Multimedia Broadcast Multicast Service
eNB	evolved Node B
EPC	Evolved Packet Core
ETWS	Earthquake and Tsunami Warning System
g-RNTI	group Radio Network Temporary Identifier
Gbps	Giga-bit per second
gn-eNB	next generation evolved Node B
gNB	next generation Node B
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobiles
HFN	Hyper Frame Number
HSPA	Evolved High Speed Packet Access
HSS	Home Subscriber Server
HTC	Human Type Communications
I-DRX	Idle mode DRX
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
IP	Internet Protocol
Kbps	Kilo-bits per second
KHz	kilo Hertz
KSI	Key Set Identifier
LDS	Low Density Spreading
LPWAN	Low Power Wide Area Network
LTE	Long Term Evolution
LTE-A	Long Term Evolution - Advanced
M2M	Machine-to-Machine

MAC	Medium Access Control
MBMS	Multimedia Broadcast Multicast Service
MBSFN	eMBMS over a Single Frequency Network
MC-CDMA	Multi-carrier Code Division Multiple Access
MCC	Mobile Country Code
MCCH	Multicast Control Channel
MCH	Multicast Channel
MCL	Maximum Coupling Loss
MCU	Micro-Controller Unit
MEC	Multi-Access Edge Computing
MIB	Master Information Block
MIB-NB	Narrowband Master Information Block
MIMO	Massive Input Massive Output
MME	Mobility Management Entity
MMS	Multimedia Messaging Service
MNC	Mobile Network Code
MTC	Machine-Type Communications
MTCH	Multicast Traffic Channel
NAS	Non Access Stratum
NB-IoT	NarrowBand Internet of Things
NCCE	Narrowband Control Channel Element
NCellID	Narrowband Cell Identity
NEF	Network Exposure Function
NFV	Network Function Virtualisation
NOMA	Non-Orthogonal Multiple Access
NORA	Non-Orthogonal Random Access
NPBCH	Narrowband Physical Broadcast Channel
NPDCCH	Narrowband Physical Downlink Control Channel
NPDSCH	Narrowband Physical Downlink Shared Channel
NPRACH	Narrowband Physical Random Access Channel
NPSS	Narrowband Primary Synchronisation Signal
NPUSCH	Narrowband Physical Uplink Shared Channel
NRF	Network Function Repository Function
NSSF	Network Slice Selection Function
NSSS	Narrowband Secondary Synchronisation Signal

OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
P-GW	Packet Data Network Gateway
PBCH	Physical Broadcast Channel
PCCH	Paging Control Channel
PCF	Policy Control Function
PCH	Paging Channel
PCFICH	Physical Control Format Indicator Channel
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDN	Packet Data Network
PDSCH	Physical Downlink Shared Channel
PDU	Packet Data Unit
PHICH	Physical Hybrid-ARQ Indicator Channel
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PMCH	Physical Multicast Channel
PO	Paging Occasion
PRACH	Physical Random Access Channel
PRB	Physical Resource Block
PSS	Primary Synchronisation Signal
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
QPSK	Quadrature Phase Shift Keying
RA	Random Access
RA-RNTI	Random Access - Radio Network Temporary Identifier
RACH	Random Access Channel
RAI	Release Assistance Indicator
RAN	Radio Access Network
RAR	Random Access Response
RB	Resource Block
RF	Radio Frame
RE	Resource Element
RL	Reinforcement Learning
RLC	Radio Link Control

RRC	Radio Resource Control
RS	Reference Signal
RU	Resource Unit
S-GW	Service Gateway
S-TMSI	SAE Temporary Mobile Subscriber Identifier
SAE	System Architecture Evolution
SARSA	State-Action-Reward-State-Action
SBA	Service Based Architecture
SC-FDMA	Single-Carrier Frequency-Division Multiple Access
SC-MCCH	Single Cell - Multicast Control Channel
SC-MRB	Single Cell - Multicast Radio Bearer
SC-MTCH	Single Cell - Multicast Traffic Channel
SC-PTM	Single-Cell Point to Multipoint
SCMA	Sparce Code Multiple Access
SDN	Software-Defined Networking
SDT	Small Data Transmission
SF	Subframe
SFN	System Frame Number
SIB	System Information Block
SIB-NB	Narrowband System Information Block
SIC	Successive Interference Cancellation
SIM	Subscriber Identity Module
SINR	Signal to Interference and Noise Ratio
SMF	Session Management Function
SMS	Short Message Service
SNR	Signal to Noise Ratio
SR	Service Request
SRB	Signalling Radio Bearer
SSS	Secondary Synchronisation Signal
TA	Tracking Area
TAC	Tracking Area Code
TAU	Tracking Area Update
TBS	Transport Block Size
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access

UCI	Uplink Control Information
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UICC	Universal Integrated Circuit Card
UL	Uplink
ULSCH	Uplink Shared Channel
UMTS	Universal Mobile Telecommunication System
UP	User Plane
UPF	User Plane Function
URLLC	Ultra Reliable Low Latency Communications
USIM	Universal Subscriber Identity Module
VR	Virtual Reality

Table 1: Acronyms

Lay Summary

Internet of Things (IoT) refers to the interconnection of devices embedded in everyday objects, that allow them to connect to the Internet and exchange data, without human intervention. Nowadays, IoT extends the network connectivity to devices beyond the traditional smartphones, computers, laptops or tablets, to any *unintelligent* device, such as kitchen appliances.

There exist several communication technologies to serve IoT devices, such as wired networks, WiFi, Bluetooth, SigFox and Zigbee. However, cellular networks offer some unique advantages, compared to competing technologies, that can enable the large range of IoT applications. These advantages include vast deployments, with more than 60% of the global population having access to the fourth generation of cellular networks (4G), extensive area coverage that allows them to reach devices in signal challenged locations, and flexible data rates to efficiently support different IoT devices.

Although cellular networks may seem an ideal candidate to support IoT devices, they were originally designed for *Human Type Communication (HTC)* devices, such as mobile phones, smartphones and tablets. Compared to HTC, IoT devices have significantly different communication patterns (e.g. frequency of data transmission, duration of connection with the network, length of transmitted data, dependence on battery). When IoT devices first appeared, their limited numbers did not pose a challenge for cellular networks. However, it quickly became apparent that the HTC-based design of cellular networks would struggle to accommodate their rapid growth.

While are several problematic points, the challenges that today's cellular network face to support the massive numbers of IoT devices can be grouped into three distinct categories: (a) establishing a connection with the network to exchange data, (b) efficiently using the network resources, and (c) decreasing the energy usage of IoT devices. With these challenges in mind, in this thesis, we propose several enhancements and algorithms to address the inefficiencies of current cellular networks that stem form the HTC-based design, and drive them into efficiently supporting massive numbers of

IoT devices.

Chapter 1

Introduction

Internet of Things (IoT) refers to the interconnection of computing devices embedded in everyday objects, that enable them to connect to the Internet and exchange data, without human intervention. Nowadays, IoT extends the network connectivity to devices beyond the traditional smartphones, computers, laptops or tablets, to any *unintelligent* device, such as kitchen appliances.

Current IoT devices can serve a wide variety of applications, such as monitoring, asset tracking, as well as smart homes, smart cities, industry automation and autonomous driving, and new applications are being developed every day. This variety of IoT applications has brought great freedom to users, and in recent years we have seen a tremendous increase in their numbers. Numerous devices are already deployed, and these numbers are only expected to grow in the near future, with forecasts reporting that ≈ 29 billion IoT devices will be operational by 2023 [1].

Although there exist several communication technologies to serve IoT devices, cellular networks offer some unique advantages that can enable the large range of IoT applications. Cellular networks are characterised by vast deployments, with more than 60% of the global population having access to 4G (also generally referred to as *Long Term Evolution (LTE)*)[1] cellular technology already, and $\approx 85\%$ are expected to be covered by 4G by 2023 [2]. Furthermore, cellular networks offer extensive area coverage compared to competing technologies (wired networks, WiFi [3], Bluetooth [4], Zigbee [5]), while at the same time supporting a large range of data speeds (up to 1 Gbps) and increased mobility. As such, cellular networks are considered excellent candidates when reach, mobility and ease of deployment are paramount.

While these advantages are certainly desirable for IoT applications, especially for

[1] For the reminder of this thesis we will use the terms 4G and LTE interchangeably.

applications that depend on high reliability and availability (e.g. vehicular communications, haptics and remote sensing [6, 7, 8]), existing cellular networks were not really designed to accommodate them. IoT devices function without the need of a human operator and the traffic patterns they generate are substantially different from the human-centric applications that were traditionally supported by cellular networks. This causes a series of inefficiencies [9, 10, 11], both at the network, as well as the device side, which makes existing cellular networks struggle as the number of IoT devices increases.

1.1 Differences Between IoT and Human-Centric Devices

IoT devices exhibit some key differentiating characteristics from traditional *Human Type Communication (HTC)* devices (mobile phones, tablets etc.), that are the predominant cellular network users. On the surface, IoT devices usually serve a singular purpose and, as such, they are typically much cheaper and have significantly lower computational capabilities. This combination of single-purpose and low cost allows them to be ubiquitous, and be embedded in a large number of devices that would traditionally not be connected to the network. This leads to the expectation that, eventually, each user will be owning a much larger number of IoT devices compared to HTC, and thus future cellular connection will be dominated by IoT devices. In fact, the rate of growth is so high, that IoT devices are expected to outnumber HTC devices as early as 2023 [12].

The differences however are not limited to their cost and numbers. From the network perspective, IoT devices are fundamentally different compared to traditional HTC devices. While HTC devices mainly receive large amounts of data from the network in long connections, such as web-browsing and video-streaming (traffic is mainly transmitted in the downlink direction), the typical use case of IoT devices is to monitor and report measurements to applications running on servers (traffic is mainly transmitted in the uplink direction) [13, 14]. Each such report can typically be summarised in a few bytes of information, and can be transmitted in comparatively very small messages [15]. As such, connections tend to be short, with only a small burst of activity at a time. Finally, while HTC connections are generally random and uniformly distributed in time [16], the reporting of IoT devices usually happens in fixed intervals, so their connections are predominantly periodic and synchronised (e.g. temperature measurements in a factory every hour) [16]. This periodic nature is an important characteristic,

as unlike pure randomness, periodicity is easy to model and plan around.

1.2 Key Challenges to Support IoT Devices in Cellular Networks

When IoT devices first appeared, cellular networks seemed to be an ideal candidate to support them, due to their vast deployments, extended coverage and flexible data rates. While their initially limited numbers did not pose a challenge for cellular networks, it quickly became apparent that the existing cellular networks design would struggle to accommodate their rapid growth.

The root of the problem lies in the very fact that cellular networks were historically dominated by HTC devices, and they were designed and optimised for these kinds of applications. The short, frequent and often synchronised transmission patterns of IoT devices can easily flood the traditional cellular networks, leading to delays and network outages [15]. Newer technologies, such as *NarrowBand IoT (NB-IoT)*, were introduced to better serve a subclass of IoT devices, but a large part of the network design is still carried over from earlier, HTC-focused generations of cellular networks, effectively inheriting many of their shortcomings. While the individual problematic points are numerous, we can group them into three distinct areas of focus:

1. **Connection establishment:** IoT devices usually spent most of their life being idle, with no active connections to the network, and they only wake up to transmit new data (e.g. temperature measurements). As they can be highly synchronised [16, 15], it is very often the case that numerous IoT devices try to establish a connection at the same time. In cellular networks, there are limited resources that can be used to request the establishment of a new connection, and such simultaneous attempts from multiple devices result to a large number of collisions among them. Collided devices are required to wait and attempt to establish a connection at a later time, until they do not collide, or until a maximum number of attempts has been reached. This phenomenon increases their network access delay, and can lead to network outages when the network is under increased load. This issue is not only problematic in theory, but has been shown to cause problems in existing 4G networks [17, 15], and it can in fact be used against them to launch *Denial of Service (DoS)* attacks, preventing all devices in a cell from accessing the network [15]. Furthermore, the connection establishment process

consists of a series of long procedures that include a large number of signalling messages that regulate the connection parameters. For HTC connections, which are usually long and carry large amounts of data, the signalling overhead for establishing a connection is minimal compared to the amount of data exchanged during the connection. However, for IoT devices, this signaling overhead is often disproportional to the size of the actual data being transmitted. Additionally, the fixed periodicity of IoT devices means that both the congestion to establish a connection and the signalling overhead are persistent.

2. **Network resource utilisation:** A key observation for IoT devices is that large numbers of them will be running the same software or executing the same tasks (e.g. multiple sensors in a factory running the same command or numerous similar-make devices receiving the same firmware update). As such, multicasting common content (e.g. firmware/software updates, task commands) to multiple devices in the same cell simultaneously, can reserve precious resources compared to device-specific transmissions, and improve the system performance. This can be especially beneficial in resource-limited technologies, such as NB-IoT. However, the existing frameworks for group communications require the periodic usage of resources to transmit service information, even when no services are availability or ongoing at the time. Furthermore, the current procedures that devices need to follow in order to receive multicast content are numerous, long and complicated, and are thus ill-suited for low-capability IoT devices.

3. **Device energy consumption:** Although both HTC and IoT devices are mainly battery operated, HTC devices can be recharged at will, and multiple times per day. However, this is not always true for IoT devices, which are usually expected to operate for long periods of time on a single battery charge. Their set-and-forget nature necessitates that executed procedures incur the minimum possible energy consumption. As the domination of HTC devices drove the design of cellular networks, energy efficiency was not considered the major concern. For example, the connection establishment process requires the transmission and reception of a large number of messages. As that number is significantly larger than the number of actual data messages, the energy consumed for the connection setup process incur a disproportional energy overhead to IoT devices. Other procedures keep devices connected to the network even when the devices do not have any more data to transmit, or require the devices to periodically wake up

from a low energy consumption state to check for network notifications. These procedures incur a disproportionate energy overhead to IoT devices, that far exceeds the energy needs of their actual application, and are thus ill-suited to support the unique requirements of IoT devices.

1.3 Thesis Contributions

The goal of this thesis is to address the various performance issues that hinder the wide-scale deployment of IoT devices over cellular networks. To this end, we examined all three challenging areas outlined above (connection establishment, network resource utilisation and energy efficiency) to identify the crucial roadblocks, and propose mechanisms to improve upon the existing designs, from both the network and the device perspective.

1.3.1 Connection Establishment

To address the challenge of the connection establishment, we identified three specific reasons that cause delays and network outages when large numbers of IoT devices try to establish a connection: increased number of collisions, signalling overhead in the radio interface and signalling overhead in the core network. We argue that this is due to the fact that previous network designs made the assumption that connections are random, infrequent and uniformly spread through time, and that devices stay connected for relatively long periods of time. Since IoT devices violate these assumptions, we propose a simple, secure and efficient mechanism called ASPIS that targets the traffic characteristics of IoT devices, and can address all three issues simultaneously without negatively affecting traditional HTC devices.

The major novelty of our mechanism is a new (intermediate) connection state that allows IoT devices to partly retain their previously established connection between subsequent transmissions, thus alleviating the signalling cost of establishing a new connection every time. Notably, this can be done efficiently from the network side as, unlike the connected state, it does not consume network resources to remain in this intermediate state. Additionally, our mechanism exploits the periodic nature of IoT devices to predict future collision spikes during the connection establishment process, and optimally allocate the available resources between IoT and HTC devices to minimise them.

Importantly, our mechanism does not require hardware changes, and can be implemented as a software update. Additionally, it can be incrementally deployed alongside legacy IoT and HTC devices and cellular network infrastructure. Moreover, our mechanism is able to enable these capabilities without compromising security, compared to previous competing approaches, as all messages are sent encrypted. We demonstrate the ease of implementation by developing a prototype implementation on a widely used open-source 4G software platform. We note that this contribution is targeting 4G networks and predates the final specifications for 5G. The newer 5G specifications actually adopt an inactive state which is very similar to our concept of an intermediate state.

Our ASPIS mechanism is able to address the three major deficiencies of the connection establishment process simultaneously. However, we believe that the increased number of collisions due to the synchronised nature of IoT devices is the major bottleneck for a timely connection establishment, and therefore, we also propose a novel mechanism that specifically focuses on reducing those collisions. Our mechanism employs recent advances on signal processing mechanisms that allow for the separation of simultaneously transmitted signals, in order to split the cell coverage area into different logical zones. Resources for network access are then re-used among the different zones without the fear of collisions, as these can be resolved at the receiver side. Then, as a second step, our mechanism uses Reinforcement Learning (RL) in order to create statistics about the usage of those resources in each zone, which are then used by the devices to select the resources which are less likely to result in a collision.

The novel use of Reinforcement Learning allows the network to capture dynamic changes in the cell, such as changes in the number of devices due to mobility, and update the network usage accordingly, resulting in significant decrease on the number of collisions, and thus faster network access. Similarly to ASPIS, our mechanism does not require hardware changes and can be implemented as a software update, without breaking compatibility with existing devices. Finally, it does not focus on a specific generation of cellular networks and can be implement in current 4G and future 5G networks identically.

1.3.2 Network Resource Utilisation

A key observation for IoT is that there will be large numbers of identical devices running the same software or performing the same actions. Whenever there is the

need to send the same data to large numbers of devices in a short period of time (e.g. firmware updates, task commands, etc.), the downlink channels will be flooded, potentially causing delays to the delivery of the multicast data and disruption to the normal traffic, especially in resource-limited technologies, such as NB-IoT. Therefore, efficient mechanisms for multicast communications are crucial to allow the simultaneous transmission of the same content to large groups of devices, with minimal network resource wastage.

Similarly to other procedures in cellular networks, the current framework for multicast transmissions was designed for HTC applications, such as live video streaming, and around the idea of dynamically subscribing to services. Therefore, although it is a good fit for HTC-centric applications, it is based on long and complicated procedures, and can lead to resource wastage, when used by IoT devices. The major drawback of this scheme is that the person operating the device is responsible for subscribing to the desired service when needed, and unsubscribe from it when the service is no longer wanted. As IoT devices operate autonomously and without human intervention, this subscription-based scheme is not well suited for them, as deciding which services the IoT devices should subscribe to can be very challenging. Essentially, it is mainly the device manufacturer, application provider or the device owner that decides which updates or commands should be delivered to which devices, and the different multicast groups may be created on the fly.

Furthermore, the current multicast framework pre-allocates resources for the periodic transmission of control information for the multicast services (i.e. list of the offered services, whether ongoing, imminent, pending or inactive, as well as configuration information for the transmissions). For HTC-centric services, this pre-allocation may not incur a significant cost as it usually occurs when there is an ongoing or imminent service (e.g. streaming of a football match), and it can seize when the service is terminated at the provider side. Furthermore, it is the person's responsibility to check if the service is available and subscribe or unsubscribe from it. However, especially, in the cases of firmware updates or task commands for IoT devices, the provided services need to be available all the time, as new devices need to be able to subscribe to them at any given time, and there is no indication as to when the next session will start. This constant availability requires constant use of precious network resources for the transmission the service information.

To make matters worse, for IoT multicast services we can assume that devices subscribe to at least one service after their first power on (e.g., to receive updates from

the vendor). Hence, the number of concurrently available services should be at least equal to the number of types/makes/models of the devices present in the cell as even different models of the same manufacturer may require different updates. According to the current design of 4G networks, the maximum number of concurrent services is 64 [8-group communications]. However, due to the variability of the devices and the applications they run, the number of different services will be much larger. This results in severe degradation of the systems performance as precious resources are wasted for control information of pending multicast services that might not be used for long periods of time. This degradation is even more evident in resource-limited technologies such as NB-IoT. Furthermore, devices that have ongoing subscriptions to any service need to periodically monitor the network for information regarding their services. The subscription and monitoring is done on a per-service basis thus increasing the energy consumption of the devices. Therefore, we argue that the current multicasting framework used in cellular networks is not well suited for IoT devices and applications.

To address the aforementioned issues, we propose novel enhancements to the existing multicast framework that forego the subscription-based model currently used in cellular networks, in favour of an on-demand scheme. Our enhancements greatly simplify the currently used scheme, and consist of fewer and shorter procedures, which are tailored to the demands of IoT devices. As part of the process simplification, we shift the responsibility of service selection to the device owner/manufacturer or application provider to produce the list of devices that need to receive the multicast context. The network is then responsible to proactively notify the listed devices in time, in order to receive the multicast data.

Furthermore, our enhancements reduce the resource wastage for the transmission of control information for any non-ongoing services with two transmission mechanisms that provide different prioritisation between multicast data and background traffic. Our first mechanism prioritises unicast traffic over multicast and is mainly suitable for delay-tolerant transmissions, such as an application update, while our second mechanism prioritises the multicast traffic in order to complete the transmission as soon as possible, and is better suited to crucial and urgent transmissions, such as security updates.

Finally, efficiently notifying the devices individually without wasting network resources is a paramount, but non-trivial problem. Therefore, we also present three different mechanisms for synchronisation and notification of IoT devices, with different trade-offs in network resource usage, energy consumption and compliance to existing

3GPP standards for 4G networks.

1.3.3 Device Energy Efficiency

One of the major goals of IoT devices is to prolong the battery life of devices, with current technologies targeting a life expectancy of at least 10 years on a single battery charge. As such, it is important that any operations that IoT devices execute are designed to incur the minimum possible energy consumption. However, cellular networks were not initially designed with the energy consumption at their core, as HTC devices can be easily re-charged. Recently, new cellular network technologies such as NB-IoT were designed to focus on the unique demands of IoT devices, however they inherit the physical and logical design of 4G and 5G networks to a great extent, to preserve interoperability and backwards compatibility. Therefore, the suitability of the inherited frameworks for the NB-IoT context is unclear, as IoT devices may have severely limited computational resources, potentially making long and energy expensive algorithms impractical. Furthermore, as NB-IoT devices typically send small amounts of data in frequent, periodic transmissions, the existing communication procedures can incur a disproportionate energy overhead in relation to actual data communicated, that needs to be carefully examined.

As our third contribution, we focus on the energy efficiency from the perspective of the IoT device and attempt to determine the operations with the highest energy consumption, in order to identify the areas that call for optimisation. As NB-IoT devices have the strictest energy requirements of more than 10 years battery life on a single charge, and overall cost less than $5 [18], we specifically focus our attention to them. It is noteworthy however, that a large part of our energy consumption analysis is common for other types of IoT devices, and thus, some of the results should generalise.

Firstly, we perform a thorough experimental measurement of the power consumption of the individual operation that a NB-IoT device performs under normal use, using three different commercial NB-IoT devices. These operation-specific measurements offer significant insights on the distribution of the energy consumption of IoT devices over the different procedures. As such, they allow us to unearth potentially inefficient areas of the NB-IoT protocols, and can guide us towards effective optimisations. Our results also show a large deviation from the energy consumption assumptions published by 3GPP [19] as well as other works [20], which can greatly affect studies that rely on them (e.g., [21]). *To the best of our knowledge, this is the first work for NB-IoT*

devices that measures each operation in isolation.

Building on the above energy characterisation, we present an NB-IoT energy consumption model, which we use to simulate the battery life of a device under realistic traffic conditions. This gives us an estimate of the battery capacity requirements to meet the 10 year goal, and we find that it is far from realisable with current practices and previously proposed approaches, given the $5 cost constraint. We also use our model to compare against previous works that do not take all operations into account, and assess the overall energy consumption difference. We show that operations ignored by prior works (e.g., [20]) can have a significant energy cost, and it is imperative they are considered and optimised to lower the device energy consumption.

Finally, we propose novel mechanisms that exploit the distinct characteristics of NB-IoT devices, to substantially reduce their energy consumption. Furthermore, we discuss a set of best practices under the existing protocols, that device vendors and network operators should consider in order to maximise battery life.

1.4 Thesis Organisation

The reminder of this thesis is organised as follows:

Chapter 2 provides a detailed description of the background of cellular networks. We begin with the current 4G technology, and we also discuss the future 5G networks, highlighting the differences of the two generations. We present both the physical and logical the architecture of cellular networks, as well as the different procedures that devices follow to establish a connection, exchange data and terminate their connection.

In this chapter we also present the NB-IoT technology that is expected to be the dominant supporting technology for IoT devices in the future. We highlight the differences and new features compared to the widely deployed 4G networks, as well as the upcoming 5G networks. Parts of this chapter have been published in IEEE Communications Magazine, in January 2018 [22].

Chapter 3 provides an overview of the related literature, and the evolution of research for the three key challenge areas throughout the years. In this chapter, we also discuss the limitations and shortcomings of the previous works, that necessitated the need for our novel solutions.

Chapter 4 presents our ASPIS mechanism for reducing the collisions and the signalling overhead of IoT devices on 4G networks. This work has been published in the 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems

(MASS 2017) [23].

Chapter 5 focuses specifically on the collisions during the connection establishment process, and presents our novel mechanism that used RL to increase the number of successful connection attempts. This work has been performed in the framework of the Horizon 2020 project ONE5G (ICT-760809) receiving funds from the European Union, and has been accepted for publication at the 30th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) 2019.

Chapter 6 presents our work on network resource utilisation for group communications over NB-IoT, along with our work on efficient grouping and synchronisation of IoT devices. The works presented in this chapter have been published in the IEEE Internet of Things Journal, in June 2018 [24], and in the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS 2018) [25].

Chapter 7 presents our work on NB-IoT device energy performance and efficiency of the different operations that devices have to follow during their lifetime. The work presented in this chapter is under submission at the 27th IEEE International Conference on Network Protocols (ICNP) 2019.

Finally, **Chapter 8** presents our concluding remarks and summarises the work presented in this thesis. In this chapter we also discuss limitations of the presented work and future research directions.

Chapter 2

Background

This chapter provides a detailed description of the cellular network technologies, that are studied in the context of this thesis. We begin with the 4G and 5G technologies and present their physical and logical architecture, and highlighting their similarities and differences (section 2.1.1). We then present their physical layer design (frame structure, channels, signals) that is fundamental to cellular networks (section 2.1.2). We proceed to the communication procedures that are required to establish a connection, exchange data and then terminate the connection with the cellular network (section 2.1.3). Finally, we describe the existing group communications framework for multicast data, as well as the security framework that is used for all communication.

During the 4G era it became apparent that the HTC-based design of cellular networks could not adequately support the massive numbers of IoT devices, due to their unique communication patterns. To meet with the increasing requirements, 3GPP begun designing new cellular networks technologies, to better accommodate IoT devices. As *Narrowband-IoT (NB-IoT)* is one of the most well known of those technologies, we also present it here, highlight its differences with the 4G and 5G networks.

2.1 4G and 5G Cellular Technologies

2.1.1 Network Architecture

The architecture of cellular networks is split into the physical and logical architecture. The physical architecture includes the different physical entities, while the logical architecture describes the network protocols used for communication, and the different functions of the physical entities. As each generation brings new and enhanced func-

tionality, the architecture (either logical or physical) may change. In this thesis we focus on the 4th and 5th generation of cellular networks, and present their respective architecture, highlighting their major differences.

2.1.1.1 Physical Architecture

The physical architecture of the cellular networks is comprised of three main components:

1. the *User Equipment (UE)*,
2. the *Radio Access Network (RAN)*, and
3. the *Core Network (CN)*

2.1.1.1.1 User Equipment (UE): A *User Equipment (UE)* (figure 2.1) is any device, either HTC or IoT, that is able to generate data and initiate connections with the network. For the remainder of the thesis we will use the terms UE and device interchangeably. To communicate with the cellular network, each UE must be equipped with a *Universal Integrated Circuit Card (UICC)*, which contains the *Universal Subscriber Identity Module (USIM)*. The USIM is responsible for storing user-specific information, such as the identity of the serving network, and the major security keys for the provided services. The UICC also contains two important identifiers, the *International Mobile Equipment Identity (IMEI)* and the *International Mobile Subscriber Identity (IMSI)*. The IMEI uniquely identifies the physical device, and can thus be used to prevent stolen devices from accessing the network, while the IMSI identifies the network subscription and the services that can be provided.

2.1.1.1.2 Radio Access Network (RAN): The *Radio Access Network (RAN)* is the component of the cellular network that provides network access to the UEs over the air interface. In 4G networks, it is called *Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)*, while in 5G, it is called *New Generation RAN (NG-RAN)*. Its main element is the base station (BS), known as the *evolved NodeB (eNB)* in 4G networks, or the *next generation NodeB (gNB or ng-eNBs)* in 5G. For the remainder of this thesis we will be referring eNBs and gNBs as BSs if the discussed topic applies to both cellular network generations. Otherwise, we will use the appropriate terminology according to the generation that the discussed topic applies to.

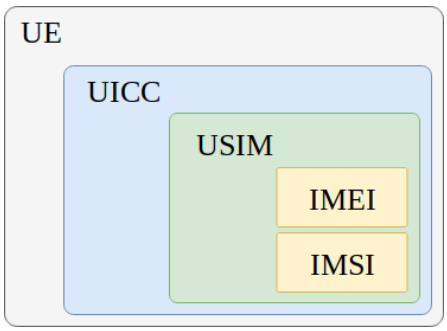


Figure 2.1: **User Equipment (UE) architecture:** The figure depicts the architecture of the User Equipment (UE) that contains the *Universal Integrated Circuit Card (UICC)*, which, in turn, contains the IMEI and the IMSI.

The BS controls the data transmission and reception of the UEs, as well as the low-level procedures, such as handovers to neighbouring BSs. In 4G networks, neighbouring eNBs are connected through the X2 interface, while each eNB connects to the core network through an S1 interface [26]. In 5G, gNBs are connected using an Xn interface, while the connection to the 5G core network is accommodated through the NG interface [27]. The two architectures and the interfaces used are depicted in figure 2.2.

2.1.1.1.3 Core Network (CN): The core of the cellular network is a packet switching backhaul domain that provides converged voice and data connections to UEs through the RAN. The core network in 4G is referred to as the *Evolved Packet Core (EPC)*, and it is significantly different from the *New Generation 5G Core (NG-core)* of 5G. We now describe both core network architectures.

2.1.1.1.3.1 4G architecture: In 4G networks, the main physical entities of the EPC (figure 2.3) include:

- the *Packet Data Network Gateway (P-GW)*,
- the *Serving Gateway (S-GW)*,
- the *Mobility Management Entity (MME)*, and
- the *Home Subscriber Server (HSS)*,

The *Packet Data Network Gateway (P-GW)* acts as the interface between the 4G core network and any other packet data network, such as the Internet, and each UE is assigned to a default P-GW the moment it establishes a connection

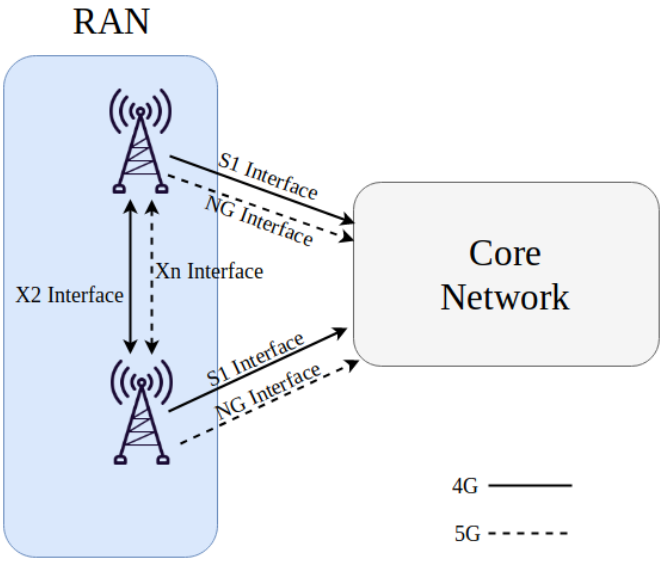


Figure 2.2: **Radio Access Network (RAN) architecture in 4G and 5G networks:** The figure depicts the architecture of the Radio Access Network (RAN) for both 4G and 5G networks, and the respective connection interfaces.

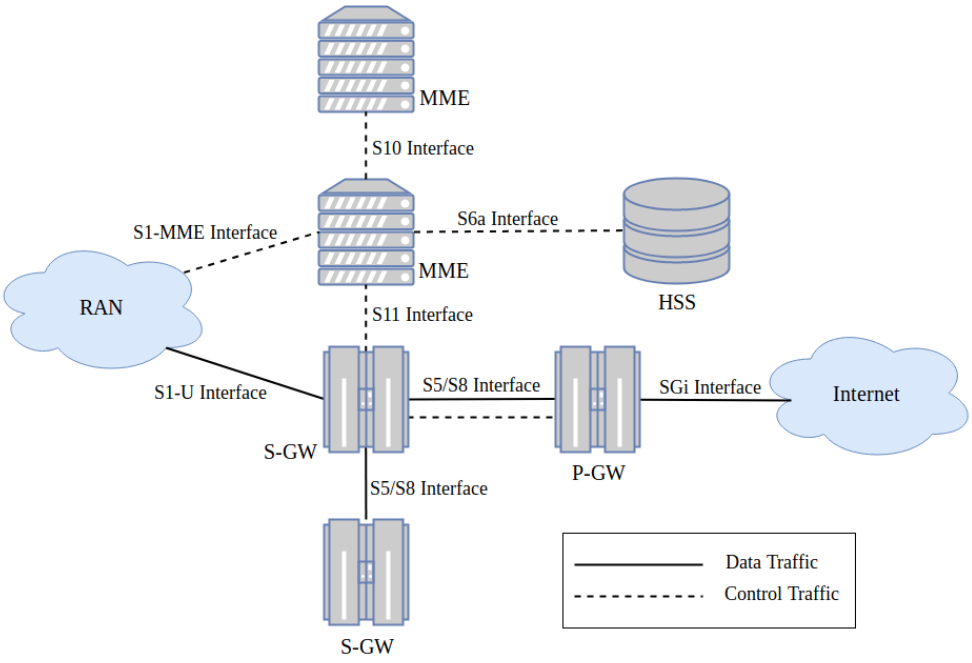


Figure 2.3: **Evolved Packet Core (EPC) architecture in 4G networks:** The figure depicts the architecture of the EPC in 4G networks, that includes the P-GW, S-GW, MME and HSS. These entities co-operativelly allow authentication and authorisation of the subscribers, and access to external networks.

The *Serving Gateway (S-GW)* is responsible for routing and forwarding the UE's data packets between the eNB and the P-GW. Usually, a 4G core network includes a number of S-GWs, each of which serves all the UEs within a certain geographical area.

The *Mobility Management Entity (MME)* manages the user authentication, authorisation as well as the key management functions, and facilitates the UE's connectivity to external networks. For connected UEs, the MME is responsible for managing their entire connection, and handovers between neighbouring eNBs. When the UE does not have an active connection with the network, the MME is responsible for keeping track of its location, and paging it when data from the network need to be delivered to it. The MME is also responsible for allocating the default S-GW and P-GW upon the UE's initial connection to the network. Similarly to the S-GW, each UE is assigned to a *serving* MME according to its geographical location, and can be re-assigned to a new one, if it moves out of the coverage area of the serving MME.

The *Home Subscriber Server (HSS)* is the central database of the 4G networks that stores the subscriber identifiers, their security information, and any other information relevant to the services that the subscriber is allowed to receive.

Throughout the cellular network, virtual end-to-end connections, called *communication bearers* are used to transmit UE data, among the different entities (figure 2.4). Each bearer has different Quality of Service (QoS) levels, and a UE can have a number of different bearers, depending on the number of services it receives, and the QoS requirements of each of them. However, every UE has at least one default *Radio Access Bearer (RAB)* between itself and the eNB, and one *Evolved Packet System (EPS)* bearer between itself and the core network. Both of these consist of the *Signalling Radio Bearer (SRB)* used to transfer control messages, and the *Data Radio Bearer (DRB)* that is used to transfer the UE's data.

2.1.1.1.3.2 5G architecture: Unlike the entity-based architecture of 4G networks, where each physical entity is solely responsible for different functions, the 5G architecture is a *Service-Based Architecture (SBA)*, where network functions will be strongly decoupled and abstracted from the physical entities (figure 2.5). In 5G networks, physical components are not independently responsible for the different functions (e.g. data packet forwarding), but instead, the different functions will be split into slices, each containing several services for the same function, allowing the subscriber to combine the specific functions and services that better suit his needs.

The main functions provided in a 5G network are:

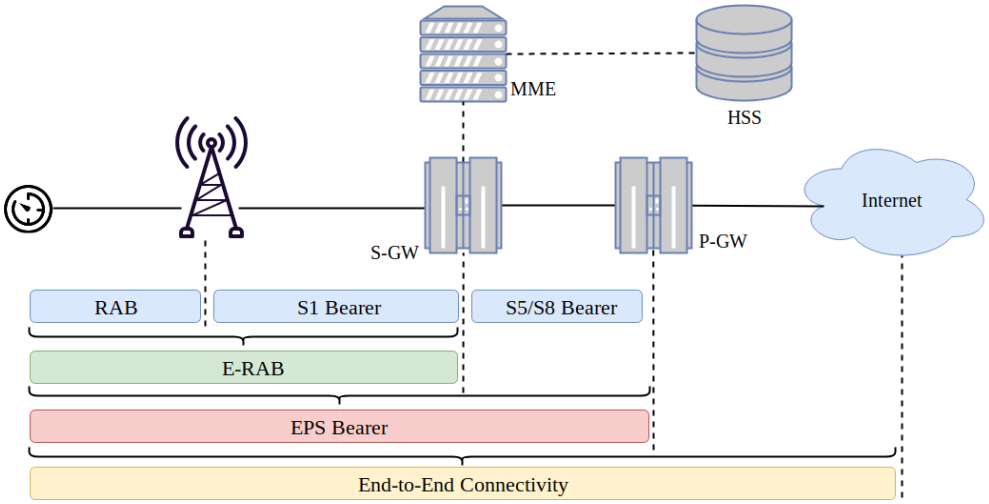


Figure 2.4: **Architecture of communication bearers in 4G networks:** The figure depicts the different communication bearers in a 4G network and their corresponding range.

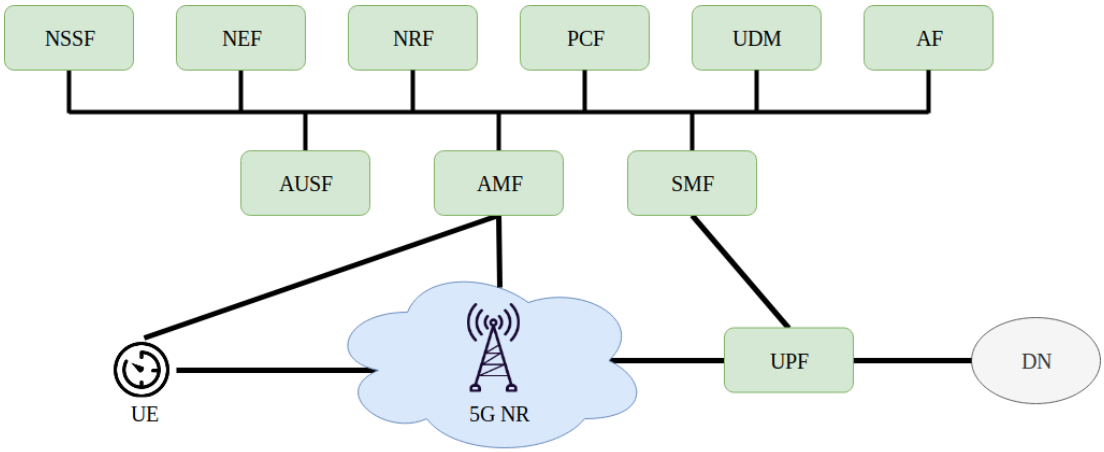


Figure 2.5: **Service-based architecture (SBA) in 5G networks:** The figure depicts the service-based architecture of future 5G networks with most of the provided functions.

- the *Access and Mobility Management Function (AMF)*,
- the *Session Management Function (SMF)*,
- the *User Plane Function (UPF)*,
- the *Policy Control Function (PCF)*,
- the *Authentication Server Function (AUSF)*,
- the *Unified Data Management (UDM)*,
- the *Network Exposure Function (NEF)*,
- the *Application Function (AF)*,
- the *Network Functions Repository Function (NRF)*, and
- the *Network Slice Selection Function (NSSF)*

The *Access and Mobility Management function (AMF)* will provide security services (e.g. ciphering), registration and connection management, as well as access authentication and authorisation.

The *Session Management function (SMF)* will be responsible for session management, including session establishment, IP address allocation to UEs, session management, notification of pending network-originated data and session release.

The *User Plane function (UPF)* will be responsible for packet routing and forwarding, QoS handling, and will act as the interconnection point to external data networks, such as the Internet.

The *Policy Control function (PCF)* will handle policy frameworks and will be providing policy rules to other functions, as well as subscription information for policy decisions.

The *Authentication Server function (AUSF)* will assume some of the responsibilities of the HSS of 4G networks and will be responsible for subscriber authentication.

The *Unified Data Management (UDM)* will assume the remaining responsibilities of the HSS that include storage of authentication credentials for subscribers. It will be responsible for user identification, access authorisation and subscription management.

The *Network Exposure function (NEF)* will be responsible for the secure exposure of the network's services and capabilities to external networks, as well as translating information about the capabilities of external networks.

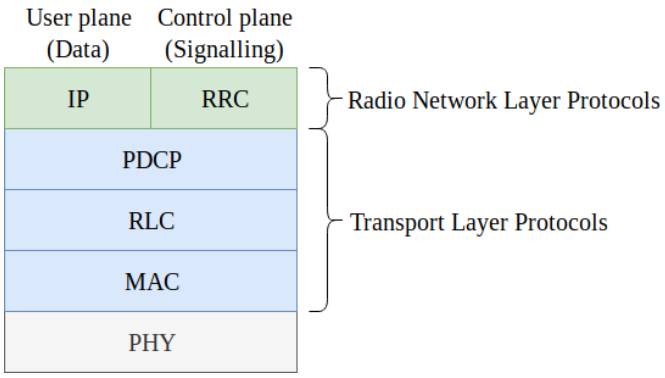


Figure 2.6: **Protocol stack of cellular networks:** The figure depicts the protocol stack used in cellular networks. The protocols are split into radio network and transport layer protocols. The radio network layer is then split into user and control plane.

The *Application function (AF)* will co-operate with the NEF to provide decisions on the traffic routing of application data.

The *Network Functions Repository function (NRF)* will support service discovery, and will maintain the available instances of other network functions.

Finally, the *Network Slice Selection function (NSSF)* will assist subscribers in selecting the appropriate network slice that better serves their needs.

In contrast to the changes in the core network, the physical architecture of the radio access network is the same in 5G networks with the previous generation.

2.1.1.2 Logical Architecture

The logical architecture of cellular networks includes the the protocol stack (figure 2.6), which is split into the *transport* and *radio network* layer.

2.1.1.2.1 Radio Network Layer The radio network layer is further split into *User Plane* and *Control Plane*. The user plane handles data that originate from, or is destined to the applications running on the UE, using the *Internet Protocol (IP)* for routing data packets from the UE to the P-GW. The control plane handles signalling messages that are used by the network to coordinate the connection and data transmission. Its *Radio Resource Control (RRC)* protocol is used by the BS to control all radio communications of the UE for connection establishment, management and connection release, as well as system information broadcasting.

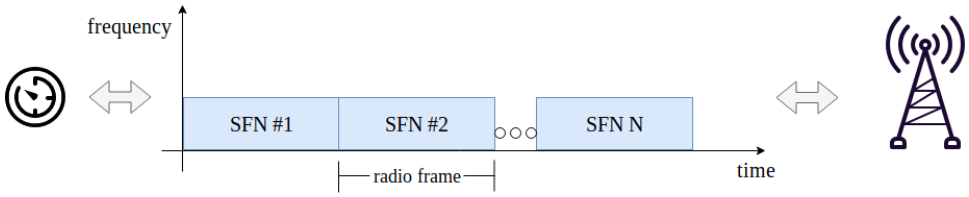


Figure 2.7: **Radio frame transmission in frequency and time:** The figure depicts the transmission of frames in the frequency and time domain, with their system frame numbers (SFNs).

2.1.1.2.2 Transport Layer The transport layer is responsible for transmitting messages from one point to another. It includes the *Medium Access Control (MAC)* protocol, which handles low-level control of the physical layer, the *Radio Link Control (RLC)* protocol which maintains the data link between the UE and the BS, and the *Packet Data Convergence Protocol (PDCP)* that is responsible for the higher level transport functions such as header compression, transfer of control and user plane data, ciphering and integrity protection.

2.1.2 Physical Layer Design

In the physical layer of the air interface, cellular networks use *Orthogonal Frequency-Division Mutliple Access (OFDMA)* in the downlink (DL), and a pre-coded version of OFDM, called *Single-Carrier Frequency-Division Multiple Access (SC-FDMA)* on the uplink (UL). Data is transmitted on signals and channels over frames, which we now describe in detail.

2.1.2.1 Frame Structure

User and control data is transmitted in the frequency and time domain using radio frames (RFs) (figure 2.7). Each radio frame is numbered with a *System Frame Number (SFN)* and a *Hyper Frame Number (HFN)*. The SFN ranges from 0 to 1023, and repeats from the start once the SFN counter reaches its maximum value. The HFN also ranges between 0 and 1023, but it increases by 1 each time the SFN counter is reset. Each radio frame is 10ms long, and 1024 frames form a hyper frame lasting $\approx$ 3 hours. There are two different types of frames, however, for the purposes of this thesis it is sufficient to describe the frame type 1, as it is the most commonly used one.

On the frequency domain, each frame is split into *Resource Blocks (RBs)*, each of which is 180 KHz in frequency. The total number of RBs in a frame depends on the

System Bandwidth (MHz)	Number of Resource Blocks
1.4	6
3	15
5	25
10	50
15	75
20	100

Table 2.1: **Number of resource blocks in 4G for different system bandwidths:** The table shows the number of resource blocks for the different system bandwidths.

system bandwidth. A 4G network can be deployed on six frequency bands and the resulting system bandwidths are 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz and 20 MHz. The number of available RBs for each bandwidth is shown in table 2.1.

Each RB is further split into subcarriers that are separated from each other using a predefined subcarrier spacing. 4G allows two subcarrier spacing of 15 KHz and 7.5 KHz, that result in an RB with 12 and 24 subcarriers respectively. The most common configuration of 12 subcarriers with 15 KHz of spacing is depicted in figure 2.8.

In the time domain, each frame is 10 ms long, and is split into 10 subframes (SFs) of 1 ms duration each. Subframes are further split into 2 slots of 0.5 ms each, which are in turn split into *Orthogonal Frequency-Division Mutliplexing (OFMD)* symbols. Each symbol is always preceded by a *cyclic prefix (CP)*, which is a guarding interval used to reduce inter-symbol interference, and prevent symbol overlapping when the transmission of the preceding symbol is delayed. The CP can be either normal or extended. When the normal CP is used, each slot consists of 7 OFDM symbols, while the extended CP results in a slot with 6 OFDM symbols. The 4G frame is depicted in figure 2.9. It is important to note that the same CP must be applied on every RB of the same subframe. Given all the above, the smallest structure in the time and frequency domains is 1 symbol long and 1 subcarrier wide, and it is called a *Resource Element (RE)*.

In 5G networks, the frame structure will be significantly different. To begin with, 5G networks will be operating in different frequency bands, and the system bandwidths will range from 5MHz to 100MHz. Furthermore, 5G will support several subcarrier spacings and slot lengths (in contrast to the fixed values currently used in 4G), resulting

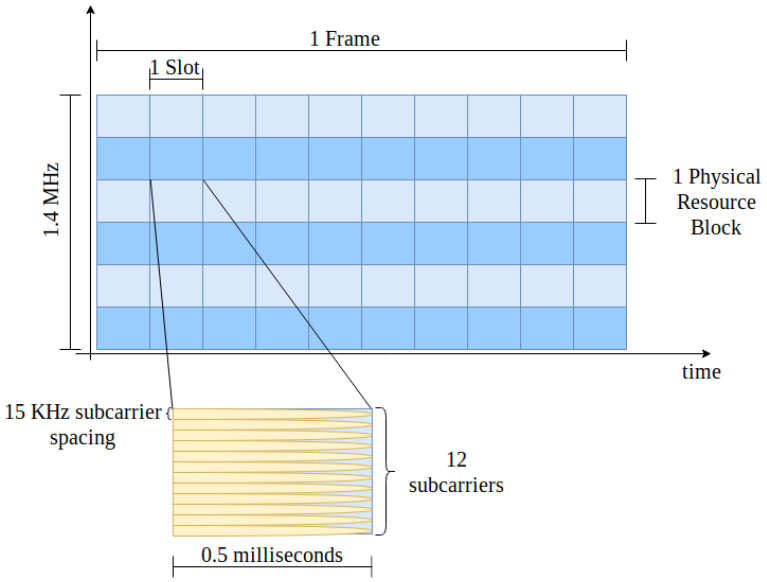


Figure 2.8: **Resource block with subcarriers in 4G networks:** The figure depicts a resource block in a 1.4 MHz system, with 12 subcarriers and 15 KHz subcarrier spacing.

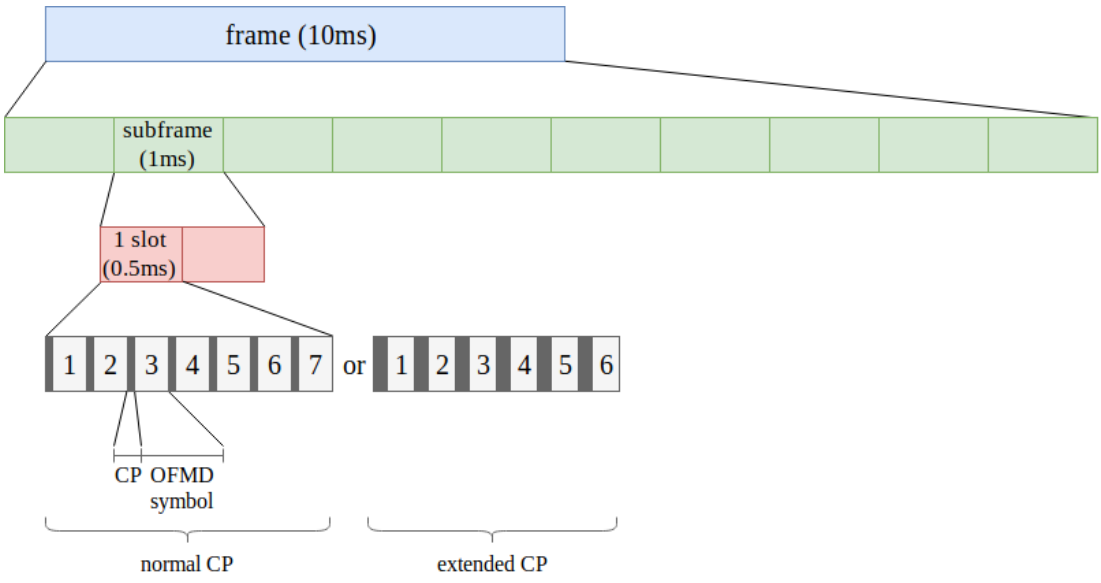


Figure 2.9: **Structure of an 4G frame:** The figure depicts the complete structure of an 4G frame. Each frame is split into 10 subframes, and each subframe is further split into 2 slots. Depending on the cyclic prefix used, each slot is then split into 6 or 7 OFDM symbols.

Subcarrier Spacing (KHz)	Number of slots / frame	Slot Length (ms)
15	10	1
30	20	0.5
60	40	0.25
120	80	0.125
240	160	0.0625

Table 2.2: **Subcarrier spacing and resulting slots in 5G networks:** The table shows the different subcarrier spacing that will be supported in 5G networks, and the resulting number of slots and slot lengths for each of them.

in flexible frame structures. Specifically, 5G will allow subcarrier spacings of 15KHz, 30KHz, 60KHz, 120KHz and 240KHz. The resulting number of slots per frame and slot lengths for each subcarrier spacing is shown in table 2.2.

2.1.2.2 Signals

Cellular networks utilise a number of different signals in order to allow UEs to discover, synchronise with and connect to a BS. These signals also carry vital information regarding the configuration of the cell to help UEs estimate the channel conditions and adapt their transmissions accordingly. The major signals are:

- the *Primary Synchronisation Signal (PSS)*,
- the *Secondary Synchronisation Signal (SSS)*,
- the *Reference Signal (RS)*,
- the *Master Information Block (MIB)*, and
- the *System Information Blocks (SIBs)*

The *Primary Synchronisation Signal (PSS)* is a physical layer signal that depends on the frequency band used, and it is used by the UEs for radio frame synchronisation in the downlink. It is transmitted on the last OFDM symbol of slots #0 and #5, and provides the physical layer identity of the cell.

The *Secondary Synchronisation Signal (SSS)* is also a physical layer signal for downlink frame synchronisation which is transmitted on the symbol before the PSS,

and also depends on the frequency band used. Using the PSS and SSS the UE can find the start and end of the frame, and thus totally synchronise with the BS at radio frame level in the time domain. The SSS also provides the physical layer cell identity group number. Using the physical layer identity of the PSS and the physical layer cell identity group number the UE can determine the *Physical Cell Identity (PCI)* of the cell based on:

$$PCI = 3 * physical\_layer\_cell\_identity\_group + physical\_layer\_identity \quad (2.1)$$

The *Reference Signals (RS)* are transmitted in various resource elements of different subframes, and are used for channel estimation, cell selection/reselection and handovers to neighbouring BSs.

The *Master Information Block (MIB)* contains information about the system bandwidth used, the SFN, as well as configuration information about the *Physical HybridARQ Indicator Channel (PHICH)* (section 2.1.2.3). The MIB is transmitted periodically, and spans over 4 frames, resulting in a total duration of 40 ms.

Finally, the *System Information Blocks (SIBs)* carry cell-specific information, such as the *Mobile Country Code (MCC)* and the *Mobile Network Code (MNC)* (which make up the *Public Land Mobile Network (PLMN)* identity of the network), and the *Tracking Area Code (TAC)* of the cell. This information is essential for the UE in order to select the appropriate cell and configure its transmission accordingly.

2.1.2.3 Channels

User and control data is transmitted over the air interface using a number of channels. Different channels are defined for the uplink and downlink direction, and on each direction they are split into logical, transport and physical layer channels. The mapping of the different uplink channels in the three layes is depicted in figure 2.10.

2.1.2.3.1 Uplink

- Logical
  - the *Common Control Channel (CCCH)* is used for random access information.
  - the *Dedicated Traffic Channel (DTCH)* is used for the transmission of user data.

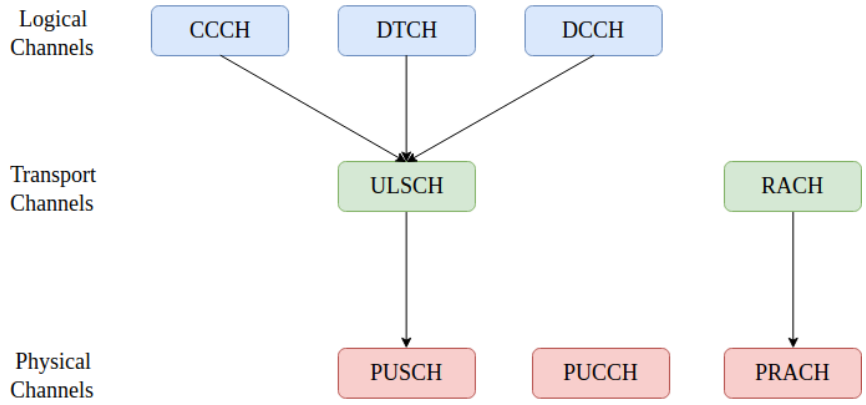


Figure 2.10: **Mapping of uplink channels:** The figure depicts the mapping of the different logical channels to the transport channels, and then to the physical channels for transmission in the uplink direction.

  - the *Dedicated Control Channel (DCCH)* carries UE-specific control information, for actions such as power control and handovers.

- Transport
  - the *Uplink Shared Channel (ULSCH)* is used to transmit UE-originated data.
  - the *Random Access Channel (RACH)* is used for the Random Access process (section 2.1.3.2).

- Physical
  - Control
    1. the *Physical Uplink Control Channel (PUCCH)* carries control information, such as the *Uplink Control Information (UCI)*.
  - Data
    1. the *Physical Uplink Shared Channel (PUSCH)* carries UE-originated data.
    2. the *Physical Random Access Channel (PRACH)* is used for the Random Access process (section 2.1.3.2), and is the only channel that can be used by non-synchronised UEs.

2.1.2.3.2 Downlink In the downlink direction the following channels are used:

- Logical
  - the *Paging Control Channel (PCCH)* transfers paging information and system change notifications.
  - the *Broadcast Control Channel (BCCH)* is used for broadcasting system information.
  - the *Common Control Channel (CCCH)* is used to transmit control information from the BS to the UE.
  - the *Dedicated Traffic Channel (DTCH)* is a point-to-point channel that carries UE-specific data.
  - the *Dedicated Control Channel (DCCH)* is a point-to-point channel that carries UE-specific control information.
- Transport
  - the *Paging Channel (PCH)* is used to convey the PCCH in the transport layer.
  - the *Broadcast Channel (BCH)* is used to convey the BCCH in the transport layer.
  - the *Downlink Shared Channel (DLSCH)* is the main channel used for downlink user data, and conveys a number of logical channels.
- Physical
  - Control
    1. the *Physical Broadcast Channel (PBCH)* carries the MIB, which is used by the UEs to acquire the initial configuration of the cell.
    2. the *Physical Downlink Control Channel (PDCCH)* is used to convey control signalling messages on the physical layer, by means of the *Downlink Control Information (DCI)*.
    3. the *Physical HybridARQ Indication Channel (PHICH)* carries the HARQ ACK/NACK signal indicating whether previously transmitted data has been correctly received.
    4. the *Physical Control Format Indicator Channel (PCFICH)* informs the UE about the format of the signal being received, and indicates the number of OFDM symbols used in the PDCCH.

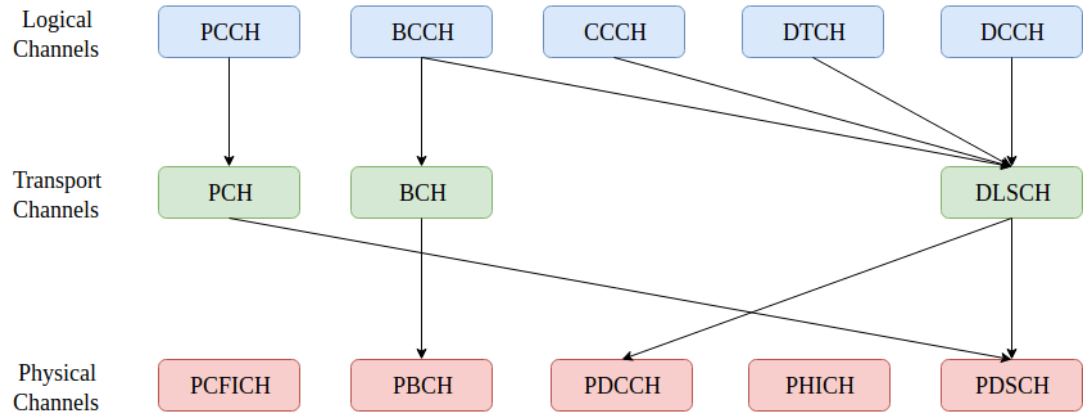


Figure 2.11: **Mapping of downlink channels:** The figure depicts the mapping of the different logical channels to the transport channels, and then to the physical channels for transmission in the downlink direction.

- Data
  1. the *Physical Downlink Share Channel (PDSCH)* is the physical channel that carries user data.

The mapping of the different downlink channels is depicted in figure 2.11.

2.1.3 Communication Procedures

In this section, we describe the different processes that a UE follows in order to connect to the network and transmit data. We begin by presenting the different RRC states that the device can be in, and we then describe the different procedures that switch a UE from one state to the other.

2.1.3.1 Radio Resource Control States

The state of the device indicates it connection status from the point of view of the BS. In 4G networks, a UE can be in one of two RRC states: *RRC Idle* or *RRC Connected*. In 5G networks, a new *RRC Inactive* state has been introduced, to reduce the number of messages needed to switch from one state to the other, and therefore the overall connection delay. The state diagram of 4G networks is depicted in figure 2.12, while the state diagram of 5G networks is shown in figure 2.13.

RRC Idle: The RRC Idle state is a low-energy state that the UEs switch to when they have no data to exchange with the network. When a UE is in the RRC Idle state it has no established/active connection with the network, and all RAN bearers, as well as

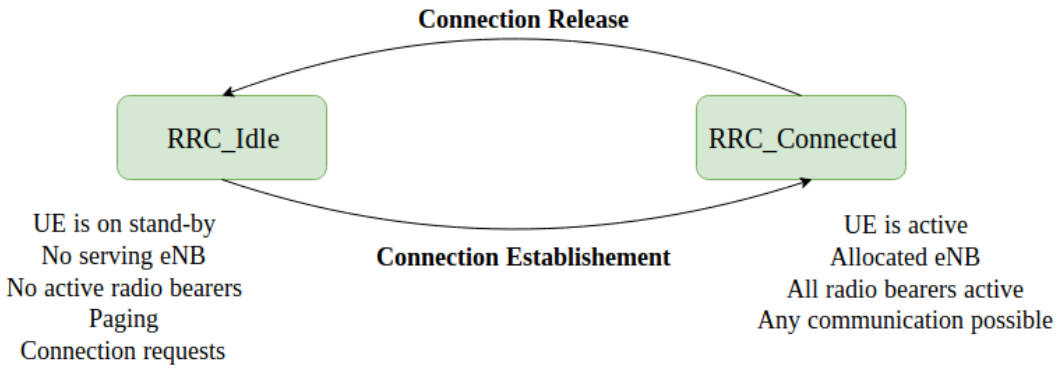


Figure 2.12: **RRC state diagram in 4G networks:** The figure depicts the RRC state diagram, and the procedures required to transition from one state to the other. The major features of each state are also mentioned.

most of the EPC bearers are torn down. From the UE's perspective there is no serving BS, and the device is only able to receive paging messages for network-originated data, or send connection requests to initiate new connections.

RRC Connected: The RRC Connected state is the state that a device is in when it has an active connection with the BS and is able to exchange data. All core network entities also have an active connection with the UE, and any communication is possible. When the UE has no more data to exchange it can be switched back to the RRC Idle state using the release connection procedure.

RRC Inactive: The RRC Inactive is similar to the idle state in terms of energy consumption (figure 2.13). However, in contrast to the idle state, the connection of a device is suspended instead of torn down, thus allowing the UEs to resume it quickly when it wishes to transmit new data. In the inactive state the bearers of the device are retained and reused on the device's next connection.

2.1.3.2 Procedures

In this section we describe the sequence of procedures that a device follows in order establish a connection with network, exchange data, and tear down its connection when it is no longer needed (figure 2.14).

To establish a new connection the device must first select the appropriate network using the *Cell (Re-)Selection* process and then associate with an BS through the *Random Access (RA)* process, which moves the device to the RRC Connected state. Following the RA process, the device needs to establish a connection with the core network using the *Attach* and *Connection (Re-)Configuration* processes, at the end of

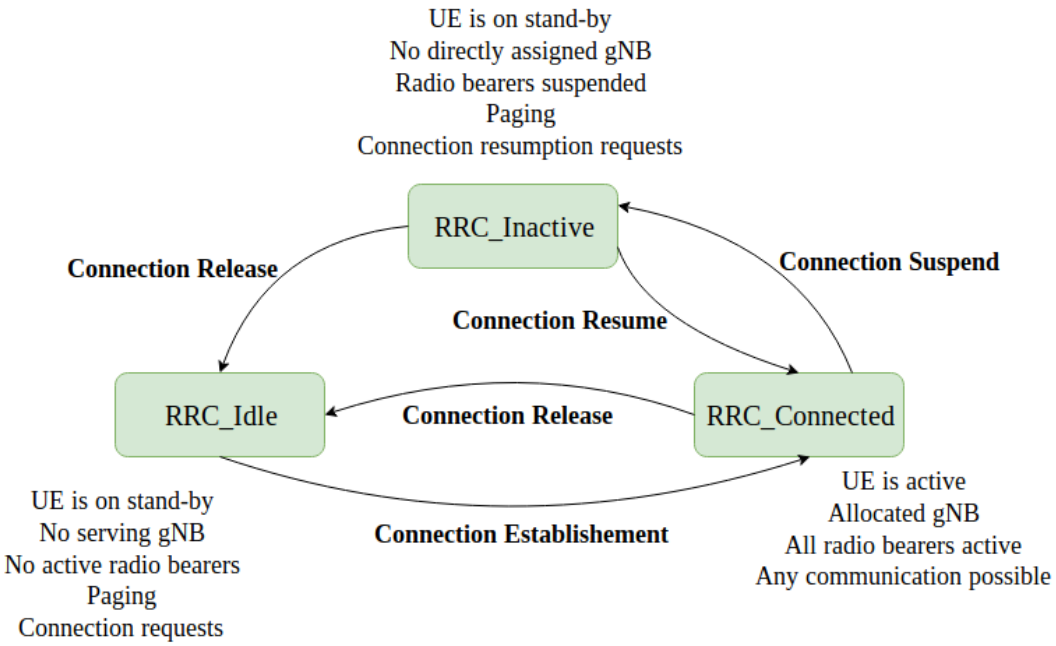


Figure 2.13: **RRC state diagram in 5G networks:** The figure depicts the RRC state diagram and the procedures required to transition from one state to the other. The new RRC Inactive state is also depicted.

which it will be able to exchange any data. This set of processes makes up the *Connection Establishment* process.

At the end of the data exchange, and if there are no other imminent transmissions, the network terminates the UE's connection and free up the allocated resources using the *Connection Release* process, moving the device back to the RRC Idle state. If the network needs to contact an idle device about network-originated data, or system configuration changes, it uses the *Paging* process. We now describe each of the individual processes in detail.

2.1.3.2.1 Cell (Re-)Selection When a UE is powered on, or upon loss of connectivity, it needs to identify an appropriate network to connect to, as multiple networks might be in operation in its area. To do so, the UE turns its radio to different frequency channels based on the frequency bands it supports, and on each frequency, it searches for the PSS and SSS to synchronise on subframe level on the time domain. This synchronisation is need in order to retrieve the cell's MIB, which is always transmitted in the PBCH. Following the decoding of the MIB, the UE decodes several SIBs to acquire essential information regarding the cell access. Specifically, the UE decodes the SIB1

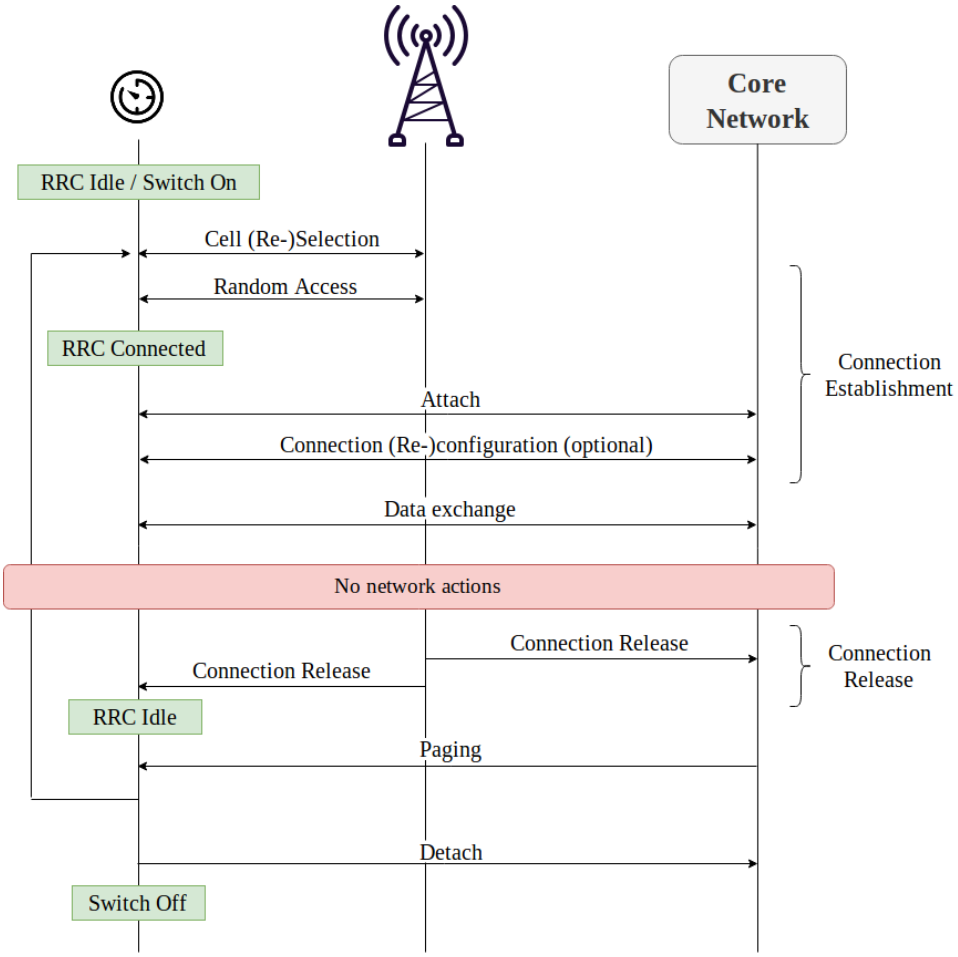


Figure 2.14: **Complete connection cycle in cellular networks:** The figure depicts a high-level overview of the complete connection cycle in cellular networks.

to retrieve the PLMN identity, and compares it with the PLMN identities stored on its USIM. If the considered network is suitable, the UE uses the scheduling information for SIB2 (which are contained in the SIB1) to retrieve the PRACH configuration information, as well as several random access parameters that are required for the *Random Access (RA)* process in the next step. Several SIBs might be transmitted by the BS, but the UEs are required to decode the MIB and at least the SIB1 and SIB2.

2.1.3.2.2 Random Access (RA) Process At the end of the Cell (Re-)Selection process the UE has achieved time synchronisation in the downlink direction, and has acquired all the required configuration information of the cell. However, it also needs to synchronise in the uplink direction before it attempts to establish a connection, as UEs in different distances from the BS will experience different propagation delays in their transmissions. This is achieved with the *Random Access (RA)* process (figure 2.15).

There are two types of RA process, the *contention-based* and the *contention-free*. In the former type, devices access the network by randomly selecting their resources, which leads in collisions when two or more devices select the same resource at the same time. In the latter type, the devices have pre-allocated resources, and thus collisions are avoided. This approach however is not feasible for deployments with large numbers of devices, as the available resources are not sufficient. It is however used when an already connected device has lost synchronisation after a handover to a neighbouring BS. We now only describe the contention-based approach, which is the one relevant to this thesis.

At the first step of the RA process the UE transmits a preamble in the PRACH. Preambles are specific signal patterns that the BS can identify. In each PRACH instance, there are 64 orthogonal preambles available (orthogonal means that the transmission of one preamble does not interfere with the transmission of the others), and the UE randomly chooses one of them. Based on the chosen preamble and the resource that it was transmitted on, the UE calculates the *Random Access - Radio Network Temporary Identifier (RA-RNTI)* that uniquely identifies the transmitted preamble.

After the preamble transmission, the BS detects the presence of preambles in the PRACH and replies with a *Random Access Response (RAR)* message for each identified preamble. The RAR messages are sent in the PDSCH and are addressed to the RA-RNTIs associated with the detected preambles. Each RAR message also contains the absolute *Timing Advance (TA)* value to control the uplink transmission timing of the UEs (i.e. how much in advance must the device start transmitting so that the re-

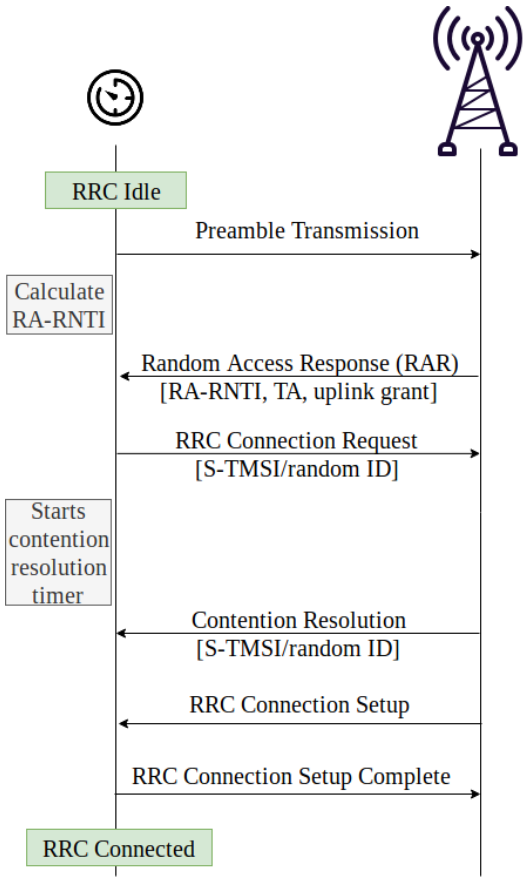


Figure 2.15: **Random Access (RA) process:** The figure depicts the Random Access (RA) process between the UE and the BS.

ceived signal at the BS is aligned with the uplink frame), an interim *Cell - Radio Network Temporary Identifier (C-RNTI)*, and an uplink scheduling grant for the UE's next transmission. The UE waits for the RAR message corresponding to its transmitted preamble for a period of time T_{RAR} equal to $3SFs + RAR\_response\_window$. The *RAR_response_window* is measured in subframes, and is specified by the network in the SIB2. If no such message is received within the T_{RAR} time the device backs off for an increasing random period of time, and tries again later. This process can be repeated up to $preambleTransMax$ times, after which the device declares a network outage. The $preambleTransMax$ value is also specified in the SIB2.

In the contention-based RA process, two UEs collide when they transmit the same preamble, at the same resource of the same RACH instance. However, the BS might not always be able to identify such a collision at the preamble transmission stage. In this case, the RAR message corresponding to the detected preamble is received by all devices that transmitted the given preamble, which then reply with an *RRC Connection Request* message using the scheduling grant and the timing advance specified in the RAR. In that message, the collided UEs include an identity, which is either their *SAE Temporary Mobile Subscriber Identity (S-TMSI)* (if one exists from a previous connection), or a random number if the former is not available. The UE also starts the contention resolution timer whose value is also specified in the SIB2.

When the BS receives the connection requests, it can determine whether a collision occurred by checking the identities in the messages (S-TMSIs or the random numbers). If a collision did occur, the BS randomly chooses at most one of the contenting devices to proceed with its connection, and responds back to it with a *Contention Resolution* message, that echoes the connection request message sent by the selected device. The contention resolution message also includes an uplink grant for the selected UE's next transmission. The UEs then compare the received contention resolution message with the one they sent, and determine whether they have been selected to proceed with their connections. If yes, the temporary C-RNTI becomes the UE's permanent C-RNTI for the remaining of its connection. Any devices that did not receive a contention resolution message before the expiration of the contention resolution timer assume that they have not been selected to proceed, discard their interim C-RNTIs, back off for a random period of time and try again later.

The BS then transmits an *RRC Connection Setup* message to the UEs that are allowed to proceed, which specifies the PHY/MAC/RLC setup for the SRBs, and sets the required configuration for the subsequent transmissions (e.g.the number of re-

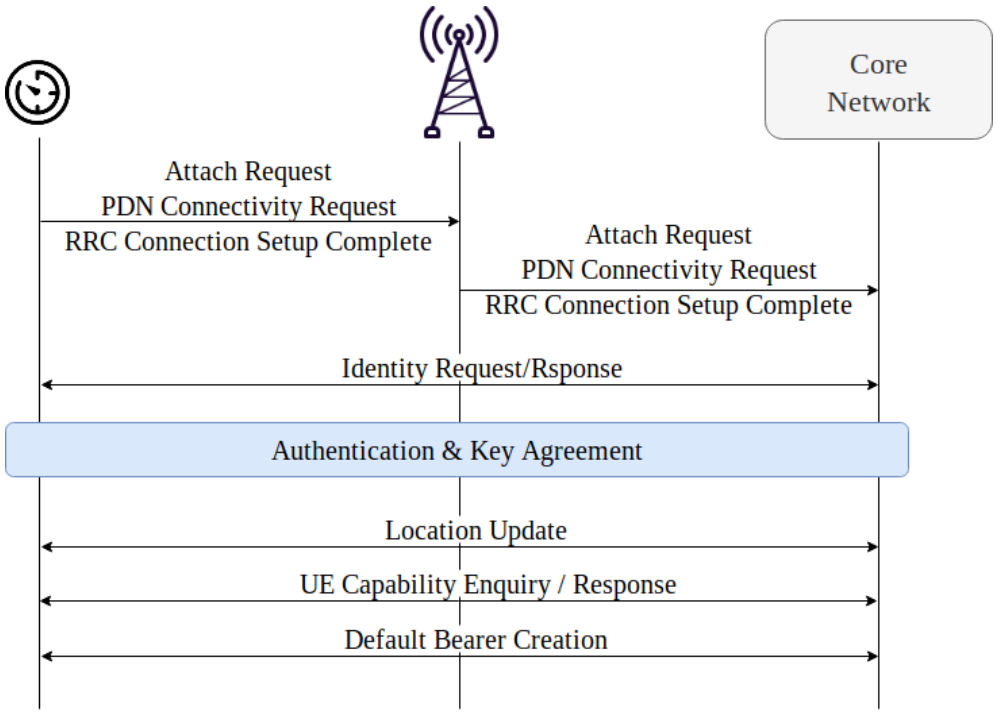


Figure 2.16: **High-level overview of the Attach process:** The figure depicts a high-level overview of the Attach process, that includes the AKA and the PDN Connectivity parts.

transmissions, the *Modulation and Coding Scheme (MCS)* to be used).

In the last step of the RA process, the UEs reply with and *RRC Connection Setup Complete* message to acknowledge the reception of the RRC Connection Setup message, and the correct configuration of the SRBs from its side. The RA process is now terminated and the UE switches to the RRC Connected state.

2.1.3.2.3 Attach Process The Attach procedure allows the device to register with the core network, and establish a connection with external *Public Data Networks (PDNs)*. This process is logically split into two parts. The *Authentication & Key Agreement (AKA)* part is used to mutually authenticate the UE and the core network, while the *PDN Connectivity* part is used to allocate an IP address to the UE, and connect to external networks through the P-GW. A high-level overview of the process is depicted in figure 2.16.

The attach process begins with the transmission of the *Attach Request* message, while the PDN Connectivity process begins with the *PDN Connectivity Request* message. To decrease the number of transmissions, both these messages are sent with the RRC Connection Setup Complete message of the RA process in one go. The BS then

sets up a dedicated connection with the MME.

During the AKA process the network authenticates the device, authorises its network access, and sets up a security context to be used in subsequent communications (section 2.1.5. At the same time, the UE also authenticates the network and verifies its legitimacy. After the mutual authentication, the device is required to update its location information to allow the network to forward downlink traffic directly to the correct MME and BS. This information is also used to efficiently page the device when it is in the RRC Idle state (section 2.1.3.2.7).

The final step of the Attach process is the PDN Connectivity, which takes place within the core network, and allows the entities to create a default session and a default bearer for the device, and allocate an IP address to the device. The default bearer provides a best-effort service with no QoS requirements, and is created to accommodate subsequent communication. To create this bearer the network questions the device about its capabilities, using the *UE Capability Enquiry* message. The UE's reply includes information such as the maximum data rate it can handle, it UE category, whether it supports half-duplex mode transmissions or not, and the latest specifications it conforms to. The network then, sets up the default bearer according to the UE's capabilities.

2.1.3.2.4 Connection Reconfiguration Process During the Attach process, the network creates the default, best-effort bearer. However the device can request additional, dedicated bearers when the QoS requirements of its applications cannot be met by the default bearer. This is be done with the Connection Reconfiguration process. Please note that regardless of additional bearers, the default bearer remains in place throughout the device's connection.

The process is similar to the default bearer creation, however, the dedicated bearers do not have different IP addresses. Instead, all dedicated bearers that belong to the same UE are linked with its default bearer, sharing the same IP address. After the Connection Reconfiguration process (if any), the core network sends an *Attach Accept* message that concludes the connection setup at the network side. The device is now able to exchange data with the network, by requesting scheduling grants on the PUCCH and transmitting its data in the PDSCH.

2.1.3.2.5 Connection Release Process Throughout the duration of an established connection, UEs are allocated PUCCH resources to allow them to request scheduling

grants for uplink data transmissions. After the data transmission, devices remain connected to the network for a specific amount of time (called herein as *active waiting*), even when no more data is being exchanged. This feature was initially designed for HTC devices, in order to avoid having to repeat the connection establishment process when they send/receive data within short periods of time. The duration of the active waiting period is defined by the *inactivity timer* [28], which is network operator specific, and is usually set between 10-50 seconds in most commercial networks. If the UE does not exchange data before the expiration of the inactivity timer, the network releases its connections and moves the device back to the RRC Idle state to free up its resources. This is done using the *Connection Release* procedure after which the SRBs, as well as the S1 and RAB bearers of the UE are torn down. If the UE wishes to transmit new data, it has to go through the complete connection establishment process again.

2.1.3.2.6 Connection Suspension & Resumption Processes In 5G networks, the new RRC Inactive state has been introduced, and the *Connection Suspension* and *Connection Resumption* process are used to switch between the RRC Connected and RRC Inactive states. These two procedures include fewer messages compared to the Connection Establishment and Connection release procedures, allowing for faster connections, and reduced energy consumption.

The *Connection Suspension* process suspends the device's previously established connection, and preserves vital information about it to allow for a fast resumption when needed. Upon suspension of the connection, the UE is provided with a *resumption ID*, and is then switched to the RRC Inactive state.

To resume a suspended connection, the device follows the RA process but transmits a *RRC Connection Resume Request* message, instead of the RRC Connection Request message, that contains its resumption ID. The gNB then decides whether to accept the resumption request or not. In the former case, the gNB retrieves the stored information of the device's previous connection, and automatically re-activates it without further signalling exchange, allowing the device to start exchanging data directly. Otherwise, the gNB rejects the resumption request, in which case the device follows the traditional connection establishment process (section 2.1.3). Please note that the first two steps of the RA process need to be followed here as well, and as such, the RRC Inactive state does not alleviate possible collisions in the RACH.

2.1.3.2.7 Paging Process Devices in the RRC Idle state do not have an active network connection and therefore, if new data arrives for the device, the network is unable to directly forward it. In this case, the network needs to page the device, and ask it to connect to the network to receive its outstanding data. This is done using the *Paging* procedure[1].

To page the device, the network transmits a paging request on the PCH. However, for idle devices, the network does not know their specific location on a BS granularity (this may have changed since the last time the device update its location to the network due to mobility). Instead, their locations are known to the network on a *Tracking Area* granularity, a logical area that is covered by a number of BSs. Therefore, the network instructs all BSs within the devices tracking area to transmit the paging request.

Each BS now sends the paging request on its paging channel. However, the UE does not check the paging channel at all times. The specific moments in time when the device checks the paging channel are called *paging occasions (POs)* and depend on the device's *Discontinuous Reception (DRX)* configuration. The DRX [29, 30] (figure 2.17) is an important feature in cellular networks that allows a device that is not sending or receiving data to turn off its reception (RF) and transmission (TX) modules, and enter a sleep mode, with minimal energy consumption. The DRX is split into the active (on) and sleep (off) period, which make the DRX cycle. During the active period, the device switches its RF module on and checks its paging occasion (PO) for paging requests. If non is detected, the device starts the sleep period, during which time its modules are turned off to reduce the energy consumption. Otherwise, the device retrieves the paging message. As multiple devices may have the same PO, the device needs to examine the list of paged IDs included in the paging message, to determine if it has actually being paged. If its ID is included in the list, the device connects to the network to receive the downlink data otherwise it directly switches its RF module off, and starts a new DRX cycle.

The length of a DRX cycle is usually negotiated between the BS and the device at connection time, and ranges from 0.32 to 2.56 seconds in 4G, and up to 10.24 seconds in 5G networks [30]. If there are no network originated data for several consecutive DRX cycles, the device can choose to increase the sleep period, and switch to a longer DRX cycle, until new data arrive from the network to further reduce the energy consumption. It is important to note that although DRX is mainly used while devices are

[1] Paging can also be used for connected devices. As the process is the same, and we do not make any distinctions.

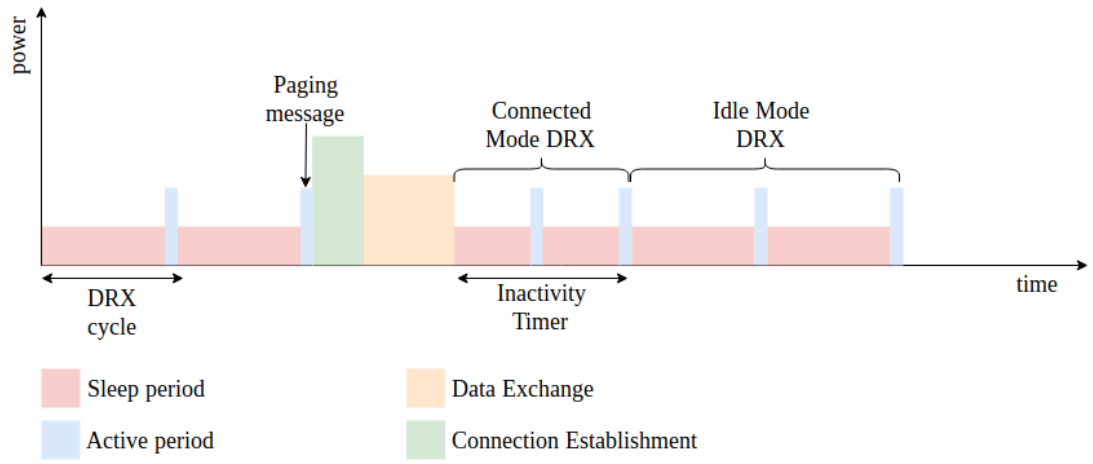


Figure 2.17: **DRX cycle example:** The figure depicts the operation of the DRX cycle with the active and sleep periods. Initially the device switches its RF and TX modules off during the sleep period. Then, during the active period it checks for paging messages. If there are none, it goes back to sleep. If a paging message exists, the device connects to the network to receive downlink data. After the data reception the device starts the inactivity timer and when it expires, during which time it can get into C-DRX with a short cycle. At the end of the inactivity timer the device switches to the RRC Idle state, and starts a new I-DRX cycle with longer cycles.

in the idle state (I-DRX), it is also possible to use the DRX cycle in the connected state (C-DRX) with short cycles, to reduce the energy consumption even in this case.

2.1.3.2.8 Detach Process The *Detach* process is the last step in a device's communication procedure. This process de-registers the device from the network and terminates any established bearers and security contexts both in the RAN and the core network. This procedure is usually initiated by the device before switching off.

2.1.3.3 evolved Multimedia Broadcast Multicast Service (eMBMS)

The *evolved Multimedia Broadcast Multicast Service (eMBMS)* [31] is a 3GPP subscription-based standard to support point-to-multipoint services by means of either single- or multi-cell (also known as single-frequency) transmissions. It was first introduced in Release 6, but has significantly changed throughout the subsequent releases to provide efficient transmissions of multicast content. In order to handle the devices subscriptions, the co-ordination of the BSs that participate in the transmission and the delivery of the multicast content, eMBMS introduces two new logical channels and several new

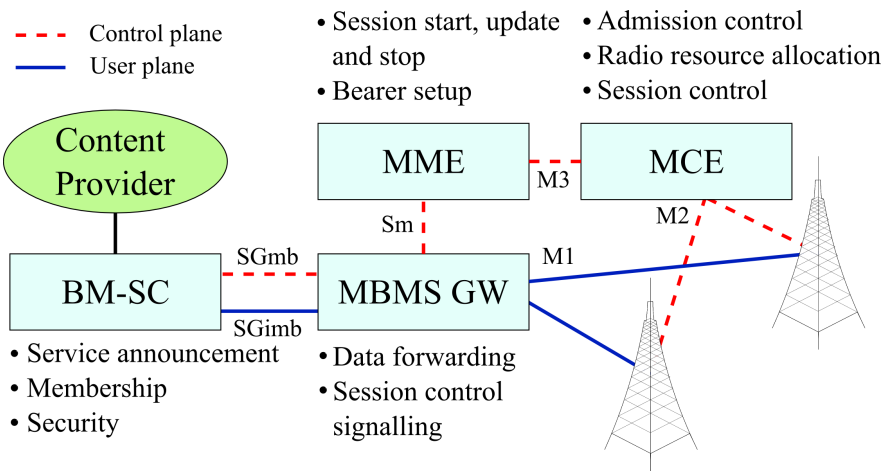


Figure 2.18: **evolved Multimedia Broadcast Multicast Service (eMBMS) architecture:** The figure depicts the eMBMS architecture with the different physical entities, and the main functionalities for each of them.

physical entities in the core network (figure 2.18). In the remainder of this section the main features of eMBMS are described. More details can be found in [31, 32, 33].

2.1.3.4 Channels

eMBMS introduces two dedicated logical channels:

1. the *Multicast Transport Channel (MTCH)*, and
2. the *Multicast Control Channel (MCCH)*

The MTCH is used for the transmission of data traffic, while the MCCH is used for transmission of control information of the multicast services. To transmit multicast content, one or more PDSCH subframes need to be allocated to eMBMS. Both channels can be mapped in the same eMBMS subframe, which may contain one MCCH and one or more MTCHs. At the transport layer, MTCH(s) and MCCH are multiplexed in the *Multicast Channel (MCH)*, which is carried by the *Physical Multicast Channel (PMCH)* in the physical layer (section 2.1.2.3).

2.1.3.5 Procedures

To deliver multicast content, the eMBMS framework employs eight different procedures (figure 2.19):

- the **Service Announcement** procedure informs the devices about the offered services. These might be ongoing, imminent, upcoming or pending.

- the **Service Subscription** procedure is used by users to establish an agreement with the content provider to receive eMBMS services.
- the **Joining** procedure allows a device to join a multicast group before, or during the eMBMS service.
- the **eMBMS Notification** procedure is used by the network to inform the devices about imminent or ongoing services.
- the **Session Start** procedure is used by the network to reserve the required resources for the multicast service, and set up the required RAN and core network bearers.
- the **Data Transfer** procedure transmits the eMBMS service to the subscribed devices.
- the **Session Stop** is used by the the network to release the allocated resources, and tear down or de-activate the bearers, until a future transmission.
- the **Leaving** is used by a user to leave a multicast group. If the user wishes to start receiving that service again, it needs to re-subscribe to the service.

2.1.3.6 Operation

eMBMS is a subscription-based standard, according to which the network operator periodically announces the provided services and their status (i.e. ongoing, imminent, pending, inactive, etc.), in the form of control information in the MCCH (*service announcement*). Devices indicate which services they are interested in receiving (*service subscription*), and join the service by receiving a group ID that uniquely identifies that service (*joining*). They then start monitoring the MCCH for notifications regarding their service (*eMBMS notification*) and setup the communication bearers (*session start*). Devices can join a service before it starts or while it is ongoing. Upon termination of the transmission of the service, the allocated resources are released, and the session is terminated (*session stop*). Please note that the session stop does not indicate the termination of the service. Subscribed devices still need to periodically monitor the MCCH for information regarding future sessions of their service. When a device wishes to stop receiving the service completely (*leaving*), it discards the group ID, deregisters from the service, and stops monitoring the MCCH for control information. In

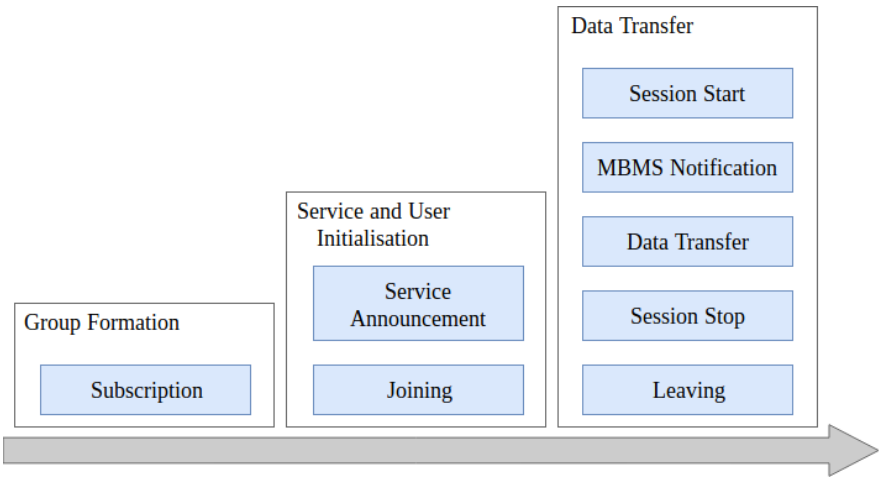


Figure 2.19: **eMBMS procedures:** The figure depicts the different procedures for the reception of an eMBMS service.

eMBMS, the subscription, joining, and leaving are performed on a per-device basis, while the rest of the procedures are performed on a per-eMBMS service basis.

2.1.3.7 eMBMS over a Single Frequency Network (MBSFN)

eMBMS supports two transmission modes depending on how many BSs are involved in the transmission. In the *single-cell* mode each BS transmits eMBMS data only to the devices within its cell, thus transmission parameters are adapted independently from neighbouring BSs. In the *multi-cell* mode, a set of BSs simultaneously transmit the same eMBMS content, composing a single frequency network (*eMBMS over a Single Frequency Network (MBSFN)*). Within the MBSFN, BSs are tightly synchronised in time and frequency and transmit the same data in the same frames/subframes and frequency resources, so that the resulting signal appears to the devices as a single transmission over a time-dispersive channel. In this case, only the extended CP may be used to avoid inter-symbol interference. However, the use of the extended CP limits the number of resources for data transmission, thus reducing the spectral efficiency.

2.1.4 Single Cell - Point to Multipoint (SC-PtM)

The *Single Cell Point-to-Multipoint (SC-PtM)* framework is a complimentary bearer service to the existing eMBMS, that provides multicast transmissions in a single cell. As such, SC-PtM inherits the subscription-based architecture of eMBMS, as well as its several procedures (section 2.1.3.5).

SC-PtM uses the following two logical channels, which are signaled in the SIB20:

1. the *Single Cell-Multicast Control Channel (SC-MCCH)* carries control information, and

2. the *Single Cell-Multicast Traffic Channel (SC-MTCH)* carries data transmission

In contrast to the previous implementation of eMBMS where the MCCH and MTCH occupied a whole subframe, the new channels are scheduled in PDCCH and can be multiplexed with the unicast traffic. The control information is transmitted periodically, with a scheduling period specified by the network operator [34]. The multicast data is scheduled using a *group Radio Network Temporary Identifier (g-RNTI)* that uniquely identifies the service. To deliver the data, a *Single-Cell Multimedia Radio Bearer (SC-MRB)* is used, which is set up before the session start, and is accessed by all devices receiving the same service.

2.1.5 Security Framework

In this section we present the security framework used in LTE and 5G networks. The same framework has also been inherited in newer cellular network technologies, such as the LTE-M or NB-IoT, and ensures that data sent between the network and the device is encrypted and its integrity protected.

The security framework relies on a mutually agreed hierarchy of keys and associated algorithms to generate them [35], which are referred to as the device's *security context*. The security context needs to be set up on the device's initial connection to the network, and renewed from time to time throughout the device's life. In fact, 3GPP advises that a new security context should be established at each connection [35]. The security context is derived from a unique, user-specific, root key K that is stored solely on the SIM card of the device and the core network entity responsible for user authentication (e.g. the HSS in 4G networks, or the AUSF in 5G networks). The integrity of the security framework relies on the assumption that K will not be disclosed to, or stolen by unauthorised parties, and is thus never shared with any other entities within the whole network. A one-to-one mapping exists between the device's globally unique IMSI, saved on the device's USIM card, and its K value, and thus its security context.

Each time a device establishes a new connection, the network checks whether a security context already exists for it (i.e. the device has previously attached, and the security context has not expired). If not, the network requests its IMSI to identify the

device, and retrieve its K key. Note that, as no security context exists at this stage, the identity request message and the response that includes the IMSI are sent unencrypted. Next, the device and the network need to mutually authenticate. To do so, the network starts by generating the *authentication vector*, which comprises of:

1. a random number ($RAND$), used as an authentication challenge,
2. the expected response to the challenge ($XRES$), and
3. an authentication token ($AUTN$) derived from K.

The $AUTN$ includes a sequence number to prevent replaying attacks [36]. The $RAND$ and $AUTN$ parts are sent to the device, while the $XRES$ is securely distributed to the appropriate entities within the core network for verification. The device uses the $AUTN$ to authenticate the identity of the network, based on the assumption that it can only be produced by someone with knowledge of K, authenticating the network in this way. It then derives the answer to the challenge using the $RAND$ and its root key K. The network can now authenticate the device by comparing its response to $XRES$, based on the assumption that only someone with knowledge of K could have produced a valid response to the challenge [35]. The mutual authentication procedure is followed by the set up of the security context. Specifically, the K key is used to derive the ciphering key CK and the integrity key IK, using secure and lightweight functions such as EEA2 [35]. In turn, these keys are used to derive further keys used for encryption (e.g. K_{NASenc}, K_{RRCenc}) and integrity protection (e.g. K_{NASint}, K_{RRCint}) of the transmitted control information and user data[2]. The complete key hierarchies for 4G and 5G networks are depicted in figure 2.20.

According to the specifications [35], the network can request a re-authentication of a device as often as it wishes, even if the device is already connected to the network. In some cases, the network is obliged to delete the context of a device, and request a re-authentication at the device's next connection, such as the *Tracking Area Update (TAU)* process.

2.1.5.1 Identity Privacy Mechanism

A recent addition to the security framework in 5G networks is the *Identity Privacy Mechanism* [35], which protects against leaking the IMSI to malicious eavesdroppers.

[2]User data is not required to be encrypted or integrity protected, and it is up to the device manufacture to implement this feature. In contrast, control data are required to always be both encrypted and integrity protected.

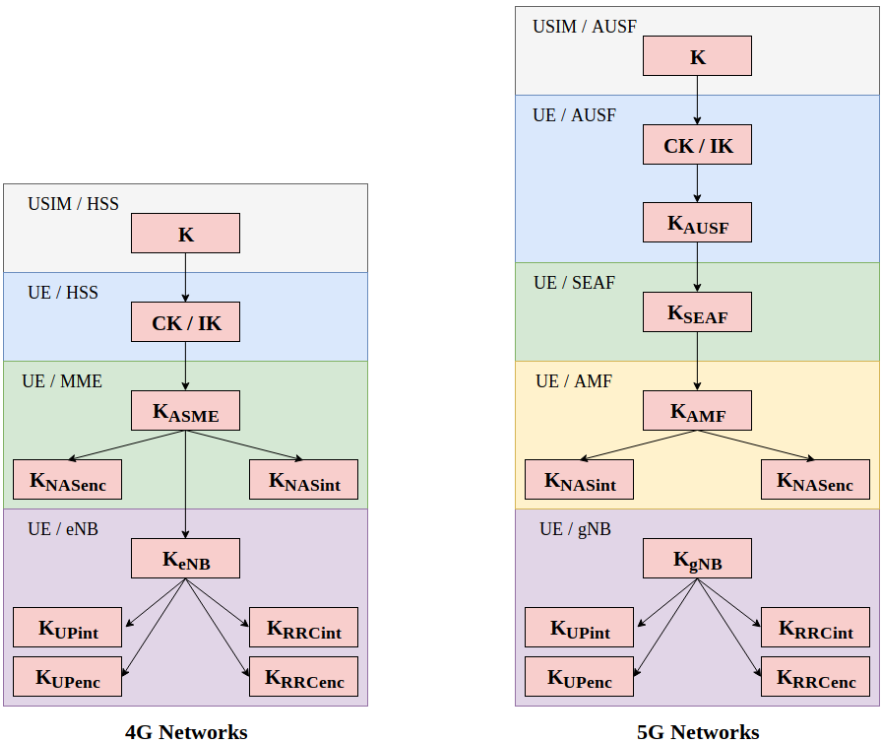


Figure 2.20: **Security keys hierarchy in 4G and 5G:** The figure depicts the hierarchies of the security keys in 4G and 5G networks, along with the physical or logical entities that each one of them corresponds to.

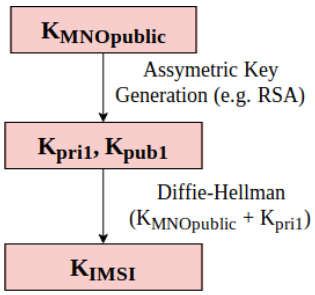


Figure 2.21: **Identity Privacy Mechanism (IPM) for secure IMSI transmission:** The figure depicts the key generation process according to the Identity Privacy Mechanism (IPM) for the secure transmission of the IMSI.

The IMSI is an important identifier that allows a device to connect to the network when no other information is available. Having gained access to the IMSI, a malicious actor can launch a range of attacks, including user-targeted DoS attacks [37, 38, 39], and leaking of user location history [36, 40, 41, 42]. For these reasons, 3GPP introduced the *Subscription Permanent Identifier (SUPI)* and the *Subscription Concealed Identifier (SUCI)* for the upcoming 5G networks [43, 35], to protect the IMSI from unauthorised parties. These identifiers aim to conceal the IMSI from unauthorised parties based on a *Public Key Infrastructure (PKI)*, where the IMSI is sent in an encrypted form using a combination of public key generation, ephemeral key generation using Diffie-Hellman, and symmetric encryption (figure 2.21).

Specifically, in order to transmit its IMSI, a device first generates a pair of ephemeral private-public keys using the home network's public asymmetric key, which is permanently stored on the device's USIM. The ephemeral key pair can be generated using any asymmetric algorithm mutually supported by both the network and the device (e.g. RSA [44]). The device then uses the Diffie-Hellman algorithm to generate a second ephemeral key using the home network's public key and its own ephemeral private key. This second ephemeral key is the one used to encrypt the IMSI with a symmetric algorithm such as AES. To avoid replay attacks, 3GPP specifies that new keys must be generated for each IMSI transmission, thus multiple encrypted messages cannot be used to infer the IMSI.

2.2 Evolution of IoT-specific technologies

As the number of IoT devices increased, and their traffic requirements could not easily be met by traditional cellular network technologies (e.g. 3G), researchers and network

operators started developing a more application-specific categorisation of IoT applications started emerging with the goal of identifying the key challenges of the major categories, and thus drive the deployment of IoT-specific technologies. These categories include, but are not limited to:

1. **Massive IoT** (e.g. smart metering, asset management) includes devices that are usually deployed in large numbers, such as sensors, meters, wearables and trackers. Additionally, they are low-complexity devices that send or receive data infrequently, and in pre-defined periodicities, and the applications they run are often delay-tolerant. An important characteristics is that such devices are often deployed in signal-challenged locations such as basements, and therefore, deep signal penetration. Furthermore, the challenging locations means that these devices may rely solely on a battery power supply, which puts extreme requirements on the device's battery life.

2. **Broadband IoT** (e.g. fleet management, Augmented Reality (AR)/Virtual Reality (VR)) includes devices that require higher data rates, low latency and high reliability compared to massive IoT. At the same time they are also characterised by functionalities that are specific to massive IoT, such as extended coverage and increased battery life.

3. **Critical IoT** (e.g. traffic safety and control) includes devices that depend on extremely low latency ($\approx$ 100ms) connections, and/or high reliability of up to 99.9999%. Use cases in this category include smart grids, smart manufacturing, intelligent and autonomous transportation (V2X), or devices that allow distant human interaction, such as tele-operated driving or remote surgery.

4. **Industrial automation IoT** (e.g. collaborative robotics, smart grid automation) includes devices that are primarily focusing on manufacturing, but also extend to any other industrial application with similar, such as railway control systems, or power generation and distribution.

Despite the different requirements, the common goals of all IoT technologies are low-power performance and long-range coverage, and as such these technologies are also known as *Low Power Wide Area Network (LPWAN)* technologies. In the unlicensed spectrum technologies such as SigFox [45], LoRa [46] and ZigBee [5] were deployed, however they were prone to interference and thus resulted in low data rates

and increased latency. Furthermore, the unlicensed nature meant that deployment was limited, thus increasing their overall cost.

In the licensed spectrum, the development of IoT-specific technologies followed the standardisation path of the 3G, 4G and 5G technologies. Initially, Cat-1 was introduced during the 4G area. Although its performance was inferior to that of 3G, it is considered a good option for low cost and delay-tolerant massive IoT devices. However, the complexity and device cost of this category was above the requirements for IoT, and therefore, Cat-0 was later developed with the goals of long battery life and low cost on top of the existing support for massive number of devices, enhanced coverage (increased signal penetration), and long-range/wide spectrum. To deliver low-cost functionality, Cat-0 sacrifices features that supported medium to high data rates. Cat-0 was further developed to Cat-M1/Cat-M/LTE-M technologies that are often viewed as the second generation of LTE designed specifically for IoT applications. These technologies further reduce the cost and power of IoT devices by putting an upper limit on the system bandwidth (1.4 MHz), and are each specialised on a different IoT use case.

Close to the end of the 4G era, NB-IoT (also called Cat-M2) was developed. NB-IoT is similar to Cat-M1/Cat-M/LTE-M, but is focused on much lower data rates in order to further decrease the battery life and increase the coverage. Furthermore, the elimination of features such as dual connectivity and mobility resulted in further decrease of the device cost. Nowadays, the major cellular IoT technologies are LTE-M and NB-IoT, with each of these technologies targeting different IoT use cases.

2.3 NarrowBand-Internet of Things (NB-IoT)

NB-IoT, is a new, licensed, *Low Power Wide Area Network (LPWAN)* radio technology, that focuses on increased indoor coverage for massive numbers of low-cost, low-capability and low-power IoT devices. As a new 3GPP radio-access technology, NB-IoT is not fully backwards compatible with previous generations of cellular networks, meaning that existing devices are not able to use it directly. However, it has been designed to co-exist with legacy architectures [47] by re-using the physical layer design of the existing systems to a great extent. According to 3GPP's Rel. 13 specification [48], NB-IoT targets the following four goals (figure 2.22):

1. **Extended coverage:** Provide enhancements to increase the indoors coverage by 20dB.

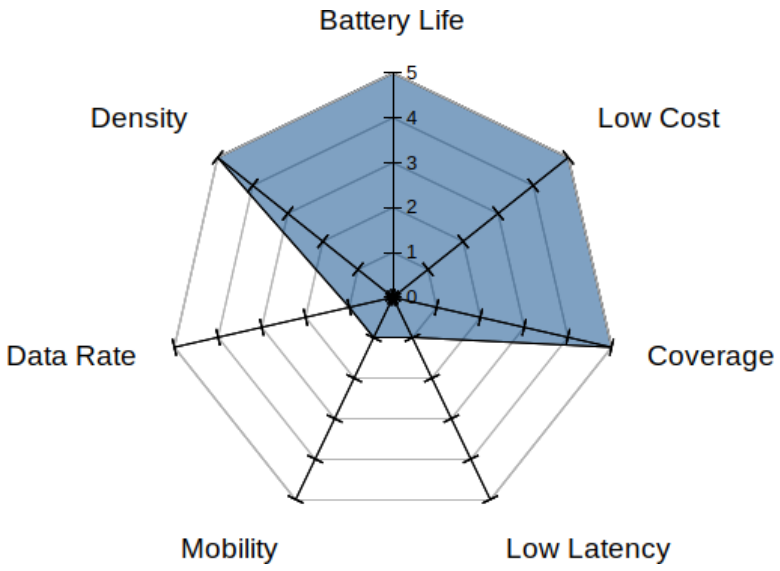


Figure 2.22: **Performance targets for NB-IoT:** The figure depicts the major performance targets of the NB-IoT technology.

2. **Increased battery life:** Provide low-complexity procedures to increase battery life to 10 or more years on a single charge. Furthermore, the required initial battery capacity is 5 Wh.

3. **Low cost:** Allow for a total cost of $5 per device.

4. **Support of massive numbers of devices:** Support at least 52547 low throughput devices within the same cell. 3GPP's traffic model assumes 40 devices per home, or 20 devices per person.

NB-IoT aims to improve indoors coverage by 20dB by realising a Maximum Coupling Loss (MCL) of 164dB[3], while supporting a minimum data rate of 160 Bps at the application layer. Furthermore, NB-IoT implements several IoT-oriented enhancements, such as narrow-band transmissions that allows the receiver to filter out more noise, thus improving the Signal to Interference and Noise Ratio (SINR), use of repetitions and differentiation of UE performance [29, 49, 50]. These enhancements are expected to achieve coverage enhancements of 20dB in order to reach devices in signal-challenged locations, such as basements or underground.

As IoT devices are typically battery powered and expected to operate for long periods of time on a single battery charge, NB-IoT aims to achieve at least 10 years of

[3]Maximum Coupling Loss has been chosen by 3GPP as the metric to evaluate coverage of a radio access technology. In theory, it can be defined as the maximum loss in the conducted power level that a system can tolerate and still be operational (defined by a minimum acceptable received power level).

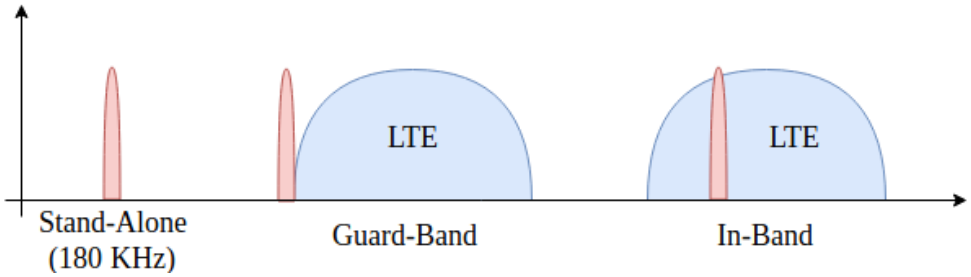


Figure 2.23: **NB-IoT deployment modes:** The figure depicts the three different deployment modes of NB-IoT: (i) stand-alone where NB-IoT is deployed in unused frequencies, separately of other cellular network deployments, (ii) guard-band where NB-IoT is deployed on one of the guard bands of a 4G/5G deployment, and (iii) in-band where NB-IoT is deployed in one or more of the PRBs of an existing 4G/5G deployment.

battery life by incorporating features such as *Power Saving Mode (PSM)*, and *extended Discontinuous Reception (eDRX)*. Finally, NB-IoT offers a limited set of functionality compared to existing cellular technologies in order to reduce its complexity and thus, the cost of the NB-IoT devices. For example, functionality such as connected-mode mobility and dual connectivity are not supported by NB-IoT.

In this section we provide a detailed presentation of the NB-IoT technology, highlighting its major similarities and differences compared to existing cellular network technologies. We begin by presenting the different deployment options and how they can co-exist with existing 4G/5G deployments. We then present the physical layer design. Finally, we present new and altered communication procedures. Part of this work has been published in IEEE Communications Magazine 2019 [22].

2.3.1 Deployment Options

NB-IoT offers great deployment flexibility that allows a network operator to use part of its available spectrum, without affecting the already deployed 4G or 5G technology. NB-IoT can be deployed in three different modes: in-band, guard-band or stand-alone (figure 2.23). In the in-band mode, the NB-IoT carrier occupies one or more PRB within a wideband 4G/5G carrier[4]. In the guard-band mode, NB-IoT is deployed in the guard bands[5] of the 4G/5G carrier. Finally, in the stand-alone mode, NB-IoT is deployed in unused frequencies, away from a 4G/5G carrier.

[4]The carrier must be larger than 1.4 MHz.

[5]A guard band is an unused part of the radio spectrum between radio bands, for the purpose of preventing interference.

One of the major design goals of NB-IoT is the increased coverage of 20dB. Towards this end, NB-IoT supports up to three coverage enhancement (CE) classes (0 to 2) in each of the operation modes, that correspond to *normal*, *robust* and *extreme* coverage conditions. CE classes are differentiated by thresholds based on the signal strength, the cell deployment, the propagation environment (i.e. outdoors, indoors, deep-indoors, underground, etc.), and the spatial distribution of the devices. In order to increase the success probability of signal reception, NB-IoT supports up to 2048 repetitions in the downlink, and up to 128 repetitions in the uplink. Each replica can be decoded separately, or multiple replicas can be combined to further increase the reception probability. The number of repetitions can be tuned separately for each CE class.

2.3.2 Physical Layer Design

NB-IoT uses the existing cellular network design to a great extent to allow for seamless co-existence and interoperability. Regardless of the deployment mode, NB-IoT requires a minimum channel bandwidth of 180 KHz, which corresponds to one PRB. Similar to 4G, NB-IoT supports OFDMA transmissions in the downlink and SC-FDMA transmissions in the uplink. A subcarrier spacing of 15 KHz is supported in the downlink, while in the uplink either a 15 KHz or 3.75 KHz subcarrier spacing can be applied.

2.3.2.1 Frame Structure

NB-IoT follows the same numerology used in 4G networks both in the uplink and downlink. In the downlink, OFDMA is used with normal cyclic prefix. In the time domain, the radio frames, subframes, slots and symbols have the same duration as in 4G, while in the frequency domain the channel is divided into 12 subcarriers of 15 KHz each. In order to allow for interoperability when the NB-IoT is deployed in-band, the first three OFDM symbols in each subframe are not used, as they may carry the PDCCH of 4G.

In the uplink direction, SC-FDMA is applied on 12 subcarriers with 15 KHz subcarrier spacing and normal cyclic prefix. To further improve the coverage, 48 subcarriers with 3.75 kHz subcarrier spacing each are also supported, which are used for the preamble transmission of the RA procedure (section 2.3.3), and optionally for uplink transmissions. In this case, the slot lasts for 2 ms and each frame is composed of 5

slots. The two different frame structures are depicted in figure 2.24.

2.3.2.2 Signals

Similarly to 4G and 5G networks, different signals are used in NB-IoT to allow devices to discover, synchronise with, and connect to the BS. These signals are:

- the *Narrowband Primary Synchronisation Signal (NPSS)*,
- the *Narrowband Secondary Synchronisation Signal (NSSS)*,
- the *Cell-Specific Reference Signal (CRS)*,
- the *Narrowband Reference Signal (NRS)*,
- the *Narrowband Master Information Block (MIB-NB)*, and
- the *Narrowband System Information Blocks (SIBs-NB)*

The *Narrowband Primary Synchronisation Signal (NPSS)* is used for initial time and frequency synchronisation of the device in the downlink, and to get partial information regarding the cell identity. This signal is always transmitted on subframe #5 of every NB-IoT frame (i.e. it has a 10 ms periodicity), using the last 11 OFMD symbols.

The *Narrowband Secondary Synchronisation Signal (NSSS)* is used to accomplish full time and frequency synchronisation in the downlink by carrying the *Narrowband Cell Identity (NCellID)*. This signal is always transmitted in subframe #9 of every odd frame (i.e. it has a 20 ms periodicity), using the last 11 OFDM symbols.

The *Cell-Specific Reference Signal (CRS)* is the 4G/5G reference signal that is always present if the NB-IoT is deployed in-band, to allow for interoperability between the two technologies. This signal is only used by devices using 4G/5G in order to synchronise without noticing the presence of the NB-IoT deployment.

The *Narrowband Reference Signal (NRS)* is the equivalent of the CRS for the NB-IoT devices. This signal is the reference point for the downlink transmission power, and is used to estimate the channel quality and adapt the signal transmission accordingly. The REs occupied by the NRS depend on the NCellID.

The *Narrowband Master Information Block (MIB-NB)* is the equivalent of the MIB in 4G/5G, containing the same information for the NB-IoT deployment. For transmission, the MIB-NB is 34-bit long and is split into 8 blocks. Each block is then repeated

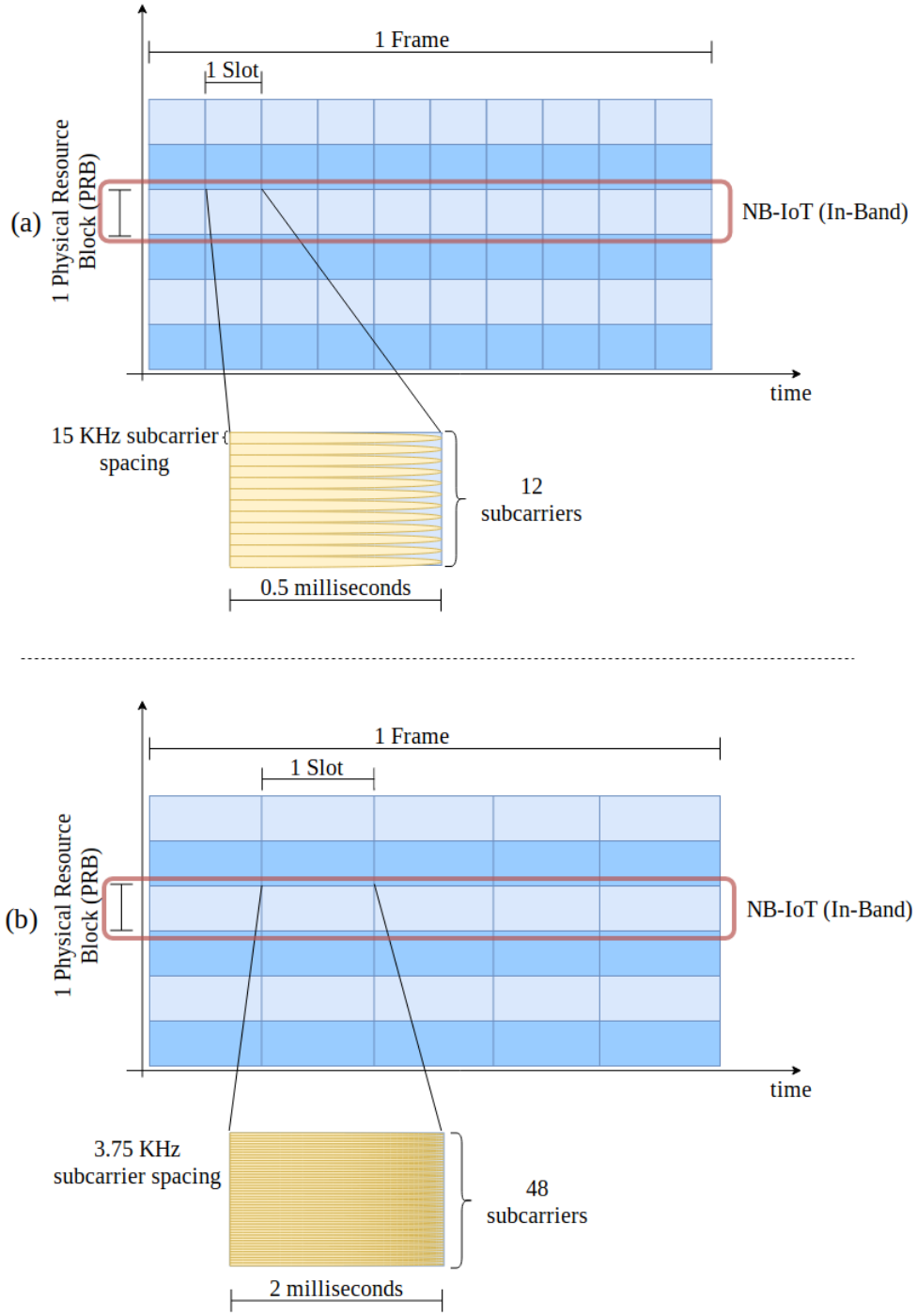


Figure 2.24: **Frame structures in NB-IoT:** The figure depicts the two different frame structures that can be used in NB-IoT, with 12 subcarriers and 15 KHz subcarrier spacing (subfigure (a)) and 48 subcarriers with 3.7 KHz subcarrier spacing (subfigure (b)). The difference in the slot length is also depicted.

8 times, resulting in an overall transmission time of 640 ms. This signal is always transmitted on subframe #0 of every frame.

Finally, the *Narrowband System Information Blocks (SIBs-NB)* are system information messages that carry cell configuration information. As with 4G/5G, the most important is the SIB1-NB that delivers the TAC, the PLMN identity, and the scheduling information of other SIBs-NB. The duration of SIB1-NB is 2650 ms, which it is transmitted on subframe #4 of 16 consecutive frames, and be repeated 4, 8 or 16 times. Another important SIB is SIB2-NB that carries configuration information about the paging channel and the required parameters for the RA process.

2.3.2.3 Channels

As with 4G/5G, data and control information is transmitted on both directions using different channels. However, the number of channels in NB-IoT are significantly less, in order to reduce the complexity of the technology. Furthermore, in contrast to 4G/5G, each NB-IoT subframe can only carry one of the channels.

In the downlink direction NB-IoT defines the following physical channels:

- the *Narrowband Physical Broadcast Channel (NPBCH)*,
- the *Narrowband Physical Downlink Control Channel (NPDCCH)*, and
- the *Narrowband Physical Downlink Shared Channel (NPDSCH)*

The *Narrowband Physical Broadcast Channel (NPBCH)* is used to broadcast information about the cell and network configuration through the MIB-NB. This channel always occupies the subframe #0 of every NB-IoT frame.

The *Narrowband Physical Downlink Control Channel (NPDCCH)* is used to carry UE-specific DCIs for both data reception and transmission.

The *Narrowband Physical Downlink Shared Channel (NPDSCH)* is used for data transmission from the BS to the UE. It is scheduled in the NPDCCH, and is used for dedicated data transmission towards the UEs, RRC signalling, and transmission of the SIB-NBs. The NPDSCH is always transmitted 4 ,ms after the NPDCCH to give low-capability NB-IoT devices plenty of time to decode the NPDCCH.

Apart from the subframes statically allocated to NPBCH, NPSS and NSSS, the rest of the downlink subframes are dynamically allocated to either NPDCCH or NPDSCH. The complete frame with the different channels is depicted in figure 2.25.

In the uplink direction, only two channels are defined:

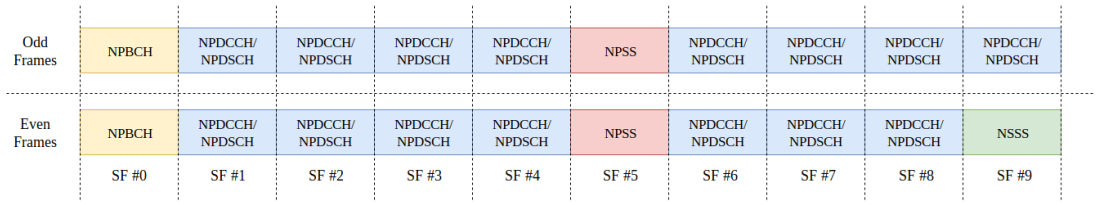


Figure 2.25: **NB-IoT downlink frame:** The figure depicts an NB-IoT downlink frame, and the channels mapped in each subframe. Subframe #0 and #5 of every frame are always occupied by the NPBCH and NPSS respectively, while subframe #9 of every even frame always carries the NSSS. The remaining subframes are dynamically allocated to either NPDCCH or NPDSCH.

- the *Narrowband Physical Random Access Channel (NPRACH)* and

- the *Narrowband Physical Uplink Shared Channel (NPUSCH)*

The *Narrowband Physical Random Access Channel (NPRACH)* is used to perform the RA procedure. It is composed of a contiguous set of either 12, 24, 36 or 48 subcarriers with 3.75 KHz spacing, which are repeated with a periodicity of $40 - 2560$ ms. The number of repetitions, the periodicity, and the number of subcarriers used for the NPRACH are separately determined for each CE class.

The *Narrowband Physical Uplink Shared Channel (NPUSCH)* occupies the remaining uplink resources, and is used for the data transmission from the UE towards the network. In terms of modulation schemes only Binary Phase Shift Keying (BPSK) or Quadrature Phase Shift Keying (QPSK) are supported, and a UE can either use a single subcarrier or multiple subcarriers for its data transmission.

2.3.3 Communication Procedures

Most of the communication processes used in NB-IoT are inherited from 4G/5G, however some of them have been adapted to operate using the limited resources available in NB-IoT. In this section, we only describe the RA process, and the processes for data transmission and reception that present differences compared to those used in 4G/5G networks. Finally, we present the power saving enhancements that have been introduced in NB-IoT to increase the battery life.

2.3.3.1 Random Access Process

The RA procedure starts with the transmission of a preamble, that lasts either 5.6 ms or 6.4 ms, depending on the size of the cell. To improve the cell coverage, a preamble can be transmitted up to 128 times. A preamble in NB-IoT is composed of four symbol groups, each of which is transmitted on a different subcarrier. The first subcarrier is chosen randomly, while the following ones are determined according to a deterministic sequence that depends on the initial subcarrier. Hence, in each NPRACH the number of orthogonal preambles is equal to the number of its subcarriers [50]. Since the preamble symbol groups depend on the initially selected subcarrier, collisions occur when UEs select the same initial subcarrier, and until the end of the preamble sequence.

Similarly to 4G/5G networks, the RA includes four messages. After the preamble transmission the UE starts a RAR window timer. However, in contrast to 4G/5G the RAR window timer lasts from 2 to 10 times the NPDCCH period, during which, the UE expects to receive the RAR message. The RAR includes the RA-RNTI which univocally identifies the preambles, and allows the UE to determine if the RAR is addressed to it. In each RAR, the BS provides the TA information, and an uplink grant for the transmission of the RRC Connection Request. The UEs that did not receive a RAR within their RAR window back-off for a random period of time and attempt the RA process again later. The remaining of the RA process is identical to RA process in 4G/5G networks, with only the following differences:

1. The Contention Resolution timer lasts from 1 to 64 times the NPDCCH period.

2. Devices that were not selected to proceed, back off for a period of time of up to ≈ 9 minutes.

3. If a device reaches the maximum number of attempts of its coverage class (up to 10 in most commercial networks), it starts performing the RA process in another coverage class. If the total number of attempts in *all* coverage classes is reached without succeeding, the UE declares an RA failure and stops trying.

2.3.3.2 Data Transmission & Reception Processes

Once the UE has established a connection with the network, it can start transmitting/receiving its data. In the downlink direction, the BS uses the DCI to inform a UE about an imminent downlink transmission. The DCI indicates the resource allocation, the number of subframes that the transmission spans, the number of repetitions

of each transmissions, and whether an ACK is being expected. If repetitions are indicated, then identical copies of the data are transmitted in consecutive subframes, using one subframe inter-leaving. If no repetitions are used, the transmission is mapped in continuous subframes.

2.3.3.3 3GPP Optimisations for NB-IoT Device Energy Savings

To better accommodate battery constrained NB-IoT devices, 3GPP has specified both *Control Plane (CP)* and *User Plane (UP)* optimisations [51]. In the CP optimisation, the devices encapsulate application data in control messages and transmit them over their SRBs, avoiding the need to setup a new DRB at each connection, essentially skipping the Attach process. Although the support for CP optimisation is mandatory, the use of the SRBs limits the size of the data that can be encapsulated, and as such, it can only be used for small data transmissions. Additionally, the QoS of the data transmission is upper bounded by the QoS that can be supported by the SRBs, which might be unsatisfactory to the device application. Finally, the transmission of application data in control messages does not allow for encryption of the transmitted data. Therefore, to accommodate larger packet transmissions, better QoS and secure transmissions, 3GPP also defines UP optimisation according to which a device's pre-established connection can be suspended and resumed with less number of control messages [34]. Essentially, the network retains control information about the pre-established connection, which can be used when the device wishes to transmit new data, without the need to setup new SRBs. However, a new DRB still needs to be set up for the transmission of the application data, as this is not retained from the device's previous connection.

2.3.4 Power Saving Techniques

To allow for long battery life of more than 10 years on a single battery charge, NB-IoT uses two power saving techniques:

1. the *extended Discontinuous Reception (eDRX)*, and
2. the *Power Saving Mode (PSM)*

The extended Discontinuous Reception (eDRX) is similar to the DRX used in 4G/5G (section 2.1.3.2.7) with extended periodicities ranging from 20.48s to a maximum value of $\approx$ 175 minutes (table 2.3). As in 4G/5G networks, the length of a DRX/eDRX cycle is usually negotiated between the BS and the device at connection

DRX	eDRX
0.32s	20.48s
0.64s	40.96s
1.28s	81.92s ($\approx$ 1.365 mins)
2.56s	163.84s ($\approx$ 2.73 mins)
5.12s	327.68s ($\approx$ 5.46 mins)
10.24s	655.36s ($\approx$ 10.92 mins)
	1410.72s ($\approx$ 23.52 mins)
	2621.44s ($\approx$ 43.69 mins)
	5242.88s ($\approx$ 87.28 mins)
	10465.76s ($\approx$ 174.4 mins)

Table 2.3: **Extended DRX (eDRX) cycle values:** The table shows the new values in seconds for the eDRX cycle length.

time. However, the BS can unilaterally decide on the DRX/eDRX cycle. Furthermore, DRX/eDRX values are always twice as long as the preceding shorter DRX/eDRX value (e.g. 20.48 sec, then 40.96 sec, then 81.92 sec and so forth, until 10485.76 sec). The shorter the DRX/eDRX cycle the more often the device will wake up, resulting in increased uptime [52]. This is an important consideration for NB-IoT devices, for which the battery life is expected to be more than 10 years. An important thing to note is that eDRX cycles cannot be used in C-DRX.

The Power Saving Mode (PSM) [51] feature (figure 2.26), allows a device to further decrease its battery consumption compared to DRX/eDRX, by entering a deep sleep state, and operate with power consumption close to power-off. When a device uses the PSM feature, the network retains its connection for a period of time in agreement with the device. To enter the PSM mode, the device is required to remain in the idle state for a period of time determined by the T3324 timer (up to $\approx$ 3 hours), during which time the device monitors the paging channel. After the expiration of the T3324 timer, the device moves to the PSM and is not reachable by the network. This phase is called the PSM cycle and its duration is determined by the T3412 timer (up to $\approx$ 413 days). If the device wishes to transmit data sooner than the time agreed with the network, the previously established connection can be used. Otherwise, the network releases the connection and the device needs to establish a new one on its next transmission attempt. At the end of the PSM cycle the device is required to perform a *Tracking Area*

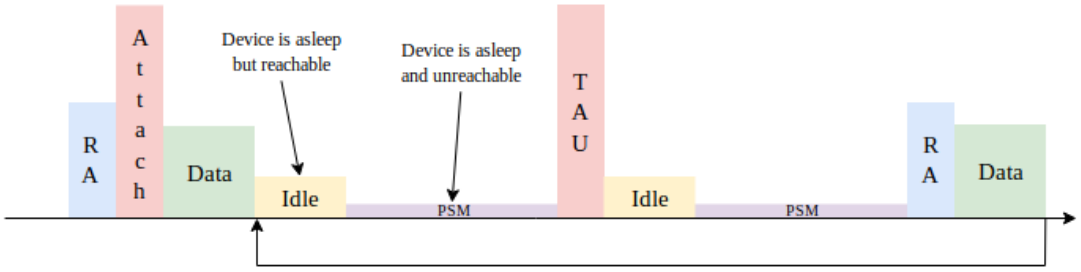


Figure 2.26: **Power Saving Mode (PSM):** The figure depicts the PSM feature. After data transmission the device switches to the idle state for a short period of time determined by the T3324 timer. Upon expiration of T3324, the device starts the PSM cycle that lasts for a period of time determined by the T3412 timer, during which it is not reachable by the network. The maximum duration of a PSM cycle is ≈ 413 days.

Update (TAU). The purpose of the TAU process is two fold. First, it is used to inform the network about the device's current Tracking Area, and second, it is used to indicate that the device is available for a short period of time to receive network originated data (if any). If no data is pending, the device switches back to the idle state, and the process starts again.

2.3.5 Group Communications

Group communications were not initially included in NB-IoT, however the SC-PTM framework was later introduced in NB-IoT Release 14 to support multicast transmission [34]. Upon subscribing to a service, the UE receives a g-RNTI for the subscribed service. Similarly to the eMBMS, the available services for SC-PTM and their control information are periodically broadcasted, using the NPDCCH and NPDSCH. Specifically, control information (session start, session stop, resource allocation, etc.) is carried in the NPDSCH, and is transmitted with a periodicity from 320 ms to 163.84 seconds [34], while the NPDSCH carries the multicast data. For data reception, a generic *Single-Cell Multicast Radio Bearer (SC-MRB)* is established, and UEs receive the multicast content in a similar way as for UE-specific unicast transmission based on the service's g-RNTI.

Chapter 3

Related Work

In this chapter we present an overview of the literature related to the problems discussed in the context of this thesis. For each key challenge, we present the most characteristic works, and show how the research evolved throughout the years. We also discuss shortcomings that drive the need for the novel solutions presented in the following chapters.

3.1 Connection Establishment

When IoT devices started being deployed in large numbers, several surveys [9, 10, 11] identified various major inefficiencies, and highlighted important research directions. One of the major issues that was highlighted in the majority of works was the increased number of collisions in the RACH during the RA process. Therefore, following surveys [53, 54, 55, 56, 57] focused mainly on the RA process, and the impact of IoT traffic on HTC as, at that time, IoT devices were mainly considered of lower priority and importance compared to HTC devices. As the number of IoT devices deployed over cellular networks continued to grow, newer surveys [58, 59, 60] revealed further issues, and identified new research directions. At the same time and due to their tremendously growth, the research community started considering IoT devices to be of equal importance and priority to HTC, and most recent surveys [61, 62, 63] focused on highlighting the problems that affect both types of devices both in 4G and future 5G networks.

Throughout the years, research on the connection establishment process in cellular IoT communications followed many parallel paths, and it can be generally split into three broad categories depending on the underlying issue they address:

1. reducing the signalling load in the EPC,

2. reducing the signalling load in the RAN,

3. reducing the collisions in the RACH

3.1.1 Reduction of Signalling Load in the EPC

Due to the higher capacity of the EPC compared to the RAN, the signalling load in the EPC was not initially considered as a major point of inefficiency. Early works [64, 65] attempted to reduce the signalling load in the EPC by using existing signalling messages to transmit data, thus avoiding the need to setup new signalling and data bearers at each connection. However, although these schemes manage to reduce the signalling overhead for setting up new bearers in the EPC, they are only suitable for short packet transmissions, and revert back to the standard process (section 2.1.3) for larger messages. Additionally, they still use the current RA process, that is ill-suited for periodic transmissions [15, 36]. Later on, the impact of massive IoT devices became apparent in the EPC as well, and later works [66, 67, 68, 69, 70] attempted to aggregate or reuse pre-established bearers of other devices that had similar traffic characteristics (e.g. QoS requirements, data rate). However, even minor differences in the requirements, would either trigger the connection reconfiguration process, or the setup of a bearer from scratch, and the large variance in applications and devices resulted in reduced re-usability of bearers, thus these schemes did not yield significant savings.

3.1.2 Reduction of Signalling Load in the RAN

The signalling load in the RAN was one of the first points of inefficiency to be identified, especially due to the limited capacity compared to EPC, and it still remains as such. Works in this area [71, 64, 65, 66, 72, 73, 74, 75, 76, 77, 78, 79] proposed using messages of the RA process to transmit short packets, avoiding the need to setup communication bearers with the BS at each connection. However, these schemes are inherently insecure, as no security context has been established at the time of transmission of those messages. Furthermore, some works [78, 79] are only suitable for infrequent and non-periodic transmissions, such as those generated by HTC devices, as they lead to an excessive number of collisions otherwise.

3.1.3 Collision Reduction in the RACH

The number of collisions in the RACH was, and still is considered the major inefficiency in supporting massive numbers of devices [53, 54, 55, 56, 57], mainly due to the limited number of preambles available for the RA process. As such, it has received the majority of attention throughout the years, and research on this area has followed multiple orthogonal directions.

Grouping-based approaches: These approaches attempt to reduce the collisions in the RACH by grouping devices together, and controlling when and/or how the groups would access the network. Early works [80, 81, 82] attempted to decrease collisions by scheduling resources among the different groups and page the devices according to their group, requesting them to transmit data only then, thus completely disregarding the device's periodicity. Later works [83, 84] required devices to form groups by themselves and elect a group leader, which was the only member of the group allowed to connect to the network and transmit the data of all the devices in its group. These approaches assumed some form of short-distance communication technology (e.g. Bluetooth), which might be beyond the capabilities of some IoT devices. Finally, more recent works [85, 86, 87, 88, 89, 90, 91] grouped devices based on different criteria, such as the QoS requirements or their physical location. The network would then allocated resources to the groups based on their size and grouping criteria.

Access barring schemes: *Access Class Barring (ACB)* and *Enhanced Access Barring (EAB)* schemes are other types of grouping that attempt to decrease the collisions in the RACH, by placing devices in different priority classes, controlled by different access probabilities (APs). Network access is then controlled by adapting the APs. These schemes are based on an early study by 3GPP on RAN improvements for IoT devices [64], and several ACB schemes were initially proposed [92, 93, 94, 95]. However, these schemes used static barring parameters, and it soon became apparent that this rigid approach had a significant impact on the network access delay for the devices belonging to classes with low priorities. Therefore, subsequent works proposed the EAB schemes that could dynamically adapt the barring parameters, to find a balance between the number of collisions and the network access delay experienced [96, 97, 98, 99, 100, 101, 102, 103]. These schemes are complimentary to our work and can in fact be implemented on top of ASPIS.

Preamble splitting schemes: Initially, IoT devices were not considered as important as HTC devices and some works focused on splitting the preambles between the

two categories, mainly allocating more preambles to the HTC devices, in an attempt to minimise the impact of IoT traffic on HTC. Very few works (e.g. [104]) used static splits, but it was shown that such approaches did not provide sufficient flexibility and the research direction soon shifted towards more adaptive schemes [101, 105, 106, 107, 108, 109]. Despite their dynamic approach, these schemes limit the access opportunities for IoT devices by misallocating the available preambles, thus reducing the performance of the overall system under load. Furthermore, they mainly depend on observed network conditions and historical information, thus adapting the preamble splits reactively. In contrast, ASPIS features a proactive preamble splitting technique which can predict RACH congestion before it occurs, and optimally allocate the preambles between HTC and IoT devices.

Collision resolution and Non-Orthogonal Multiple Access (NOMA) schemes:

The majority of the research avenues attempted to address collisions before they happened, and used the existing collision resolution mechanism of the RA process to tackle them when they occurred, based on the use of orthogonal preamble transmissions.

However, recently and mainly due to the increased demands for network connectivity, research began following a different direction that was based on non-orthogonal transmissions, called *Non-Orthogonal Multiple Access (NOMA)*. Although different NOMA schemes have been developed (e.g. *Sparse Code Multiple Access (SCMA)*, *Low Density Spreading (LDS)*), they are all based on the same principle that more than one non-orthogonal signals can be transmitted on the same resource, and the receiver is responsible for separating the different signals. NOMA has been shown to improve spectral efficiency in cellular networks [110], and it can support massive connectivity, a key enabler for IoT. As such, it has been included in the 5G design.

With the collisions in the RACH being the major bottleneck in cellular networks, NOMA schemes were also extended to the RA process, for what is known as *Non-Orthogonal Random Access (NORA)*. Essentially, NORA schemes allow the transmission of non-orthogonal preambles, and attempt to resolve the collisions after they have happened in order to allow most or all of the collided devices to proceed with their connections. In one of the first proposals [111] the authors exploit the TA value of the RA process to distinguish between collided devices in the time domain, while a few years later the authors of [112] used a collision resolution algorithm based on splitting trees. In [113, 114] the modulation codes and signal spreading are combined so that new bits are mapped to a multi-dimensional codeword of an SCMA codebook set. Other

works [115, 116] use patterns to define the mapping of transmitted data to a resource group that can consist of time, frequency, and spatial resources or any combination of these resources.

A key component of the NOMA schemes is the techniques used to separate the non-orthogonal signals (e.g. superposition coding, message passing algorithm[1]). One of the most prominent schemes is the *Successive Interference Cancellation (SIC)* [117], according to which the different signals (in the case of the RA process, the signals are the preambles) are decoded successively. Essentially, when SIC is applied, the strongest signal is decoded first, while all others are treated as interference. The decoded signal is then subtracted from the originally received signal, and the process repeats until no more distinct signals can be decoded. Therefore, similar preambles can be decoded with SIC if they are significantly apart in the power domain to treat one as interference while decoding the other. Subsequent works [118, 119, 120, 121] built on top of [117] to further enhance the performance of the RA process in terms of the number of collisions and network access delay.

3.2 Network Resource Utilisation

3.2.1 Group Communications and Multicasting

The MBMS and eMBMS frameworks were designed mainly for multimedia services such as video streaming. As such, the majority of research focused on enhancing the QoS of the multimedia transmission and the user satisfaction, with different modulation schemes, resource allocation and grouping proposals. However, to the best of our knowledge, very limited research has focused on the multicast transmission functionality, energy consumption and network usage for IoT devices.

Initial proposals attempted to increase the throughput of multicasting transmissions by adapting the MCS used [122, 123, 124, 125]. However, as the number of devices using multicast services increased, these schemes proved insufficient to provide satisfactory services without affecting the unicast traffic in the cell, or introducing significant processing overhead at the BS.

In the following years, different parallel approaches were explored that mainly focused on enhancing the experienced QoS of the users. In [126] the authors distinguish

[1] Although these schemes were invented several years ago, their use in cellular networks is new in 5G, and is not present in 4G.

between bandwidth intensive streaming services and file delivery, while in [125, 127] devices are split into groups based on the received services. They are then further split into subgroups based on the MCS that they can use, which is then dynamically adapted to yield the best result. In the same period, the authors of [128] placed users in groups based on the channel quality feedback they provided. Similar approaches were proposed during the following years with several works grouping the devices based on the experienced QoS [129, 130, 131, 132, 133]. Research on group communications is continuing on for the 5G networks, as it is expected that multicast functionality will be the basis of a lot of different applications, some of which will be mission critical [132, 134, 135, 136, 137].

In parallel to the aforementioned approaches, other works on multicasting in cellular networks have focused on analysing the effects of resource allocation for multicast transmissions over the background (unicast) traffic, as well as reducing the transmission latency. In [138] the author discusses the resource sharing between HTC and IoT devices, when IoT groups receiving unicast data share the same resources with HTC devices receiving multicast data. The authors of [139] propose a resource allocation scheme for multicast transmissions to reduce the transmission latency and the effect on existing unicast traffic, while in [140] the scheduling of the resources used for multicast transmissions is based on the *Signal-to-Noise Ratio (SNR)*. However, these works do not take into account the different DRX/eDRX cycles of the devices, nor do the consider the energy consumption and battery limitations of IoT devices. In [141] the authors propose a scheduling scheme for 5G networks based on the users' interests. However, this scheme allocates different subcarriers to different multicast groups, and thus it is not applicable in bandwidth restricted technologies, such as NB-IoT.

Focusing specifically on the multicast transmission latency, several works [142, 143, 144, 145, 146, 147, 148, 149] proposed the idea of content caching, according to which the multicast data is pre-saved on a cellular network entity before being distributed to the devices. Finally, in [150] the authors study the effects of the increasing number of devices in a cell and utilise the newly introduced non-orthogonal transmission schemes to examine the performance of the system.

3.2.2 DRX Adaptation Techniques

DRX adaption techniques have recently received a lot of focus, mainly due to the fact that the DRX is highly associated with the energy consumption, which is in turn of

major importance to IoT devices. The works of [52, 151] present an analysis on the average energy consumption and latency in 4G systems using different DRX cycle values. Other works [152, 153, 154, 155, 156] attempt to adapt the DRX cycles to achieve various results, such as energy consumption minimisation, or preservation of the QoS of the received service. However, this adaptation is done for each device independently thus adding significant processing overhead to the network. In [152, 157] different DRX adaptation methods are presented that aim to minimise the energy consumption while guaranteeing the QoS of the multicast service being received. Similarly, [153, 154] attempt to fine-tune the DRX configuration to result in optimised energy consumption. In [155, 156] the authors present enhancements to the existing DRX mechanism that compromise the QoS in favor of the energy consumption. Finally, several works [158, 159, 160, 161, 162] analysed the trade-off between energy consumption and delay, and attempted to strike a balance among them by adapting the DRX accordingly.

3.3 Energy Consumption

Energy conservation both on the device as well as the network side was identified as a major goal for current and future cellular networks, since the beginning of the IoT era. Several works [163, 164, 165, 166] highlighted open issues and discussed several research directions, while other works [167, 168] attempted to analyse the proposed energy saving practises and approaches, and identify best candidates based on the expectations on the evolution of both the network and the devices.

Initially, the majority of research on energy consumption focused on the network side. The goal was to reduce the energy consumption of the BS, which was expected to significantly increase due to the large number of devices they now had to serve. Different parallel research directions were followed, which can be broadly split into the following categories.

Resource allocation schemes: These proposals [167, 168, 169, 159] discuss different transmission strategies and resource allocation schemes based on the transmission patterns of the devices (either HTC or IoT) and the overall traffic load in the cell.

BS transmission parameter optimisation: These approaches [164, 170, 171, 172, 173] try to dynamically adapt the transmission and operation parameters of the BSs, to decrease their energy consumption using on/off switching and discontinuous transmission schemes.

Interference mitigation schemes: These proposals attempt to decrease the energy consumption of BSs by mitigating interference among different transmissions and avoid retransmissions of previous messages [174, 175, 176, 177].

As the number of devices started to increase, the research community started focusing more on the energy consumption of the devices, and different parallel directions were followed. A lot of works [157, 178, 179, 160, 180, 155, 159, 175, 181] attempted to optimise the DRX parameters of IoT devices in order to increase their sleep period, and thus decrease the energy consumption. However, the energy consumed during a DRX cycle is orders of magnitude lower than that of the RA or Attach processes, and thus optimising the currently used protocols is equally important. Other works (e.g., [178, 179, 180]) use the DRX settings, as well as information (e.g., device capabilities, battery level), as a basis to efficiently schedule data transmissions in order to achieve low power usage. Although these approaches can synchronise when the device checks for network-originated data with the times it needs to send its own data, they do not optimise the currently used procedures that incur a significant energy cost

Due to the large number of messages exchanged (either control or data), several works (e.g., [182, 183, 184, 185, 186, 187, 188]) attempted to optimise the transmission parameters to decrease the energy consumption of IoT devices while in the working state, by tuning parameters such as the duty cycles or the number of transmissions [182] or optimising the resource allocation and data transmission parameters (e.g. the data rate) [183, 184]. While such changes can have a positive impact on the battery life, the major inefficiencies stem from the fact that the protocols used were initially designed for HTC devices, that present significantly different traffic characteristics. As such, they are ill-suited for IoT devices.

Until recently, works on energy consumption did not focus on a specific cellular network technology, and thus the individual characteristics of the devices were not taken into account. Some research works focused on providing general models of how the energy is consumed in IoT devices [189, 190, 191, 192] Although such modeling can provide useful insights, the communication technology used and the specific characteristics of the devices play a more important role in the total energy consumption. Therefore, subsequent works [188, 191, 193, 194, 195, 21, 196] discuss NB-IoT technology in terms of energy consumption, analyzing the different modes of operation and their associated energy cost. However, they only measure the transmission and reception operations. The work of [196] provides an experimental study on the energy consumption of a commercial, off-the-shelf, NB-IoT modules for the basic communi-

cation operations. In [21] the authors provide an empirical estimation of the battery lifetime of NB-IoT devices with different transmission power levels.

Some recent works [21, 20] attempt to model and experimentally measure the energy consumption of NB-IoT devices. However, they differ to our work in terms of scope and granularity of the measurements. Both [21, 20] focus on measuring the energy consumption of data transmission/reception as a function of different network configurations (data rates, etc.) In contrast, our work additionally measures the energy that the devices need to spend for operations that facilitate these data exchanges (encryption, communication with the core network, active waiting etc.), which gives a better estimate of the overall energy consumption of the device. Also crucially, [21, 20] are limited to modeling the energy consumption and do not assess the impact and necessity of each individual operation, nor do they propose improvements. Instead, we examine which processes contribute the most to the energy consumption, and propose appropriate protocol optimisations and best practices aimed at lowering the corresponding components of device energy consumption.

Chapter 4

Connection Establishment for IoT Devices in 4G Networks

In this chapter we focus on the challenges during the connection establishment process that arise when large numbers of devices attempt to transmit data simultaneously. Given the often periodic and synchronised transmissions of IoT devices, such signalling storms are not a theoretical concern, but have been observed in real networks [15, 16]. This work predates the finalisation of the 5G standard, and as such it was targeting 4G networks. Note, however, that a similar approach has since been standardised for 5G.

Previous approaches for IoT support over LTE networks focus on only one of the three major parts of the connection establishment: (i) reducing the number of collisions in the RACH; (ii) reducing the signalling load in the EPC; or (iii) reducing the signalling load in the RAN. In contrast, we proposed a novel mechanism, called ASPIS, which is able to simultaneously address all three issues. Our approach relies on extending the previous RRC states in LTE (idle and connected), with a novel intermediate state, that allows IoT devices to partly retain their communication bearers between transmissions, significantly reducing the signalling overhead in the EPC. ASPIS also uses a modified RA process with fewer messages, which is specifically designed for IoT devices, to reduce the signalling load in the RAN and the total access delay. Furthermore, ASPIS can efficiently support short packet transmissions (< 80 bytes) for which RAN bearers may not be needed. Finally, ASPIS uses a proactive, dynamic preamble splitting scheme that finds an optimal split between preambles used by HTC and IoT devices, by exploiting the periodicity of IoT devices to predict future RACH congestion before it happens, thereby lowering RACH collisions. Please

note that preamble splitting is aligned with the direction taken by 3GPP [64] for LTE networks, However, preamble splitting schemes at the time of publication were either static or reactive.

ASPIS can be supported by existing LTE networks without hardware changes, requiring only a software update, and can be incrementally deployed alongside legacy IoT/HTC devices and eNBs. To assess its performance, we evaluated ASPIS via a combination of small-scale evaluations with a prototype implementation (section 4.2), and large-scale evaluations with thousands of devices (section 4.3), using a custom simulator based on realistic traffic patterns [16].

The work presented in this chapter has been published in the 2017 Mobile Ad-Hoc and Sensor Systems (MASS) conference [197].

4.1 ASPIS

In this section we present ASPIS, our novel mechanism that addresses the congestion during the connection establishment process, i.e. the increased number of collisions in the RACH, and the signalling load in the RAN and EPC, of IoT devices in 4G networks. The core of ASPIS is the intermediate RRC state (sec. 4.1.1) that IoT devices switch to, instead of idle (sec. 4.1.2), after a period of inactivity. The intermediate state preserves part of the devices previously established connection in the EPC, and reuses it each time that device wishes to transmit new data, thus reducing the signalling overhead in the EPC, as well as the access delay.

To reduce the signalling load in the RAN, we introduce a new, shorter RA procedure to be used in conjunction with the intermediate state, that reduces the signalling load in the RAN. The new RA process (sec. 4.1.2) requires fewer messages compared to the standard RA process, and also provisions for short packet transmissions (< 80 bytes) for which SRBs and DRBs are not needed. Finally, to reduce the collisions in the RACH, we introduce a new, proactive preamble split scheme that exploits the periodic nature of IoT devices, to predict future RACH congestion before it happens, and allocates preambles to HTC and IoT devices, so that the RACH collisions are minimised. We now describe the intermediate state, the procedures to switch to and from it, as well as the proactive preamble split scheme in detail.

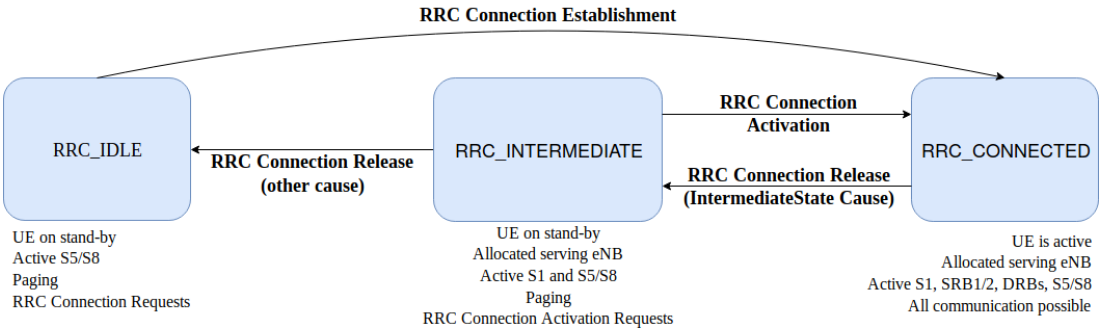


Figure 4.1: **ASPIS RRC state diagram:** The figure depicts the new RRC state diagram, with the existing states and their characteristics. A UE in the intermediate state is allocated an eNB, but does not have an active connection with it. However, the connection to the core network still exists in an active state, and the core network is unaware of the device being in the intermediate state.

4.1.1 RRC Intermediate State

In this section we present the intermediate state, which combines some characteristics of the two existing states in LTE networks, and allows IoT devices to remain in a semi-connected mode. As with the idle state (section 2.1.3.1), devices in the intermediate state do not have dedicated RAN resources, and cannot directly transmit scheduling requests. However, they can receive paging requests and connection reconfiguration messages, as well as MAC alignment control commands. Furthermore, devices in the intermediate state remain registered with their MME and S-GW. All previously established EPC bearers still exist and can be used immediately after the device switches back to the connected state, resulting in decreased signalling load in the EPC. Essentially, the intermediate state is only visible to the eNB.

In order to facilitate transitions between the different RRC states (figure 4.1), we introduce two additional inactivity timers. The first one is used to move a device from the connected to the intermediate state, while the second one switches the device from the intermediate to the idle state. Similar to the inactivity timer in standard LTE, these timers are set by the network operator and can be chosen to strike a balance between required resources and signalling load (sec. 4.3).

4.1.2 RRC State Transition Procedures

In this section we describe the procedures that the IoT devices need to follow to transition between the three RRC states: connected, intermediate and idle (figure 4.1). We

also discuss how our procedures result in reduced signalling load both in the EPC and the RAN. The complete procedure is depicted in figure 4.2.

4.1.2.1 Transition from CONNECTED to INTERMEDIATE

After a period of inactivity in the connected state (controlled by our first inactivity timer), the eNB releases the connection, frees any allocated resources, and moves the device to the intermediate state. To do so, the eNB sends a RRC Connection Release message with the release cause set to the new value of *IntermediateState*. This instructs the device to tear down its DRB and all its SRB bearers in the RAN, and moves it to the intermediate state. This allows ASPIS-capable devices to be supported even by legacy eNBs. The eNB preserves all the bearers of the device in the core network (S1, S5/S8). Note that the MME and S-GW are never informed about the new state of the device, thus avoiding unnecessary load in the EPC. For eNBs without ASPIS support, the current release cause value is used, switching the device to the idle state, and the default RA process is followed on the device's next connection.

When switched to the intermediate state, the device keeps its security context that was established during the AKA procedure (section 2.1.3), to be reused once it switches back to the connected state. Therefore, the AKA process happens only once, and does not incur any additional overhead for the subsequent transitions to the connected state.

4.1.2.2 Transition from INTERMEDIATE to IDLE

When a device remains inactive in the intermediate state for a period of time specified by our second inactivity timer, the eNB releases its EPC connection, in order to reduce the number of unused resources, and avoid potential DoS attacks. To accomplish that, the eNB follows the standard Connection Release procedure (section 2.1.3) to request the release of the device's connection in EPC.

4.1.2.3 Transition from INTERMEDIATE to CONNECTED

When a device in the intermediate state wishes to transmit new data, it first needs to transition to the connected state, following our new RA process. In the current procedure, preambles for contention-based RA are split in two groups (groupA and groupB) [26]. We introduce a new preamble group (groupC) that is only used by devices in the intermediate state. The use of a preamble from that group indicates that

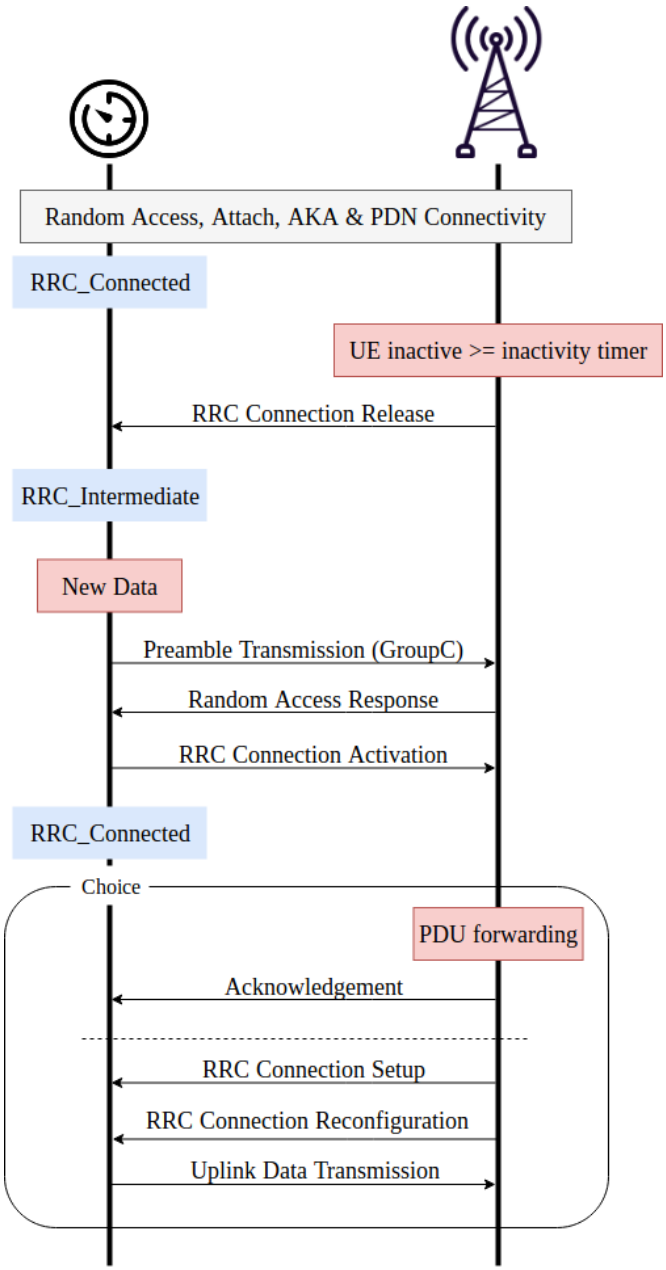


Figure 4.2: **ASPIS mechanism:** The figure depicts the complete procedure of ASPIS with all the different cases it handles. The devices switch to the Intermediate state after some time of inactivity. From there, they can quickly switch back to the Connected state to transmit new data, using any of the group C preambles. Depending on the size of the message they want to transmit, they can directly send it encapsulated in their Connection Activation message, otherwise the eNB sets up new RAN bearers to accommodate the transmission.

the device has previously registered with and connected to the network, and has an established EPC bearer. Similarly to groups A and B, the allocation of preambles in groupC is done by the eNB (sec. 4.1.2.4).

The transition procedure is as follows. Initially, the device chooses a random preamble from groupC and transmits it in the RACH. Similarly to the standard RA procedure (sec. 2.1.3), the eNB replies with a RAR message for every identified preamble, and the device then replies with a RRC Connection Activation message. This is a newly introduced message that contains the KSI (Key Set Identifier) of the device-specific security context (that was established during the initial AKA procedure), the S-TMSI of the device, a NAS PDU container, an indicator of whether a small NAS PDU message is encapsulated in the NAS PDU container, the APN of the PDUs destination, and the devices EPC bearer-ID. The NAS PDU indicator, NAS PDU, EPC bearer-ID and APN are encrypted with the devices specific security keys. Based on the preamble used in the first step, the eNB knows that the connection activation message was transmitted by a device in the intermediate state, and therefore, it uses the 1-1 map between the device and its KSI along with the S-TMSI to uniquely identify it and authenticate the received message. In case of authentication failure, or if the eNB has released the device's connection due to the expiration of the second inactivity timer, the eNB discards the packet and instructs the device to follow the standard RA process.

The NAS PDU indicator informs the eNB whether a short packet ($<$ 80 bytes) is encapsulated in the NAS PDU container. If the NAS PDU indicator is set to false (i.e., the device wants to transmit a message larger than 80 bytes), the eNB needs to set up RAN bearers. This is done with a RRC Connection Setup message followed by the RRC Connection Reconfiguration message, that sets up the default DRB and SRBs for the device. The device can then use its existing EPC bearer and start transmitting its data. If the indicator is set to true, the eNB does not need to set up RAN bearers for the device. Instead, the eNB contacts the MME to identify the device's allocated S-GW. It then extracts the PDU, encapsulates it in a GTP packet and forwards it to the S-GW using the device's existing bearer. Having to set up RAN bearers for small packets introduces a signalling load disproportional to the actual size of the message. Therefore, our provision for short data transmission further decreases the signalling load in the RAN. The inspiration for this enhancement is based on the *Small Data Transmission (SDT)* [198, 199] procedure of 3GPP. However, unlike ASPIS, the SDT procedure requires the establishment of an RRC connection before any data transmission occurs.

Since all devices in the intermediate state share the same preambles (groupC), colli-

sions may occur. When a device does not receive a RAR within a pre-defined period of time (similarly to the current RA procedure), it backs off and tries again at a later time. If the device fails to re-activate its connection after a number of attempts specified by the network operator, it switches back to the idle state and attempts the conventional RA process. However, in this case the device needs to inform the eNB about its active EPC bearer when it connects again, using the RRC Connection Setup Complete message, to prevent the eNB and EPC from creating a new bearer, and reduce the number of stalling EPC bearers.

4.1.2.4 Proactive Dynamic Preamble Split

So far we have presented the new intermediate state and RA process, that reduce the signalling load in the EPC and RAN. Here, we present our proactive preamble split mechanism that exploits the periodicity of IoT devices to reduce the number of collisions in the RACH. Intuitively, our proactive preamble split mechanism predicts the number of IoT devices that are likely to transmit in each frame, and proactively adapts the number of preambles allocated between devices in the intermediate state (groupC), and any devices in the idle state (groupA and groupB), in order to minimise the overall number of collisions.

The idea behind this enhancement is that the majority of IoT devices transmit data on predefined intervals specified by their periodicity. Based on the time of their last transmission and the periodicity of each device, we can predict how many devices are expected to transmit in any given frame. This allows us to optimally split the preambles in advance, so that we do not needlessly over-allocate preambles to any group. The dynamic preamble split is broadcasted in the SIB2, similar to the split between preambles in groups A and B.

The eNB learns the periodicity of the devices using the *Capabilities Enquiry* message, which is transmitted as part of the Attach process. This can be accomplished with a new UE-category value (e.g. 13) to indicate that the device supports ASPIS. The device indicates its periodicity in milliseconds in the Indicator-31 of the *featureGroupIndicators* field in the capabilities message.

We pose our proactive preamble allocation mechanism as an optimisation problem. Let R be the total number of preambles available for contention-based access, r_M be the number of preambles allocated for devices in the intermediate state, and n_t be the number of such devices expected to transmit in frame t. The expected number of colliding IoT devices in the intermediate state is:

$$E_{n_t,r_M} = n_t(1-(1-\frac{1}{r_M})^{n_t-1}) \tag{4.1}$$

Similarly, let n'_t be the number of idle devices that transmit in frame t. Since we cannot predict when these devices will transmit, we can approximate n'_t as the running average over a number of previous frames. The expected number of collisions of these devices can be expressed as:

$$E_{n'_t,R-r_M} = n'_t(1-(1-\frac{1}{R-r_M})^{n'_t-1}) \tag{4.2}$$

As the total number of contention-based preambles is fixed, increasing the preambles for devices in the intermediate state decreases the available preambles for idle devices. Ideally, we would like to minimise the total number of collisions for all devices. Additionally, we would like to guarantee a minimum number r_{min} of preambles given to idle devices, as their expected transmissions are only an approximation based on past frames. In other words, we want to minimise:

$$\arg\min_r(E(n_t,r_M))+E(n'_t,R-r_M), \quad s.t.\ r_M R \leq r_{min} \tag{4.3}$$

Note that $E(n_t,r_M)$ is monotonically increasing while $E(n'_t,Rr_M)$ is monotonically decreasing as r_M increases, so eq. 4.3 is convex. As such, minimising eq. 4.3 can be done efficiently with a simple modification of the binary search algorithm. Furthermore, the total number of preambles R is typically very small so the overall computation time is negligible for the eNB.

Due to the use of the proactive preamble allocation, ASPIS is able to scale gracefully. If there are no devices with active connections, the system will not waste resources on them, and it will naturally fall back to the current behaviour of allocating all preambles to devices in the idle state.

4.2 Prototype Implementation and Experimental Results

An attractive aspect of ASPIS is the ease of realisation in the context of current and emerging cellular network standards. ASPIS can be implemented via software updates to eNB and devices, and does not require hardware changes. More specifically, implementing ASPIS involves changes to the RRC and PHY layers at the eNB and the device. In the RRC layer, ASPIS requires the introduction of the new intermediate state, two additional inactivity timers, the RRC Connection Activation message, and a

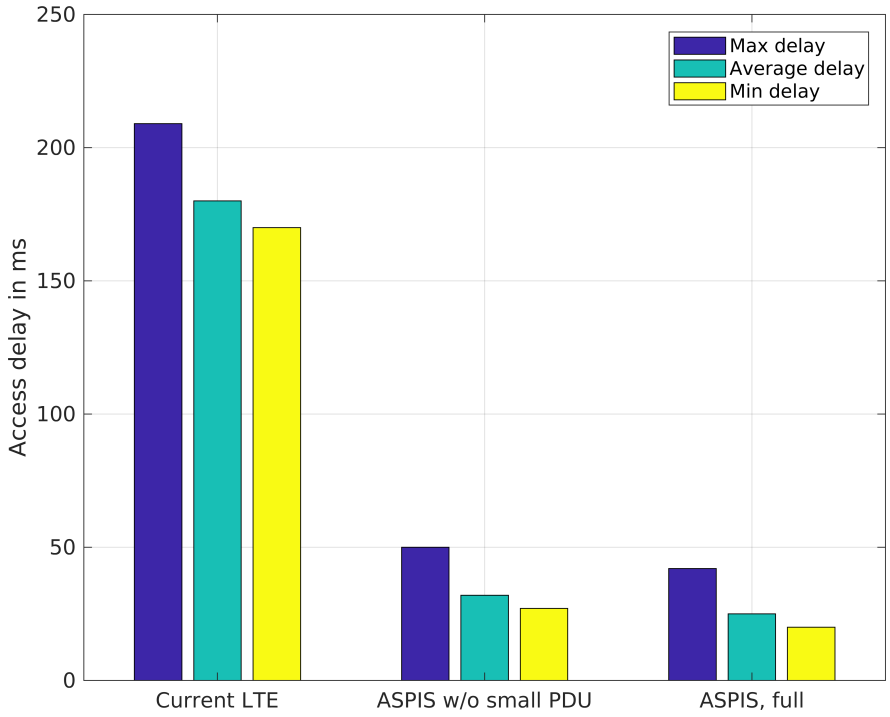


Figure 4.3: **Network access delay of ASPIS, compared to LTE specifications, implemented over OAI:** The figure depicts a comparison between variants of ASPIS and the standard LTE, in a setting with 10 ASPIS-enabled devices implemented over OAI. We can see that ASPIS significantly outperforms the current approach of LTE networks without the small PDU enhancement. As expected, the use of the small PDU enhancement results in lower access delay, however the difference to the APSIS without the small PDU enhancement is small, due to the overall shorter connection activation procedure.

new release cause. Minimal changes at the PHY layer are required for the proactive preamble splitting technique so that the eNB can update the SIB2 about the allocation of the preambles to the different groups. A noteworthy aspect of ASPIS is that it can be incrementally deployed, i.e. legacy devices can connect to ASPIS-enabled eNBs, and ASPIS-enabled IoT devices can still connect to legacy eNBs (sec. 4.1.2).

We have developed a prototype implementation over the OpenAirInterface (OAI) platform [200], a well-known open source software implementation of the LTE RAN and EPC components. Doing so was straightforward and essentially involved making the changes outlined above. We have used this prototype implementation of ASPIS for experimental evaluation in small-scale settings (10 ASPIS-enabled IoT devices), considering various different metrics. As signalling load results are similar to those obtained via simulations (section 4.3), we only present a sample experimental on ac-

cess delay based on the implementation. 3GPP defines access delay as the time elapsed from the transmission of the first preamble until the receipt of the RRC Connection Setup message [34] (figure 2.15). As we alter the existing RA process, we instead measure the access delay as the time passed from the first preamble transmission until the start of the actual data transmission. We used 10 ASPIS-enabled devices associated with an eNB implemented over OAI. Figure 4.3 shows the average access delay of ASPIS in comparison with standard LTE specifications. ASPIS-enabled devices present lower network access delay with and without the optimisation for small PDUs. This is expected as the device requires fewer messages to switch to the connected state from the intermediate state rather than the idle.

4.3 Large Scale Evaluation via Simulations

4.3.1 Experimental Setup

While the ASPIS implementation over OAI demonstrates its practicality, OAI cannot support large number of devices. To assess ASPIS with respect to the key metrics of interest (signalling load in the EPC/RAN and the number of collisions in the RACH during the preamble transmission), in scenarios with thousands of IoT devices, we developed a custom simulator in Matlab, that models both standard LTE procedures, as well as the ASPIS mechanism.

More specifically, our simulator implemented the PDCP, RLC and RRC layers of the cellular network protocol stack. For the standard LTE procedures, the existing RA and Attach messages were implemented according to the asn specifications [201]. For the ASPIS mechanism, we implemented the new RRC Connection Activation and RRC Connection De-Activation messages according to the asn specifications for RRC messages [201]. Furthermore, we implemented the new procedures at the eNB and the core side to support the new intermediate state, i.e. the retaining of the UEs connection to the core network when its connection is de-activated by the eNB, and the UEs connection release both at the eNB and core network, when the inactivity timer 2 is reached. Finally, we simulated at least 18000 LTE frames, assuming a RACH instance in each of them.

For our simulations, we used realistic traffic models based on [16] which, at the time of publication, was the largest publicly available study on IoT traffic patterns, with 1000 HTC devices per cell and a varying number of IoT devices, ranging from

500 to 4000, randomly distributed within the coverage area of the cell. All simulations included different types of IoT devices, with 50% of them transmitting small PDUs (up to 80 bytes), which accounts for the 41% of the total IoT traffic. We set our inactivity timer 1 to 2.8 seconds, based on the average session inter-arrival and length of IoT devices [16], while the inactivity timer 2 was set to 90 seconds. Please note that each data point in the plots is an average obtained from 10 repetitions for that data point.

Similarly to the RA process followed in LTE networks, we used 56 preambles in total. For the standard LTE [64], we allocate 12 preambles for IoT devices, and 44 preambles for HTC devices, while the split between the HTC and the IoT devices for ASPIS was decided using the proactive split mechanism (sec. 4.1.2.4). In addition, the maximum number of collisions that a device could experience before declaring a network outage was set to 3, similarly to the standard LTE. To ensure the correct implementation of our simulator, we validated it against the ASPIS OAI implementation in small-scale settings where we implemented the same changes in terms of the new RRC message, the new state, and the new procedures. We compared the signalling load for the same number of devices, and we obtained similar results

We compare ASPIS against the current LTE specifications [29] which we used as our baseline, the SDT proposal of 3GGP [198, 199], and the works of [79, 66]. The SDT proposal [198, 199] avoids setting up EPC bearers for small packets, but still requires the establishment of a connection in the RAN. Maldonado et al. [79] present an alternative RA process for small packet transmission, however, devices still need to go through the complete RA and Attach processes for larger packets. Finally, in [66] the authors use the authentication messages of the AKA procedure to securely transmit short messages. This approach uses the existing RA without any modifications, and as such it requires the completion of the Attach process when larger packets need to be transmitted. We do not compare against the works like [71, 74], as these use the preamble transmission or RRC Connection Request messages of the RA process to transmit data, which is inherently insecure (sec. 3.1).

4.3.2 Results

4.3.2.1 Load in EPC

Figure 4.4 depicts the signalling load in the EPC for the five compared approaches. For clarity, we normalised the signaling load by the total number of simulated frames. Our results show that ASPIS significantly outperforms the LTE connection establishment

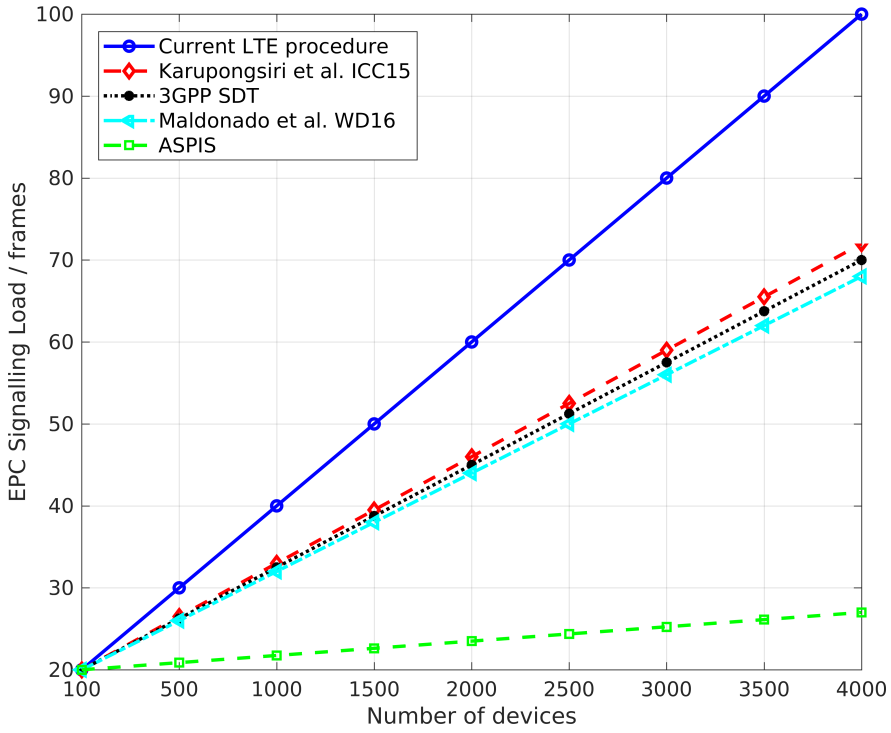


Figure 4.4: **EPC signalling load of ASPIS, compared to LTE specifications and other proposals:** The figure depicts the EPC signalling load as a function of the number of devices, for ASPIS and alternative approaches. We can see that the ASPIS mechanism significantly outperforms both the mechanism used in current LTE networks, the SDT proposal [198, 199], as well as other proposals [79, 66]

process [29], the SDT proposal [198, 199], as well as the schemes of [79, 66], as the focus of most of these alternative approaches is only on the transmission of small packets that can be contained in existing RRC messages. As such, they still have the set up bearers in the core network for larger packets, thus increasing the overall signalling load. ASPIS provisions for both small (up to 80 bytes) and larger messages, thus alleviating the signalling load in both cases. Key contributor to the savings with ASPIS is the use of the intermediate RRC state that gives the illusion to the EPC that the device is connected even though it may be dormant between periodic transmissions.

4.3.2.2 Load in the RAN

We then assess the signalling load of ASPIS in the RAN against the current RA procedure [29], and the work of [79] (figure 4.5). Recall that [198, 199] and [66] use the conventional RA procedure, so the load in the RAN is identical to that of the current LTE procedure. Our results show that ASPIS significantly decreases the signaling

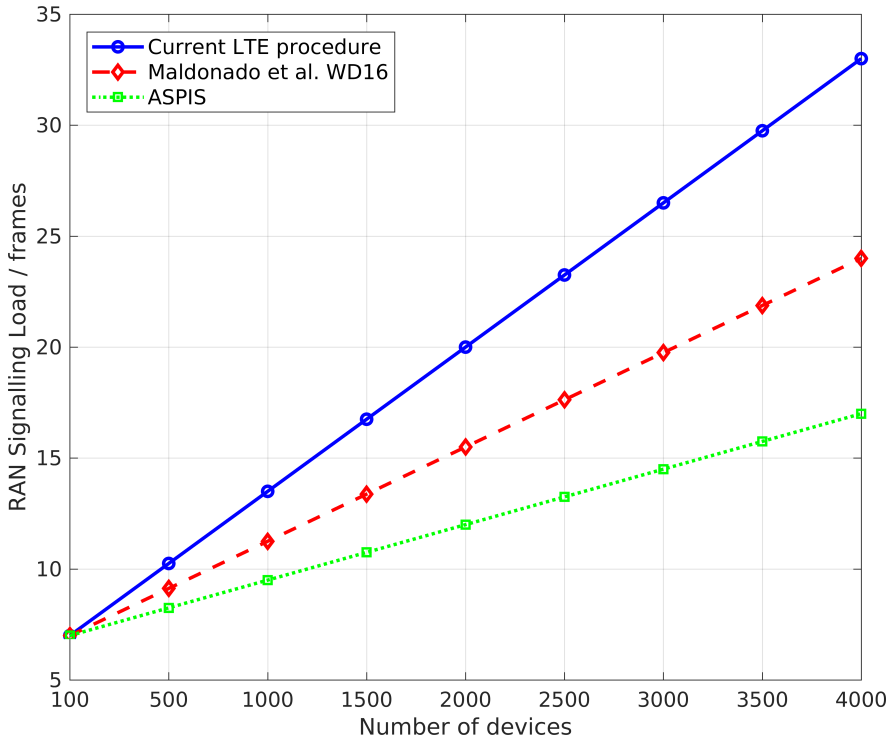


Figure 4.5: **RAN signalling load of ASPIS, compared to LTE specifications and other proposals:** The figure depicts the RAN signalling load as a function of the number of devices with ASPIS and alternative approaches. Once again, our approach outperforms the approach currently used by LTE networks as well as the work of [79]

load in the RAN, compared to standard LTE and the mechanism of [79] due to the fewer messages that it requires, as the alternative mechanisms either do not account for IoT traffic, or only alleviate load for small PDU messages [79]. In contrast to these schemes, ASPIS decreases the signalling load for all connections, regardless of the size of the data being transmitted.

4.3.2.2.1 Small PDUs vs Larger PDUs To better appreciate the benefit of the provision for the small PDUs, we compare the signalling load of ASPIS in the RAN and EPC, with the provision for the small PDUs transmission (shown as 'full') and without it (figures 4.6, 4.7). The current RA procedure is included in the comparison for reference. One interesting thing to notice is that in the EPC, the signalling load introduced when small packets are transmitted is marginally greater than when ASPIS does not provision for small PDUs. This is because the transmission of small PDUs uses an extra signalling message in the EPC to forward the PDU (section 4.1.2). However, both our approaches result in significant gains in terms of the signalling load compared to

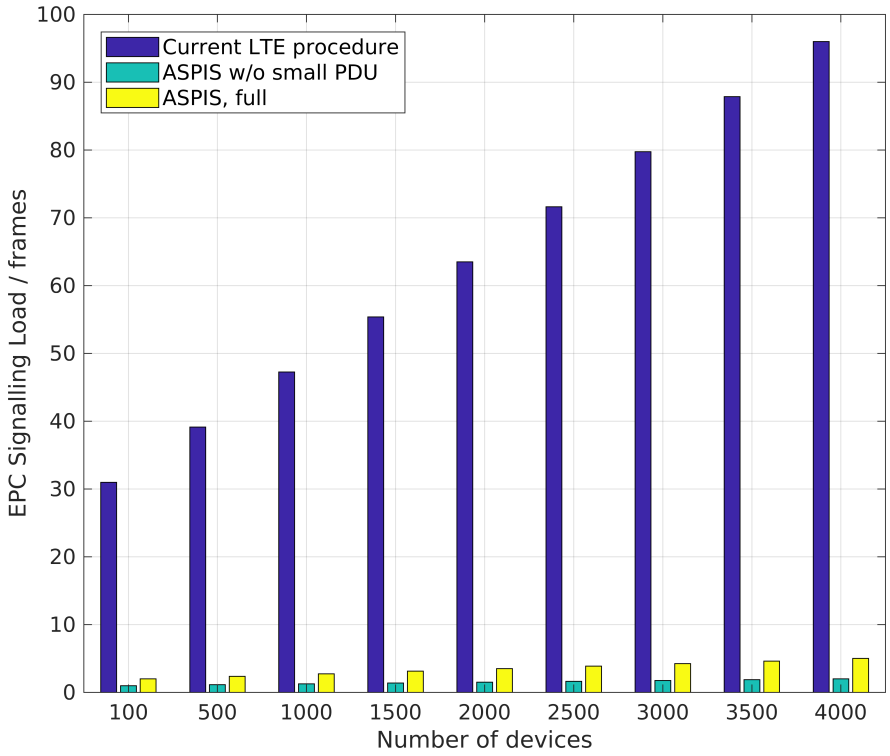


Figure 4.6: **Signalling load of ASPIS in the EPC with and without the small PDU provision, compared to current LTE:** The figure depicts the normalised signalling load of ASPIS in the EPC against the current RA procedure.

the standard LTE.

4.3.2.3 Collisions

Finally, we evaluate the collisions in the RACH of the proactive preamble split scheme of ASPIS compared to other splitting schemes (figure 4.8), which is an important aspect of our method. We measure the collisions during the preamble transmissions and compare it against the standard LTE procedure with the static preamble split [64], which we use as our baseline. Furthermore, we consider every possible static preamble split to show that our proactive preamble split results in fewer collisions in all cases. For visual clarity, we only show the static split with the best performance. Please also note that all these splitting schemes use the intermediate state. Our proactive split performs comparably to the best possible static split, even though the latter is impossible to know in advance, as future HTC traffic cannot be known a-priori.

Finally, we evaluate the collisions in the RACH for a fixed number of IoT devices (4000), and a number of HTC devices varying from 1000 to 10000 (figure 4.9). As before, we only show the standard LTE approach, and the best performing static

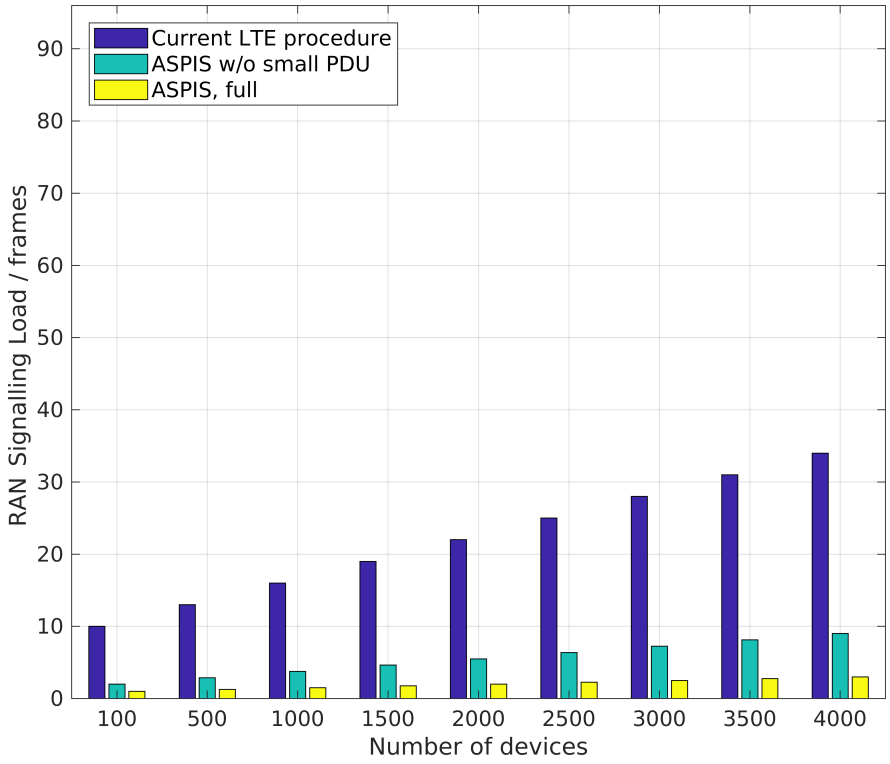


Figure 4.7: **Signalling load of ASPIS in the RAN with and without the small PDU provision, compared to current LTE:** The figure depicts the normalised signalling load of ASPIS in the RAN against the current RA procedure.

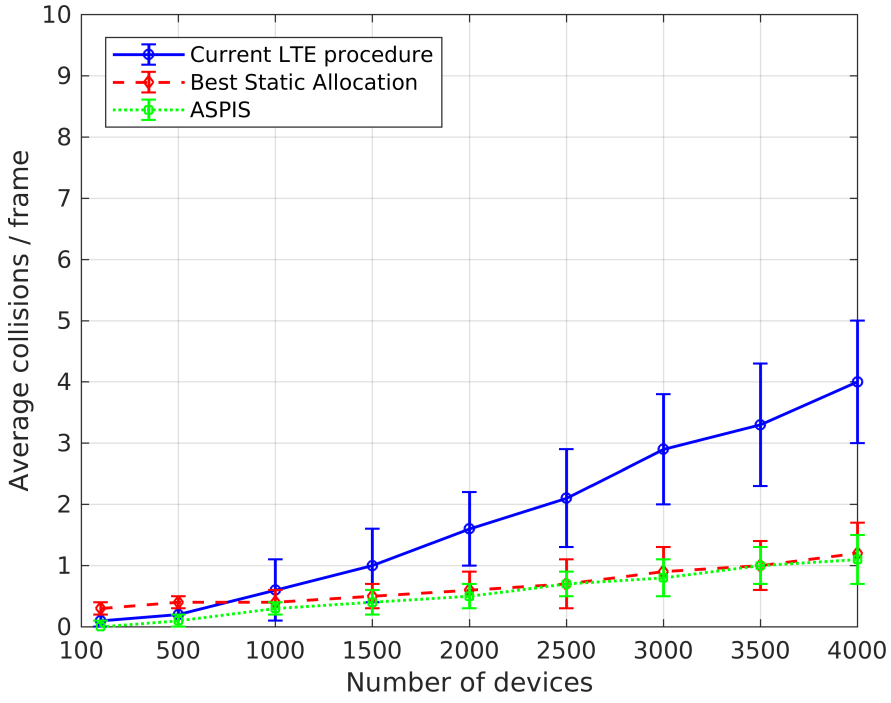


Figure 4.8: **Number of collisions of ASPIS, compared to the current LTE procedure, and the best static preamble split:** The figure depicts the average number of collisions in the RACH, normalised by the number of frames, with the current static scheme of cellular networks, the best performing static split, and our proactive preamble split scheme.

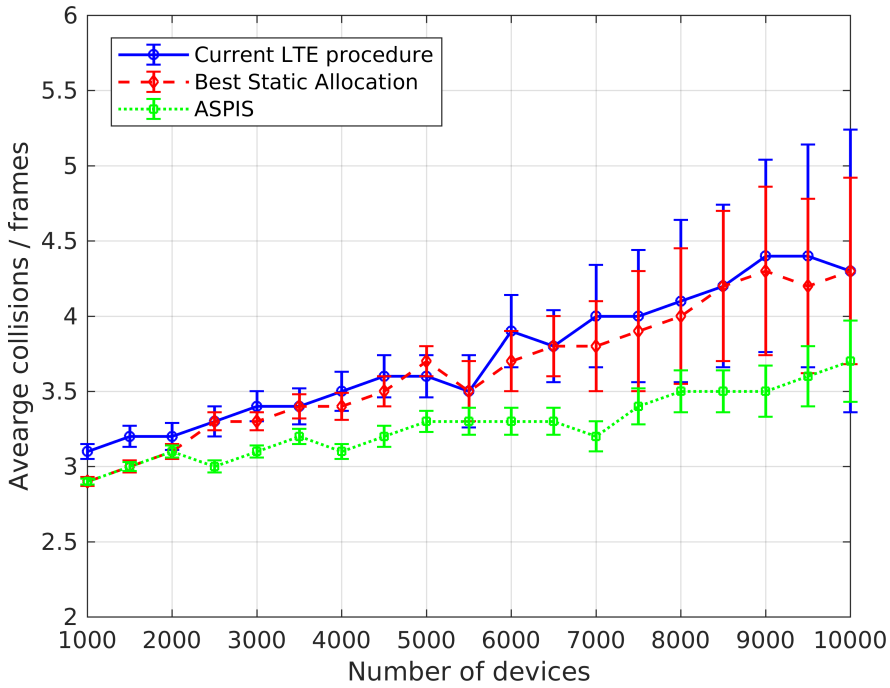


Figure 4.9: **Number of collisions of ASPIS, compared to the current LTE procedure and the best static preamble split, with varying numbers of HTC devices:** The figure depicts the collisions in the RACH, normalised by the number of frames, for varying number of HTC devices, and a fixed number of IoT devices, compared against the current cellular networks procedure, and the best performing static split.

split for clarity. Notice how the differences between the different methods are smaller compared to figure 4.8. As mentioned before, IoT devices have a significantly more negative effect on the number of collisions compared to the HTC devices, due to their frequent and synchronised connections. As a result, increasing the number of HTC devices does not increase the number of collisions as much. However, the results show that ASPIS results in lower collisions in this scenario as well.

4.4 Conclusions

In this chapter we focused on the connection establishment process for large numbers of IoT devices that require periodic and repeated transmissions over LTE networks. Due to their traffic patterns, such devices introduce a considerable signaling load both in the EPC and the RAN, which is usually disproportional to the size of their actual message. Furthermore, such devices can be highly synchronised, resulting in increased collisions in the RACH. To address all these problems, we presented ASPIS, an effi-

cient mechanism that is easy to implement, and works with existing hardware. ASPIS introduces an additional RRC intermediate state that partially preserves a device's connection to be reused in future transmissions, alleviating the need to set up a new connection each time. It is noteworthy that our intermediate state bears similarity to the new inactive state that has been introduced in 5G networks. ASPIS also features an improved RA process with fewer messages that also provisions for short packet transmissions. While we have not conducted energy consumption measurements, as these vary significantly between different devices, we note that reducing the number of messages being exchanged would directly result in reduced energy consumption. Finally, ASPIS incorporates a proactive preamble split scheme that predicts future increases in the access requests, and dynamically adapts the preamble split between IoT and HTC devices, to alleviate collisions before they happen. We showed the practicality of ASPIS through a prototype implementation over the OpenAirInterface platform, and assess its performance via large-scale simulations. Our results show that, at the time of publication, ASPIS outperformed the standard LTE procedures, as well as other recent proposals.

Chapter 5

Probabilistic Preamble Selection with Reinforcement Learning

Possibly the major challenge of the connection establishment process is the increased number of collisions, when multiple devices attempt to establish a connection with the network simultaneously. In this chapter, we focus specifically on the RACH collisions, and propose a novel algorithm that aims to limit them, and thus increase the number of successful connection attempts.

Our proposal is based on the *Non-Orthogonal Multiple Access (NOMA)* [202, 203] transmission schemes that attempt to serve more than one transmissions in the same resources. NOMA schemes have shown to increase the system throughput, and decrease the system latency by using signal separation schemes (e.g.*Successive Interference Cancellation (SIC)* [204]) to distinguish among multiple signals transmitted simultaneously at the same resource. Signal separation can be done on different domains (time, power, etc.), or even on a combination of more than one domain. Such schemes have also been applied in cellular networks as *Non-Orthogonal Random Access (NORA)* [120] to allow for multiple preambles to be transmitted on the same RACH resource without colliding.

In this work we extend the NORA approach, and propose a novel scheme that uses reinforcement learning to increase the number of successful connection attempts and decrease the network access delay during the RA process, for massive numbers of IoT devices. As a first step, we use the existing NORA schemes with signal separation in the time domain, to determine a minimum time difference (called herein as *separation distance*), that two preambles can be distinguished even if they are transmitted on the same resource. We then use the minimum separation distance to split the coverage

area of a cell in different, logical zones, and split the preambles among those zones, so that preambles are re-used in zones which are further apart in the time domain than the separation distance. As the minimum separation distance guarantees the colliding preambles will still be separated, devices can use the same preambles without the fear of collisions.

As a second step, we alter the current RA approach where devices select their preambles with equal probability, and present a mechanism according to which devices choose their preambles probabilistically, based on the preamble usages of the immediate past. The period of time that the preamble usages are observed is dynamically updated with the use of reinforcement learning, to yield the best results in terms of successful connection attempts. Our results show that our proposal allows for a larger number of successful connections with fewer connection attempts, thus significantly decreasing the network access delay.

This work has been accepted for publication at PIMRC 2019 [205].

5.1 Proposed Probabilistic Preamble Selection Mechanism

In this section we present our novel proposal to increase the number of successful connection attempts, and decrease the network access delay. We begin by explaining how the cell area can be split into zones to allow for preamble re-use, and we continue by describing our probabilistic preamble selection algorithm. Finally, we present our reinforcement learning enhancement.

5.1.1 Zone Splitting

The first step of our mechanism is to split the cell coverage area in several logical zones using the NORA methods, so that preambles can be re-used among zones, effectively increasing the total number of preambles available in the cell (figure 5.1).

In order to split the cell area, the BS needs to determine a separation distance Δ_{sep} that allows for successful separation of the preambles. The Δ_{sep} can be determined in different ways based on the domain or combination of domains that the preamble separation is performed (e.g. time domain (Δ_t), power domain (Δ_p), or in any other domain (Δ_x), and the algorithms used by the BS to distinguish among different UEs

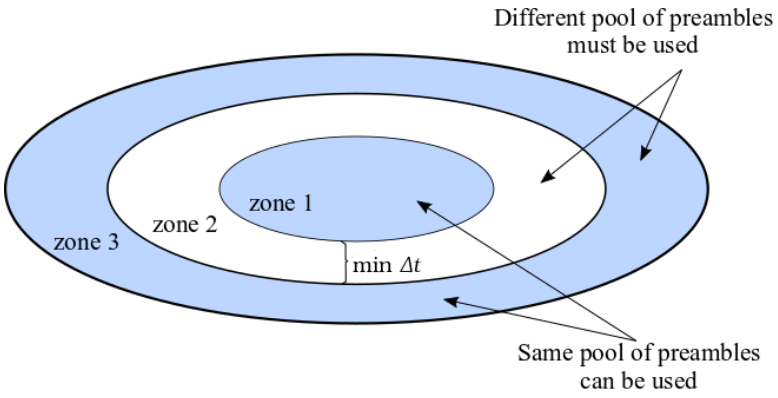


Figure 5.1: **Cell area split into 3 zones:** The figure depicts a cell area split into 3 zones based on the minimum separation distance of the preamble separation domain. Zones 1 and 3 are significantly apart, allowing for the same preambles to be re-used.

that transmitted the same preamble on the same resources. In this work, we perform preamble separation in the time domain. Our approach is as follows.

When the BS receives a preamble at the fist step of the RA process, it attempts to distinguish between possible colliding devices. If multiple preambles are detected, the BS notes the number of detected preambles n_p and the current separation distance Δ'_{sep}, and replies with a RAR for each identified preamble. Please note that, although the BS may have identified multiple devices for the same preamble, it only replies with a single RAR. On the third step, the BS receives the RRC Connection Request messages that correspond to the previously transmitted RAR, and compares their number to the number of identified preambles n_p. If the number of RRC Connection Request messages received is equal to n_p, then the BS updates the Δ_{sep} as $\Delta_{sep} = min(\Delta'_{sep}, \Delta_{sep})$, i.e. the minimum between the previous minimum separation distance, and the one identified this time. The BS also transmits a collision resolution message to resolve the unidentified collisions (if any). Using the separation distance Δ_{sep}, the BS splits the cell coverage area in different logical zones (figure 5.1). The BS then splits the available preambles into different pools and assigns them to each zone, so that zones that are separated by Δ_{sep} can use the same preambles, as even if two devices collide, the collision will be resolved and both devices will be allowed to continue with their connections. The preambles could either be split equally among the zones, or be allocated based on the average UE density of each zone.

Once the cell area is split into zones, the BS broadcasts the separation limits in one of the SIBs. In this work we propose the SIB2, as it carries other information regarding cell access. However, any SIB deemed fit could also carry that information without

changes in the proposed approach. As, in this work, we use the time domain to perform signal separation, we use the *Reference Received Signal Power (RRSP)* to determine the zone lines, however we do note that the limits can be any transmission/reception property that the network operators chooses.

This approach is based on the assumption that enough UEs will collide at the preamble transmission step in order for the BS to accurately determine the Δ_{sep}. Crucially, recent research has shown that IoT traffic is mainly periodic and highly synchronised [16, 15]. Furthermore 3GPP has recognised this behaviour as a potential threat for cellular networks [206]. It is, therefore, expected that the BS will quickly converge to the minimum separation distance, regardless of the domain used for the preamble separation.

5.1.2 Probabilistic Preamble Selection

Although the use of the zones allow UEs to transmit the same preamble without colliding, UEs in the same zone will still collide if they transmit the same preamble at the same RACH resources. Therefore, to decrease the number of collisions within the same zone, we propose a probabilistic preamble selection scheme according to which, the UEs do not choose their preamble at random with equal probability, but instead they choose them probabilistically, based on the preamble usage of the near past.

Our scheme works as follows. We introduce a new *observing window* value that defines the number of frames in the past that the BS takes into account for calculating the preamble usage. For each zone, the BS observes the transmitted preambles for the period of time equal to the *observing window*, in order to create *Preamble Usage Reports (PURs)* for each zone. The preamble usages are presented in a cumulative ascending order, essentially defining the usage percentage of preambles p_1 to p_i inclusive. An example of a PUR is shown in Table 5.1. Please note that, for illustration purposes, we used integer values to represent the preambles. In reality, the root Zadoff-Chu sequence can be used, or any other value representative of the preamble. Also, note that the PURs do not contain probabilities for each preamble, as this would result in all devices selecting the preamble with the lowest probability, effectively increasing the collisions.

When a new UE want to establish a new connection, it draws a random number $r \in [0.0, 1.0]$ and selects the preamble p_i according to $U_{p_{i-1}} < r <= U_{p_i}$, where U_{p_i} is the usage of the i^{th} preamble. The UE then retains the selected preamble for all subsequent

Preambles	1	2	3	4	5	6	7	8	9	10
Usage	0.1	0.16	0.24	0.32	0.41	0.52	0.75	0.82	0.89	1.0

Table 5.1: **Preamble usage report example:** The table shows an example of a PUR, with the usages presented in cumulative ascending order.

connections, until it is instructed to drop it by the BS, using the *RRC Connection Reconfiguration* message. With this scheme, the use of preambles is uniformly spread over the UEs in the coverage area of the BS as well as the time, leading to a reduced number of collisions.

Similarly to the zone limits, the PURs are also broadcasted in one of the SIBs, however there is no requirement that the same SIB is used for both the zone limits and the PURs. Nevertheless, the UEs are required to decode the SIB/SIBs that contain the zone limits and the PURs, before attempting the RA process, in order to determine the zone they belong to and the PUR they must use. It is also important that UEs validate their zone and PUR at each connection, as these may have changed from their last connection.

5.1.3 Reinforcement Learning Enhancement

The size of the observing window plays a crucial role on the efficiency of the system, but deciding on the window value is not straight forward, as the cell changes dynamically in terms of the total number of UEs being served at any given time, the number of UEs in each zone, the frequency at which the UEs perform the RA process, etc. As a static window may not be able to reflect such changes, it is important that the observing window is also dynamically adapted.

Therefore, we further extend our preamble selection scheme with the use of RL to dynamically adapt the observing window, in order to use the most representative one at any given point in time. Our goal is to increase the preamble throughput at each state s, which we define as:

$$R_s = \frac{N_{succ}}{N_{attempts}} \tag{5.1}$$

where N_{succ} is the number of successful connections at a given RACH instance, and $N_{attempts}$ is the total number of connection attempts at that instance.

We use different observing window values to represent ours states, and the possible actions are either to increase or decrease the observing window in multiples of n frames

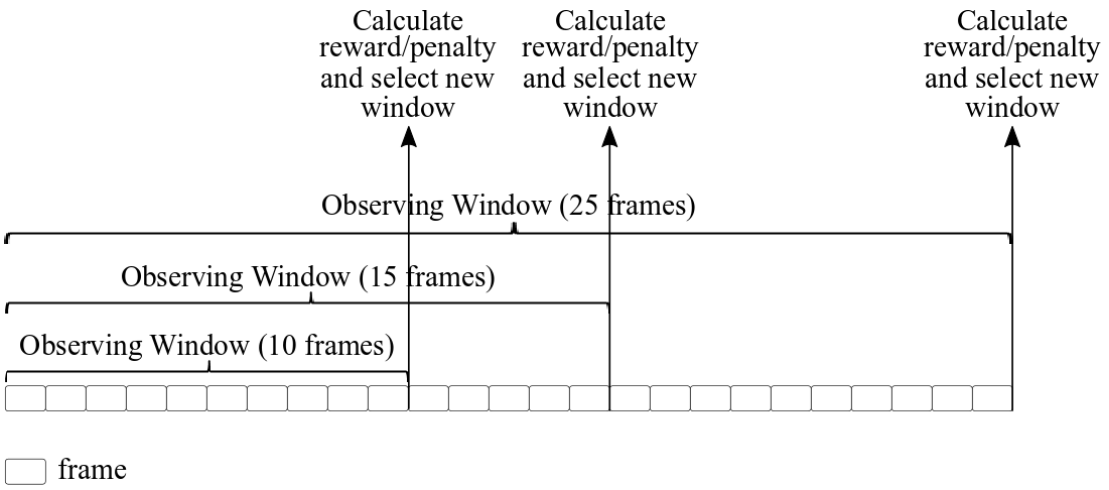


Figure 5.2: **Example of observing window update to produce PURs:** The figure depicts an example procedure to update the observing window to produce the PURs.

(fig. 5.2).

We begin by arbitrarily initializing our state to 100 radio frames, during which the BS observes and preamble usage in each zone. At the end of the observing window cycle, the BS calculates the preamble throughput. Specifically for the first cycle, the BS arbitrarily chooses an action (i.e. increase.decrease), and starts a new observing window value. At the end of each of the following observing window cycles, the BS calculates the preamble throughput and calculates the new action based on:

$$Q^{new}(s_t,\alpha_t) \leftarrow (1-\alpha)*Q(s_t,\alpha_t)+\alpha*\left(r_t+\gamma*\max_{\alpha}Q(s_{t+1},\alpha)\right) \tag{5.2}$$

where $Q^{new}(s_t,a_t)$ is the new state (i.e. observing window value), $Q(s_t,a_t)$ is the previous state, r_t is the reward/penalty for moving from state s_t to state s_{t+1} (i.e. increase/decrease the observing window), α is the learning rate that determines the extend to which new information overwrites the old one with $(0<\alpha\leq 1)$, and γ is the discount factor that determines the weight of the reward r_t, with $(0\leq\gamma\leq 1)$.

As the goal of the algorithm is to maximize its reward, we use the difference of the preamble throughput of the current and previous state based on

$$r_t = R_{s_t} - R_{s_{t-1}} \tag{5.3}$$

At the end of each observing window value, the BS also updates the PURs based on the most recent preamble usage, and broadcasts them in the selected SIB (sec. 5.1.2). To decrease the time until the next PUR update, the observing window can be applied as a sliding window, when the observing window value is increased. Essentially,

the sliding window will result in a delay of $D_{PUR} = new\_window - previous\_window$ frames, until the new PUR is broadcasted.

5.2 Simulation Setup & Evaluation

5.2.1 Simulation Setup

To assess the performance of our proposal, we performed thorough evaluations using a custom simulator written in Matlab that implements the RRC, PDCP and RLC layers of the cellular network protocol stack. In our simulations we assume a typical urban-size cell of 500m radius serving a variable number of UEs. Each UE was randomly placed in the cell area, and for each UE we calculated the probability of being in *Line-of-Sight (LOS)* or *Non-Line-of-Sight (NLOS)* according to [207]. Detailed simulation parameters can be found in Table 5.2.

5.2.2 Results

To evaluate our proposed algorithm we examine different performance metrics that relate to the preamble throughput and the network access delay. For each performance metric we present three different scenarios: (i) current approach with preamble separation, (ii) proposed approach with cell zones, preamble separation and a static observing window for the probabilistic preamble selection, and (iii) proposed approach with cell zones, preamble separation and a dynamic observing window based on Reinforcement Learning. For the second scenario, the observing window value was selected based on a heuristic algorithm, that assessed the performance of different observing windows and selected the one with the highest preamble throughput (equation 5.1).

Preamble throughput: First, we assess the preamble throughput as defined in equation 5.1, for different number of devices and the two start-time distributions defined by 3GPP [64]. Our results (figure 5.3) show that our proposed approach yields significantly better preamble throughput compared to the other two approaches, and allows for a larger number of successful connections in total.

CDF of maximum preamble transmissions for a single data transmission: Here, we present a CDF for the maximum number of preamble transmissions required before establishing a connection for each of the three approaches, for the worst case scenario of 50000 devices in the cell. Our results (figure 5.4) show that when our approach was

Parameter	**Value**
Total number of UEs	1000, 3000, 5000, 10000, 30000, 40000, 50000
Arrival distribution [64]	Uniform, Beta
RA slot period	5 ms
Cell radius	500 m
Number of available preambles in each RACH instance	54
Maximum number of preamble transmissions per UE	10
Preamble transmission time	1 ms
Preamble detection time at BS	2 ms
RAR transmission time	1 ms
RAR processing time at UE side	3 ms
RRC Connection Request transmission time	3 ms
RRC Connection Setup & Collision Resolution transmission time	3 ms
RAR response window	6 ms
Back-off maximum length	20 ms
n	5
Q-Learning α	0.5
Q-Learning γ	0.5

Table 5.2: **Simulation parameters for probabilistic preamble selection with reinforcement learning algorithm.**

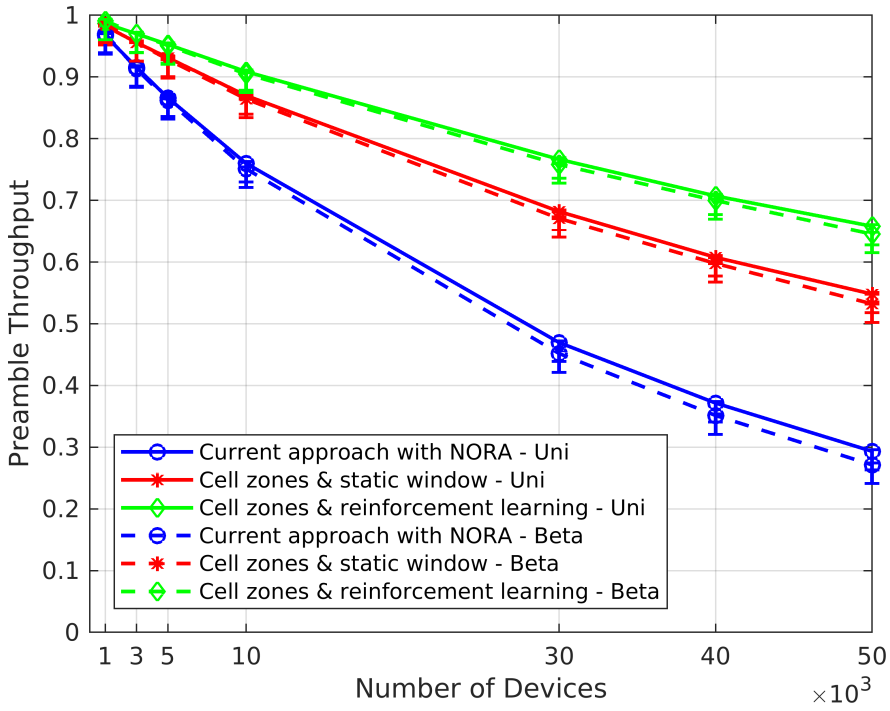


Figure 5.3: **Preamble throughput:** The figure depicts the preamble throughput for the three approaches, with both Uniform and Beta distribution [64].

used devices were able to establish a connection with fewer attempts compared to the other two approaches, with both distributions.

Average number of preamble transmissions for successful UEs: The average number of preamble transmissions required to establish a connection for the case of 50000 devices is 1.5 when our proposal is used, compared to 3.6 that are required when the current connection establishment procedure is used. Our proposal also outperforms the implementation with a static observing window that requires an average of 2 preamble transmissions to establish a connection. These results highlight the gains that our approach offers. For example, if we consider the case of a football stadium with 50000 spectators, the current scheme that only implements NORA, would allow for $\approx$ 14000 spectators to establish a connection after the first attempt. The remaining $\approx$ 36000 spectators would be required to try again, with $\approx$ 14500 spectators being able to connect on their second attempt, leaving $\approx$ 21500 spectators in need for a third attempt, further increasing the delay. On the other hand, when our proposed approach is implemented, $\approx$ 32500 spectators are able to connect on their first attempt (i.e. $\approx$ 18500 more spectators than with the current approach). Of the $\approx$ 17500 remaining spectators, $\approx$ 14900 will be able to connect on their second attempt, leaving only $\approx$ 2600 spectators to try for the third time.

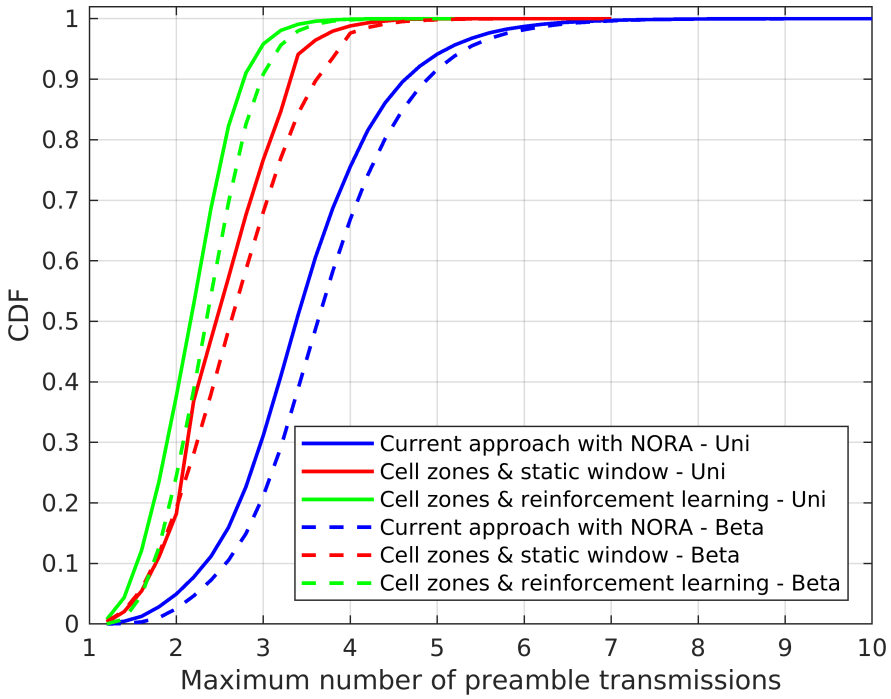


Figure 5.4: **CDF of maximum preamble transmissions for a single data transmission:** The figure depicts the CFD for the maximum number of preamble transmissions for each approach, for both Uniform and Beta distributions [64].

Average network access delay: We also assess the average experienced delay of the three approaches before the connection id established, based on the delay parameters in Table 5.2. We can see (figure 5.5) that our approach results in significantly shorter network access delays, which can greatly improve the QoS and the user experience of the applications.

CDF of network access delay: Finally, we show the CDF of the maximum access delay experienced in the worst case scenario of 50000 UEs with the Beta distribution [64]. Once again, we can see (figure 5.6) that our proposal significantly outperforms the other approaches and results in lower maximum access delay compared to the other considered approaches.

5.3 Conclusions

In this chapter we focused on the increased number of RA collisions when numerous devices attempt to establish a connection simultaneously. We present a novel algorithm that is based on the state-of-the-art proposals on NORA, and is enhanced with reinforcement learning to reduce the number of collisions during the initial network

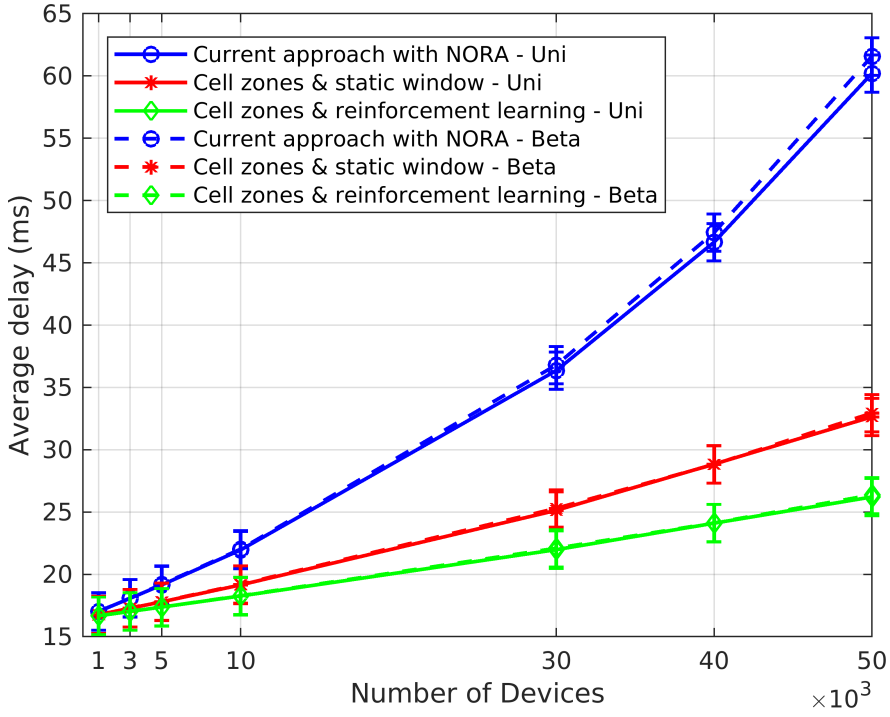


Figure 5.5: **Average network access delay:** The figure depicts the average network access delay experienced for the three different approaches, for both the Uniform and Beta distributions [64].

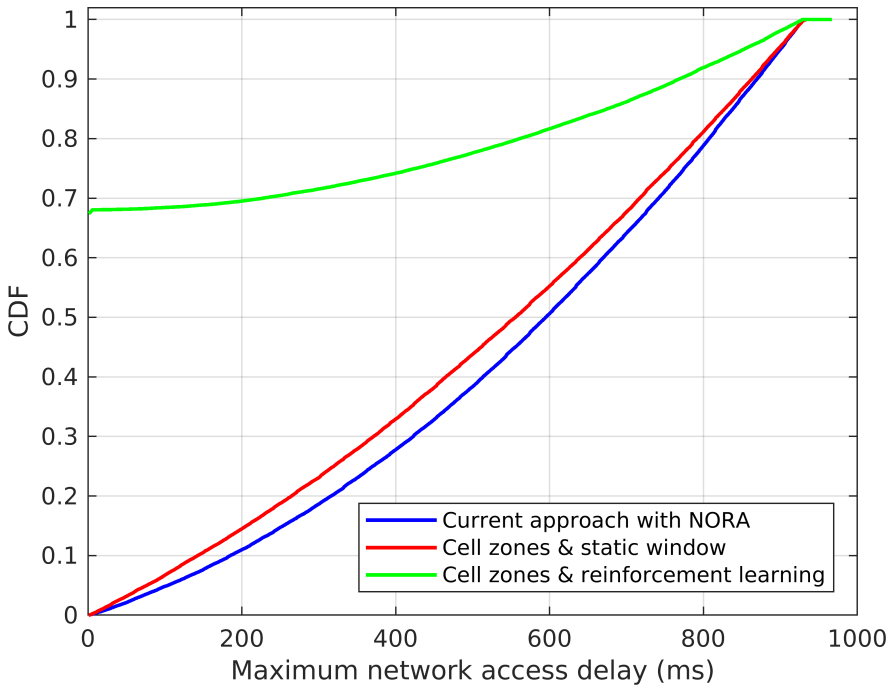


Figure 5.6: **CDF of network access delay:** The figure depicts the CDF of the network access delay for 50000 devices with Beta distribution [64].

access, and decrease the network access delay. Our results show that our proposal outperforms the currently used scheme and allows a larger number of successful connections with fewer connection attempts, significantly decreasing the network access delay.

Chapter 6

Network Resource Utilisation for Group Communications in NB-IoT

In this chapter we address the network resource utilisation and device energy consumption problems for group communications, and present a framework that is specifically tailored to the limited resources of the NB-IoT technology, and the battery requirements of NB-IoT devices (figure 6.1). Towards this end, we diverge from the subscription-based model of eMBMS (section 2.1.3.3), and present an on-demand multicasting scheme according to which devices are notified whenever there is a service they must receive, foregoing the need for periodic service announcements and channel monitoring. Our approach allows for dynamic creation of device groups (based on the device make, performed tasks, etc.), which offers greater flexibility compared to statically selecting the services that a device should subscribe to, at manufacturing time.

The main idea of our approach is that the devices no longer select services to subscribe to, but are instead individually notified by the network whenever they need to receive multicast content. Specifically, the service provider (e.g., manufacturer, owner) provides a list of devices that should receive a certain service, as well as the service content itself (e.g. firmware update) to the network operator. The network operator can then notify the listed devices directly to receive the multicast data (figure 6.1). This approach lowers the responsibility of the device while being more robust, as new services might appear over time, that the device would not be aware of. We also investigate two different multicast transmission strategies and their effect on background traffic. Furthermore we explore three novel grouping techniques in order to synchronise the devices for the reception of the multicast context.

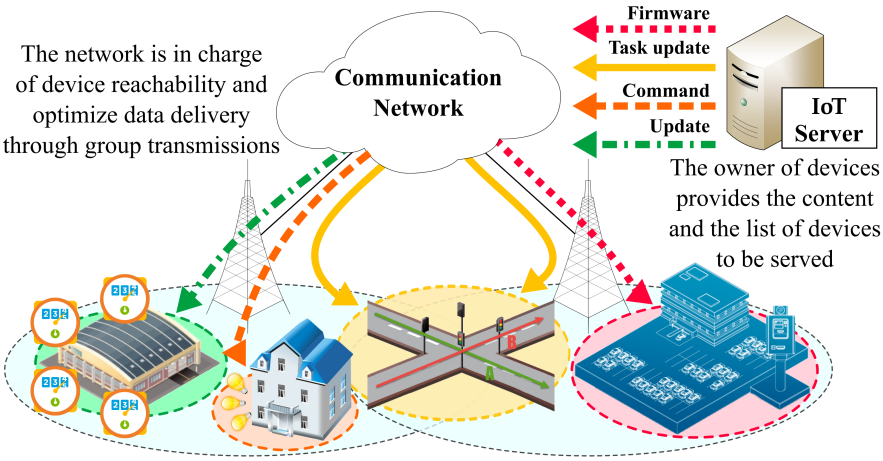


Figure 6.1: **Example of group communications in NB-IoT:** IoT devices are dynamically placed in different groups, each one receiving a specific content. The device manufacturer/owner provides the list of devices to receive the multicast transmissions, as well as the content itself. The network then pages the devices accordingly, and transmits the required service.

Our approach overcomes the major drawback of the SC-PtM framework that requires resources from the data transmission channel for the periodic service announcements, even when there are no ongoing services to announce at the time, which can result in severe system performance degradation in the resource-limited NB-IoT technology. Furthermore, the devices are not required to continuously monitor the service announcements, as services such as software/firmware updates, which would be the major use case for NB-IoT, are rare and far in-between, thus keeping the energy consumption low.

The work presented in this chapter has been published in two different publications. The work on group communications and transmissions strategies has been published in the IEEE Internet of Things Journal, in June 2018 [24], while the work on grouping and synchronisation of devices has been published in ICDCS, in July 2018 [25].

6.1 Deficiencies of SC-PtM

As SC-PtM inherits the subscription-based architecture of eMBMS (section 2.1.4), it is ill-suited for NB-IoT due to its requirement for periodic announcements and monitoring of the SC-MCCH. Specifically, in order to broadcast the control information, a number of NPDSCH resources need to be allocated to the SC-MCCH on regular in-

tervals, regardless of whether ongoing transmissions exist. Specifically for the cases of firmware updates or task commands, the services need to be available at all times, as new devices need to be able to subscribe to them at any given time. As there is no indication as to when the next session will start, SC-PtM requires constant allocation of NPDSCH resources to the SC-MCCH. In the resource limited NB-IoT technology, this may result in severe degradation of the systems performance as precious resources are wasted for control information of pending multicast services. Furthermore, devices that have subscribed to SC-PtM services need to periodically monitor the SC-MCCH for information regarding their services. The subscription and monitoring is done on a per-service basis, and on top of their own DRX/eDRX cycles, thus increasing their energy consumption.

For NB-IoT multicast services we can assume that devices subscribe to at least one service after their first power on (e.g., to receive updates from the manufacturer). Hence, the number of concurrently available services should be at least equal to the number of types/makes/models of the devices present in the cell, as even different models of the same manufacturer may require different updates. In SC-PtM, the maximum number of concurrent services is 64 [34]. However, due to the variability of the devices, the applications they run, and the fact that the services need to be available all the time, the number of different services will be much larger. Furthermore, from the device perspective, deciding which services it should subscribe to can be very challenging, as it is the device manufacturer, the application provider or the device owner that decides which updates or commands should be delivered to which devices, and the different multicast groups may need to be created on the fly. Therefore, the SC-PtM scheme is not well suited for the types of devices and applications targeted by NB-IoT.

6.2 Group Communication Framework for NB-IoT

As the eMBMS framework (section 2.1.3.3) is based on a number of complex procedures that introduce high latency and increase the energy consumption of IoT devices, we present several enhancements to align with the NB-IoT's goals of low complexity and long battery life. We begin by presenting modifications to the eMBMS procedures, specifically designed for IoT use cases, with particular focus on multicast scenarios. We then present new mechanisms for paging and delivering multicast content to a group of devices within an NB-IoT frame, that try to reduce the impact of multicast traffic on existing unicast traffic.

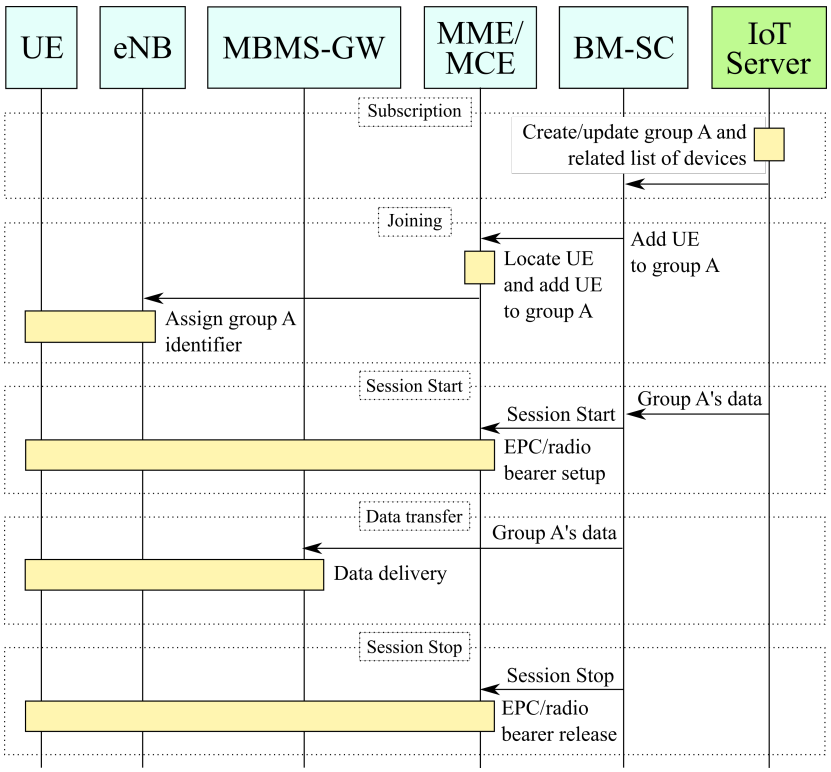


Figure 6.2: **eMBMS procedure enhancements:** The figure depicts the proposed enhancements to the eMBMS framework. A coordination entity, such as the device owner or manufacturer, decides which devices should receive a certain multicast content using an IoT server. Then, the cellular network locates the devices and pages them when the multicast transmission is imminent.

6.2.1 Proposed enhancements to eMBMS

To align with the NB-IoT's goal for low complexity, we alter the eMBMS framework to include a limited set of procedures, thus simplifying the process for multicast content reception. For our enhancements to the eMBMS standard (figure 6.2) we assume the presence of a coordination entity (e.g., owner of IoT devices, device manufacturer, application provider) that is responsible for deciding which devices must receive a specific content.

Specifically, during the *subscription* phase the co-ordination entity provides the list of devices to receive the multicast content, i.e. which devices must be placed in the same multicast group through the IoT server.

The *service announcement* has been removed, since the concept of service availability is no longer applicable in NB-IoT, where devices receive content which has been decided and provided by the coordination entity during the subscription phase.

The *joining* procedure is different from that of the current eMBMS. Instead of waiting for a device to join a group, the network now informs the devices about the multicast group(s) they belong to, by paging the devices and allocating them a multicast group id.

The *eMBMS notification* procedure has also been removed to reduce the signalling overhead towards the devices, and the session start procedure is used to inform the devices about an imminent service. The *session start* procedure is triggered when the IoT server provides the multicast content for a given multicast group. Before the session start, the network establishes the required core network bearers. It also establishes new bearers in the RAN or activates any bearers that might still exist from a previous session (section 6.2.2). Once all bearers are set up and active, the data transmission can start. At the end of the data transmission, the network triggers the *session stop* to release the bearers previously created in the core network, and deactivate the bearers in the RAN.

From an architectural point of view, the core network needs to be able to receive the list of devices from the coordination entity for a certain group, and store the information related to the membership of devices of a given group. Generally, the core network would implement policies to manage group sessions for both the control and data traffic, while considering the energy consumption of the devices and the control traffic overhead.

6.2.2 Bearer Setup and Paging

Similarly to any communication in cellular networks, bearers are also required for the transmission of the multicast data. However, bearer establishment/activation for group communications needs to be managed efficiently, to reduce the energy consumption of the devices, and the control signalling overhead, both at the RAN and the core network. Therefore, we propose a combined service activation and joining procedure according to which, the network sets up a generic eMBMS bearer for the devices according to their supported QoS, or instructs the devices to join an already existing one that meets their QoS requirements (if one exists). Devices perform this step after their first session start process, where the BS can adapt the transmission to their supported QoS [31]. At the session stop process, the BS releases the multicast bearers in the core network, but keeps the RAN multicast bearers in an idle status in order to be reused in the future. Therefore, devices perform the multicast bearer setup only once, and skip this step for

each subsequent sessions, thus decreasing both the latency and the energy consumption when receiving future multicast content. As there is only one bearer per group, the number of multicast bearers is not expected to be a significant burden to the BS.

With the multicast bearers set up, the devices need only to be informed about imminent multicast transmissions. In our scheme devices do not monitor service announcements, but are instead specifically paged to receive the multicast data. Driven by [208], we propose a grouping on-the-fly scheme, where the coordination entity provides the network with the list of devices uniquely identified by their IMSIs to receive the multicast content. While the owner of devices creates the groups without considering the DRX/eDRX cycles, it is the network that manages the grouping of devices in an efficient way, taking into account the device energy consumption, as well as the network resource utilisation (section 6.4).

6.2.3 Transmission Strategies

In LTE and 5G networks, multicast transmission is accomplished by allocating all PRBs of selected subframes to eMBMS. However, NB-IoT is deployed on a single PRB, which significantly limits the number of available resources for multicast communications. To address the resource limitations, we present two methods for multicast data transmission in NB-IoT (figure 6.3):

1. multicast with fixed guarantee (MFG)
2. multicast with priority (MP)

6.2.3.1 Multicasting with Fixed Guarantee (MFG)

The aim of this strategy is to minimise the impact of the multicast traffic on unicast. This is achieved by giving priority to the unicast traffic, and transmitting multicast content only when there are available resources. However, as the unicast traffic is generally uniformly distributed in time, giving priority to it at all times might result in having very limited resources for multicast. This will lead to increased delays for the multicast traffic, and increased energy consumption for the devices receiving it, as they would either have to remain active for longer periods of time, or they would have to be paged multiple times.

To overcome this problem, the *Multicasting with Fixed Guarantee (MFG)* transmission strategy provides a minimum number of guaranteed resources per frame N_g,

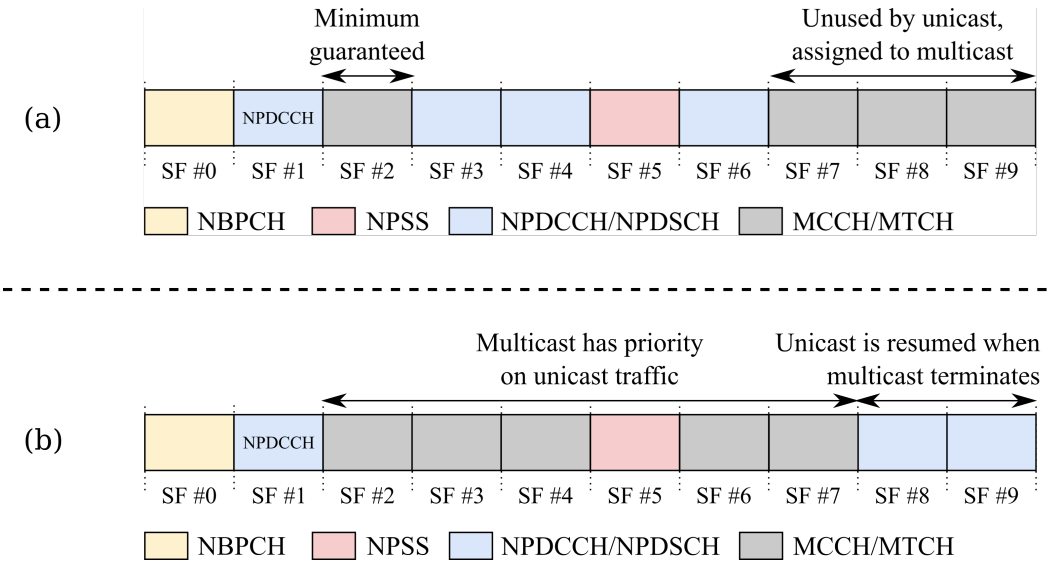


Figure 6.3: **Proposed transmission strategies for multicast context in NB-IoT:** The figure depicts the two proposed transmission strategies applied to odd frames of NB-IoT: (a) MFGs, (b) MP. In the former approach, priority is given to unicast traffic while providing a minimum guaranteed number of subframes for the multicast transmission, which can be dynamically adapted. In the latter approach, priority is given to the multicast traffic by allocating all subframes to the multicast transmission, with the goal of completing the transmission as soon as possible. The unicast transmissions are then resumed when the multicast transmission is terminated.

to be used for multicast transmissions, leaving the remaining resources available for unicast traffic. The N_g applies to all frames until the end of the multicast transmission, including any repetitions. Control and data information is carried using the existing logical control and traffic channels (MCCH and MTCH respectively), which are delivered using the resources allocated for the multicast transmission. The number of reserved resources N_g is network operator specific, and it can be decided based on the average observed traffic, or past information of the same periods of time[1]. The specific resources allocated to multicast traffic, as well as the exact scheduling of the control and data information are signaled in the SIB20-NB, similarly to SC-PtM. An example of resource allocation is depicted in figure 6.3(a), where N_g is equal to 1 subframe. It is worth noticing that resources which are not used for unicast transmissions can also be dynamically scheduled for multicast to reduce the transmission time. In this case, the BS informs the devices using the SIB20-NB, indicating a change in configuration.

6.2.3.2 Multicasting with Priority (MP)

Although MFG minimises the impact of multicast traffic on unicast, it can significantly increase the delay under high load conditions, as multicast traffic will only receive a limited amount of resources. MFG can also increase the energy consumption of the devices receiving the multicast content, either by keeping them on for longer periods of time, or by paging the multiple times.

Therefore, we further present the *Multicasting with Priority (MP)* transmission strategy that aims to deliver the multicast content as quickly as possible, thus lowering the transmission latency and the energy consumption for the devices receiving the multicast content. In MP, priority is given to the multicast transmission by allocating all the NPDSCH subframes for it. The only exception is the transmission of the SIBs-NB, which are also transmitted in the NPDSCH on frequent occasions. Unicast traffic is thus paused, and only resumed at the end of multicast session. Similarly to MFG, the same control and data channels are used, and their scheduling information is signaled in the SIB20-NB. An example of a frame allocation is depicted in figure 6.3(b).

The MP approach is suitable for short and infrequent multicast transmissions, i.e., when the amount of multicast sessions is limited or when the multicast sessions are sparse. Although it impacts the unicast traffic more compared to the MFG strategy,

[1]By exploiting the periodic nature of NB-IoT devices it is possible to predict when resources will be needed for unicast transmission, and therefore choose to transmit multicast data at times when the expected unicast traffic is low.

it only delays data exchange for devices expecting to receive unicast traffic during the multicast transmission. However, it is worth highlighting that the amount of suspended unicast transmissions is limited compared to the number of devices receiving multicast data, i.e., MP introduces delays for a smaller set of devices compared to the MFG approach (section 6.3.2). Moreover, delaying a unicast transmission does not significantly increase the energy consumption for devices receiving unicast data, as the network can buffer the downlink packets and page these devices to resume their reception after the multicast transmission is completed.

6.3 Evaluation

6.3.1 Experimental Setup

To assess the performance of our eMBMS enhancements and transmission strategies, we conducted a thorough experimental evaluation using a custom simulator written Matlab. Our experiments were performed under the presence of realistic NB-IoT background traffic (e.g., NB-SIBs, downlink RA messages, and application ACKs), based on [14, 209].

Specifically, we simulated a 5 MHz cell with an in-band NB-IoT deployment serving 55000 NB-IoT devices. For the multicast content, we transmitted 1 MB of data, which we believe to be representative of a typical firmware update for such devices. Similar to the approach taken by 3GPP [210, 211] at the time of writing, we assumed the use of an application layer forward error correction scheme based on fountain raptor codes, which increased the effective size of the multicast content to 1.2MB.

For the unicast traffic, we generated a valid DRX/eDRX cycle for each NB-IoT device, based on the allowed values [212], as well as random periodicities based on [209, 16]. We assumed that devices transmitted a single application packet of 100 bytes according to their periodicities, and received one downlink ACK of 45 bytes, that follow the same distribution as the uplink transmissions [209]. For both uplink and downlink transmissions we assumed 5B Internet protocol header, 8B packet data convergence protocol header, 8B radio link control header, and 16B MAC header, resulting in 137 of uplink data, and 82 of downlink data. In terms of subframe configuration, subframes #1 and #6 were allocated to NPDCCH, while the remaining subframes were allocated to NPDSCH. Finally, we assumed a maximum *Transport Block Size (TBS)* of 680 bits.

We assessed all scenarios using both normal and extended CP for single- and multi-cell deployments, respectively (section 2.1.3.3). Extended CP needs to be used when multiple BSs form a single frequency network to decrease the signal interference. However, using extended CP in the NB-IoT subframes requires that the same subframes of the covering LTE/5G deployment (in the case of in-band deployment) also apply the extended CP, thus reducing the overall system performance. Such degradation is, of course, undesirable but it is worth exploring the system behaviour in such cases. Finally, we assume that devices check the paging channel, based on their DRX cycle, and their active period is 1ms long.

In our experiments we measure (i) the delivery time, (ii) the NPDSCH occupancy, and (iii) the device uptime for monitoring and receiving control information. The *delivery time* is measured differently for the unicast and multicast traffic. In the former case, we measure the delivery time as the time elapsed from the moment the BS receives an acknowledgment from the IoT server (hence, different for each device) to the moment the ACK is received by the device, including all repetitions. In the later case, the delivery time is the time elapsed from the moment the BS receives the content (hence, the same instant for all devices involved in the multicast transmission) to the moment that the multicast content transmission is completed, including all repetitions. For the *NPDSCH occupancy*, we measure the percentage of NPDSC resources, over the overall availability in a frame. Finally, the *device uptime* is the average time that a device remains awake. Since the uptime caused for data transmission is unavoidable, we only measure the uptime due to monitoring and receiving the control information and paging. As actual energy consumption values are device specific, we use the uptime as a proxy for the device's energy consumption, as the longer the device remains awake, the more energy it will consume. For detailed, device-specific energy measurements and analysis please refer to chapter 7.

6.3.2 Results

6.3.2.1 Unicast Baseline

We begin by assessing the generic performance of the system when only background traffic exists to be used as our baseline, in order to understand the impact of multicast traffic on unicast. In this experiment, we examine different number of repetitions (figure 6.4), and our results show that on average the delivery time varies from a few hundreds of milliseconds up to a few seconds. The maximum delivery time is in the

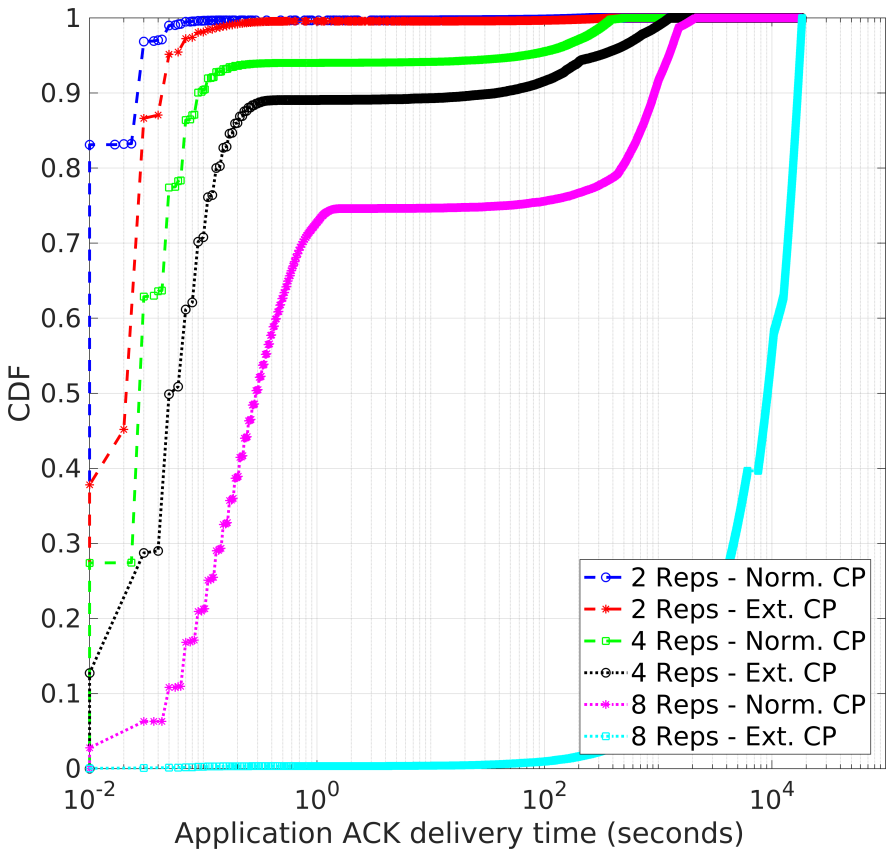


Figure 6.4: **Unicast latency for the baseline scenario:** The figure depicts the application ACK delivery time when no multicast data is being transmitted. In normal operation the delivery of an application ACK varies from a few hundreds of ms for 2 repetitions and normal CP, to a few hundreds of seconds for 8 repetitions and extended CP.

order of hundreds of seconds and it is obtained when 8 repetitions are used, with extended CP. The average NPDSCH occupancy (figure 6.5) is also drastically affected by the presence of repetitions. The occupancy is about 20% when 2 repetitions are used, and it increases to 45% with 4 repetitions. When using 8 repetitions, the occupancy reaches a value of $\approx 90\%$.

6.3.2.2 Multicasting with Fixed Guarantee (MFG)

In this experiment, we assess the impact of the different number of repetitions on the delivery time of the multicast traffic (figure 6.6), when the MFG transmission strategy is used. We can see that the maximum delivery time is significantly large when 8 repetitions are used, only 1 SF is guaranteed and the extended CP is applied. We can also observe how generally, the performance of the multicast transmission is consistently worse when extended CP is used, due to the limited resources availability.

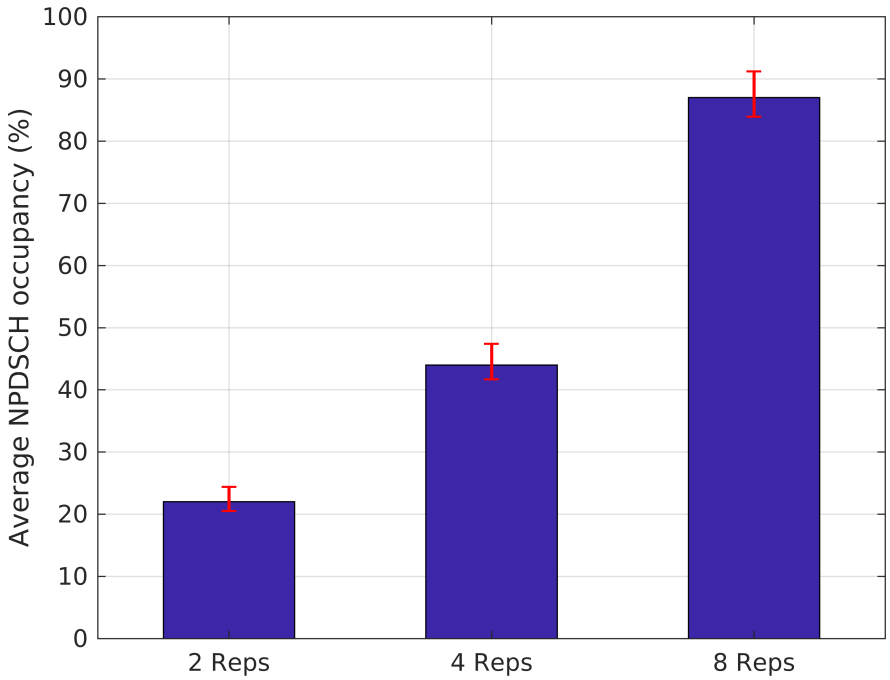


Figure 6.5: **Average NPDSCH occupancy for the baseline scenario:** The figure depicts the average NPDSCH occupancy when no multicast data is being transmitted. The NPDSCH occupancy is $\approx 20\%$ with 2 repetitions and increases close to 90% with 8 repetitions.

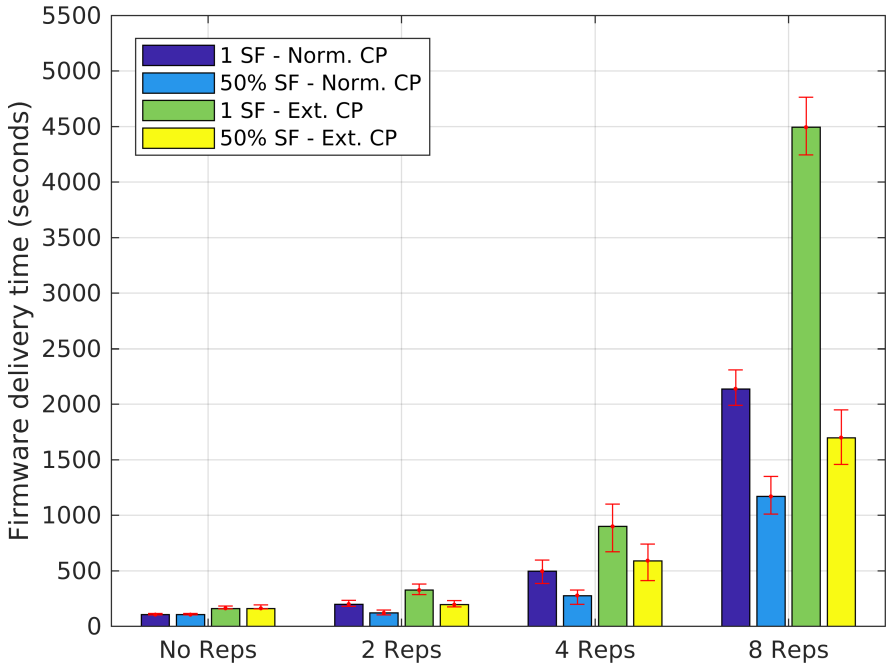


Figure 6.6: **Multicast delivery time using MFG:** The figure depicts the delivery time of multicast data using the MFG transmission strategy. The maximum delivery time is $\approx$ 4500s with 8 repetitions, 1 SF guaranteed, and extended CP. When 2 repetitions are used the delivery time is less than 500s, regardless of the CP used.

In terms of impact on the unicast traffic, results are not depicted since giving priority to the unicast traffic does not drastically affect the performance, even in the case of a multi-cell scenario with 8 repetitions. In terms of the NPDSCH occupancy (figure 6.7), we can see that it is not drastically affected compared to the unicast baseline when 1 SF is guaranteed for multicast and the normal CP is used, which is expected to be the main deployment option for multicasting in NB-IoT.

6.3.2.3 Multicast with Priority (MP)

In this experiment, we evaluate the performance of the MP transmission strategy, and its impact on the background traffic (figure 6.8). Our results show that the firmware delivery time decreases down to $\approx$ 540 seconds and $\approx$ 780 seconds when the normal and extended CP is used respectively, and 8 repetitions are used. These results highlight the benefits of the MP strategy on the firmware delivery time. At the same time, the application ACK delivery time is very similar to that obtained in the unicast baseline (figure 6.4). This denotes that the MP transmission strategy introduces delays only to a very limited number of unicast devices, i.e., those needing to receive

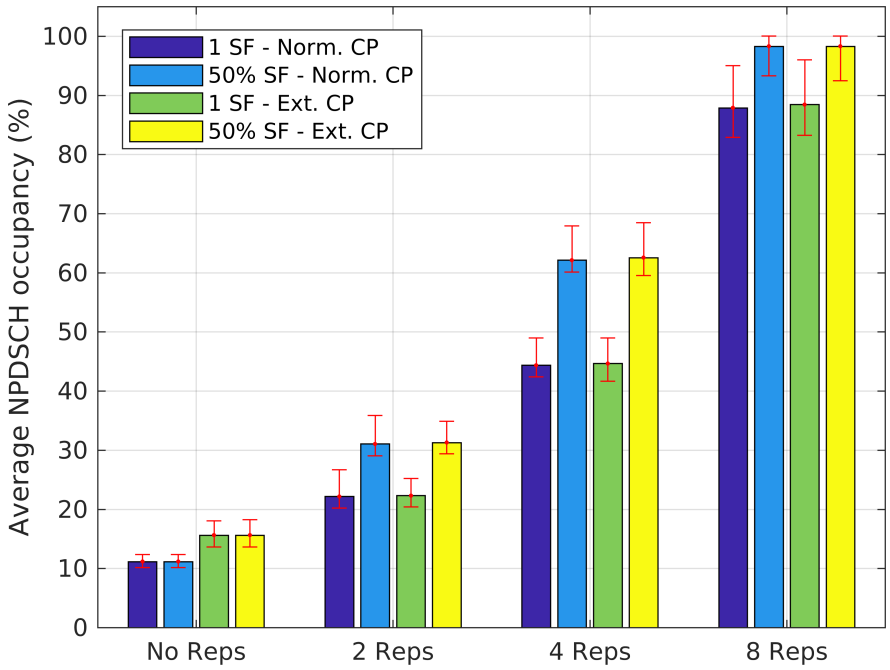


Figure 6.7: **Average NPDSCH occupancy using MFG:** The figure depicts the average occupancy percentage of NPDSCH when the MFG transmission strategy is used. The occupancy is not drastically affected by the N_g when small number of repetitions are used, but present a significant increase as the repetitions also increase.

data concurrently to the multicast transmission, without drastically affecting the overall performance of the background unicast traffic. The NPDSCH occupancy of the MP strategy (figure 6.10) presents interesting results, as there are small differences compared to figure 6.4. However, taking the latency into account these results show that the overall impact of the MP approach is limited.

6.3.2.4 Single Cell - Point to Multipoint (SC-PtM)

In this section we compare our two proposed transmission strategies with the SC-PtM approach (figure 6.11), considering three different scheduling periods, that cover the range specified by 3GPP [34]: 2, 50, and 200 RFs. Furthermore, for fair comparison, we used 50% of resources for the multicast transmission for both SC-PtM and MFG. In terms of the multicast delivery time, we can see that our MFG approach outperforms the SC-PtM, even though it prioritises the background traffic over the multicast. This is expected, as this approach utilises all available resources not used by background traffic, while at the same time it guarantees a minimum number of resources per frame for multicast data. When the MP strategy, which prioritises multicast traffic, is used the

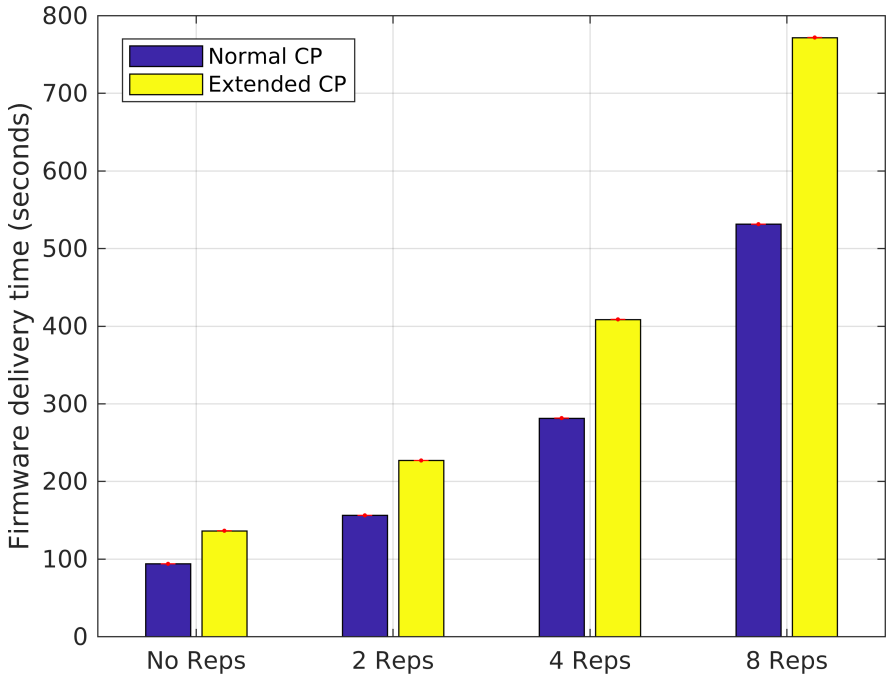


Figure 6.8: **Multicast delivery time using MP:** The figure depicts the delivery time of a firmware update with the MP transmission strategy. The firmware delivery time of the MP approach is lower compared to MFG even when 8 repetitions are used.

margin over SC-PtM only grows, while at the same time incurring a minimal impact on the background traffic.

The device uptime in SC-PtM is heavily dependent on the scheduling period of SC-MCCH. For simplicity we assume that devices are subscribed to only one service. Note that this is the best-case scenario as, subscribing to more than one services would only increase the uptime of the device even further. Similarly to our method, we use a value of 1ms for the monitoring instance of both the SC-MCCH and the paging channel. Our results (figure 6.12) show that the uptime over a period of a day is significantly larger for SC-PtM, as devices using this scheme need to monitor both the paging channel and the SC-MCCH periodically. The uptime, of course, is directly linked to the energy consumption of the devices, which can add up significantly over a span of ten years.

Next, we compare the NPDSCH occupancy dedicated to multicast data and control information (figure 6.13). For fairness, we use the same amount of resources between SC-PtM and our scheme, and the same number of repetitions. Note however, that our method guarantees a minimum number of resources while in SC-PtM the multicast content transmission is depended on the resource availability. Our approach essentially only uses the resources needed for data transmission and control information.

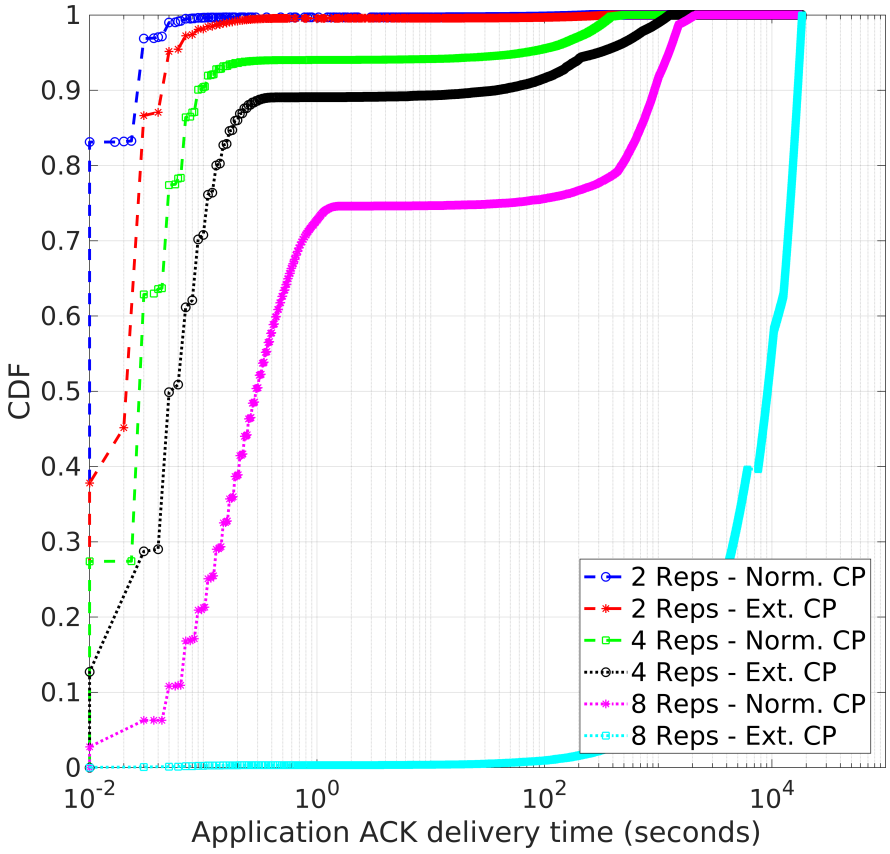


Figure 6.9: **Application delivery time using MP:** The figure depicts the delivery time of the application ACK when the MP transmission strategy is used. The ACK delivery time is similar to the baseline indicating that the impact on the background traffic is minimal.

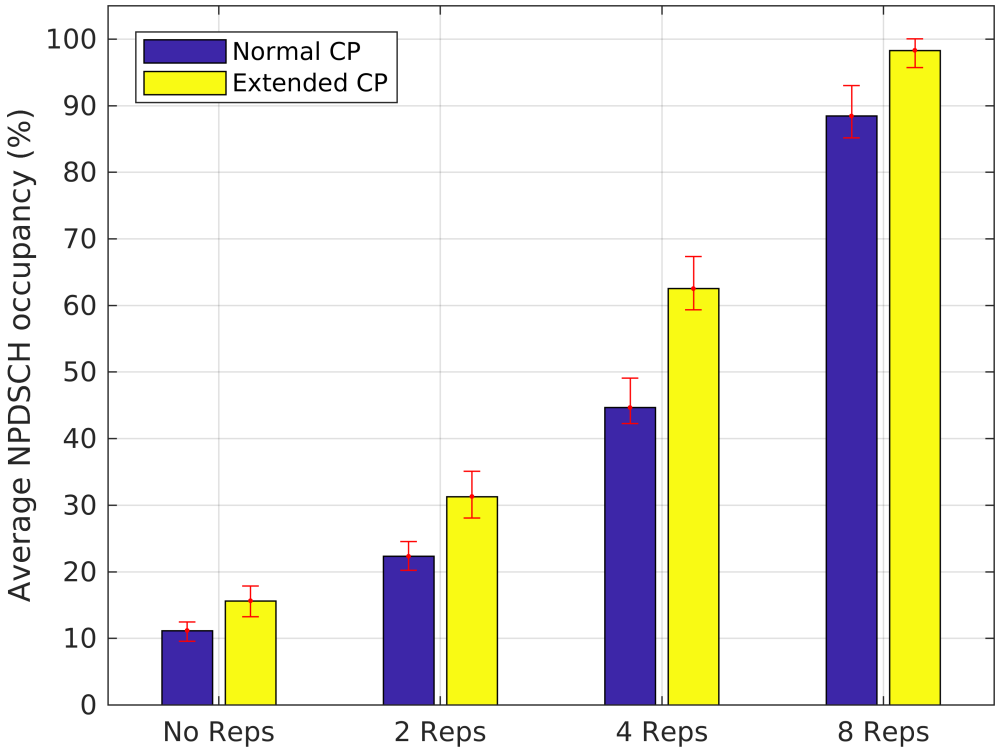


Figure 6.10: **Average NPDSCH occupancy using MP:** The figure depicts the occupancy percentage of the NPDSCH when the MP transmission strategy is used, which remains similar to baseline (figure 6.4)

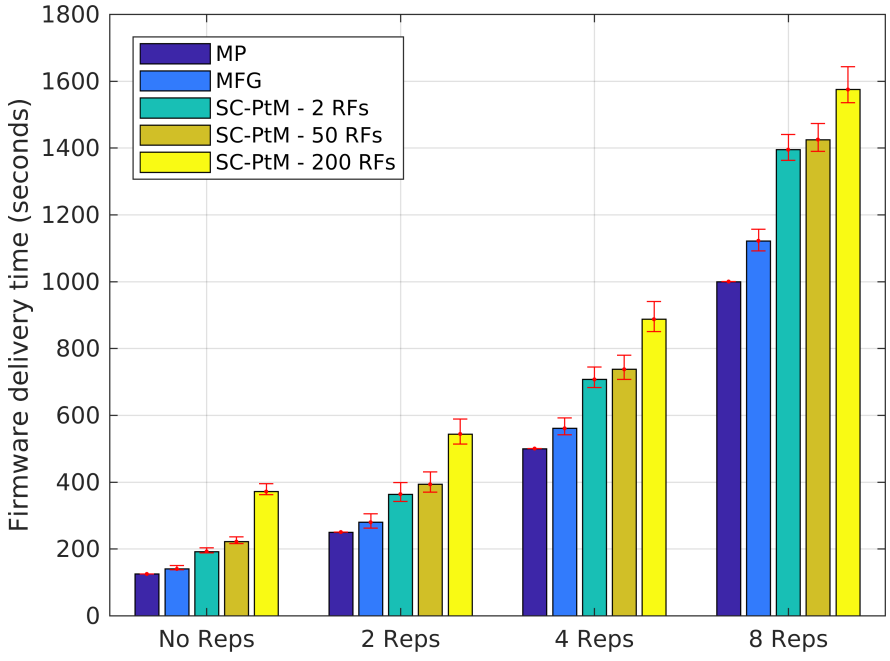


Figure 6.11: **Firmware delivery time of all multicast approaches:** The figure depicts the firmware delivery time of MFG, MP and SC-PtM approaches. We can see that both our proposed transmission strategies outperform SC-PtM.

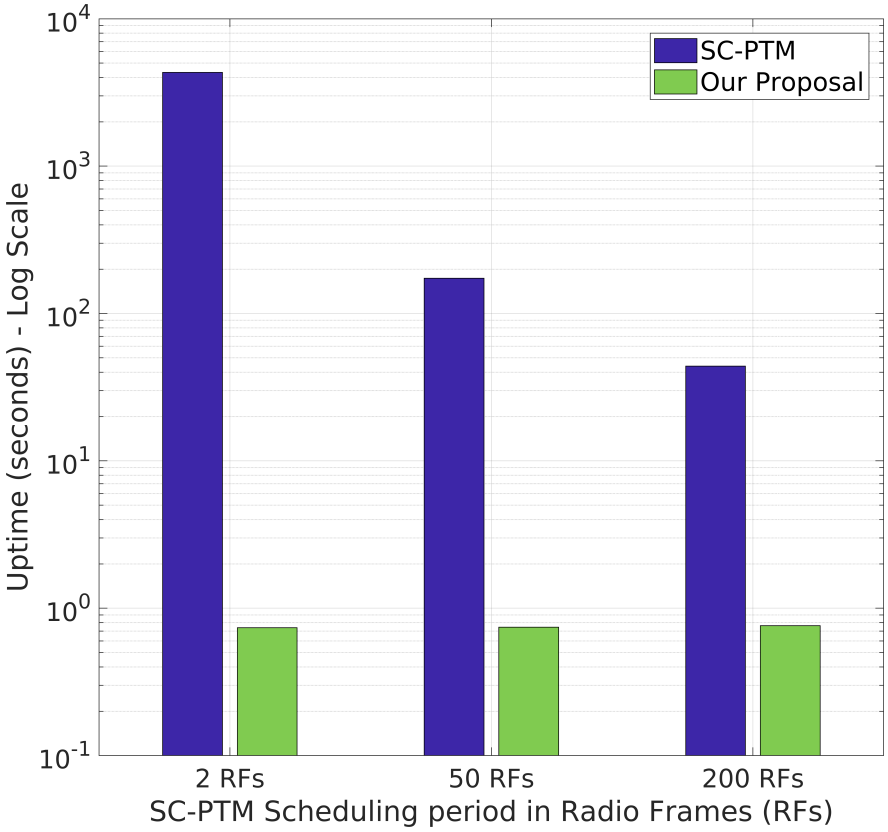


Figure 6.12: **Uptime of SC-PtM:** The figure depicts the device uptime when the SC-PtM approach is used. The uptime over a period of a day is significantly larger for SC-PtM compared to our scheme, as devices under SC-PtM need to monitor both the paging channel and the SC-MCCH.

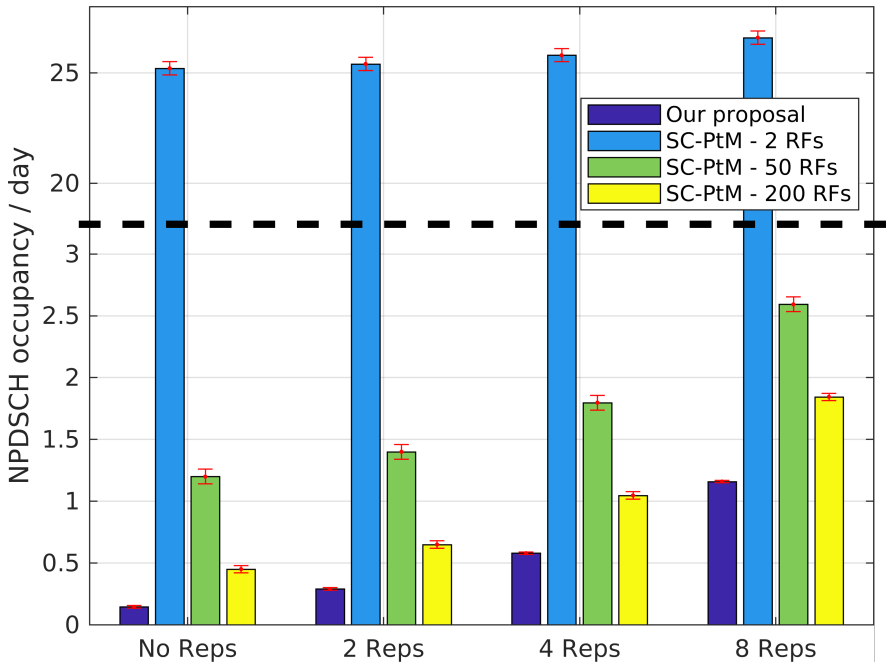


Figure 6.13: **NPDSCH occupancy of all multicast approaches:** The figure depicts the NPDSCH occupancy of our approach compared to SC-PtM. Our scheme consistently uses significantly less resources compared to SC-PtM, even when considering the extreme case of a 200 RFs scheduling period. Please note that the figure is split in two parts for better clarity.

Comparatively, SC-PtM uses significantly more resources, even when considering the extreme case of a 200 RFs scheduling period. Of course, such an extreme case comes at the expense of significantly increased delivery time (figure 6.11). When striking a balance between occupancy and delivery time (50 RFs), SC-PtM uses two times more resources than our method with 8 repetitions, or four times more resources with 2 repetitions. It is apparent that the subscription-based scheme of SC-PtM, wastes a significant amount of resources just to transmit control information, when these resources could efficiently be used for actual data transmission (multicast or unicast).

Finally, we demonstrate the behaviour of each approach as the background traffic increases (figure 6.14). For simplicity, we only present one set of parameters (normal CP with 2 repetitions, and 1 SF minimum guarantee for MFG), but the same trend can be observed in all cases. With respect to the firmware delivery time, we can see that our method (for both MP and MFG transmission strategies) scales better than SC-PtM as the load increases. This can be explained by the fact that SC-PtM needs to use part of the available resources for multicast control information, which results in fewer available resources for the multicast transmission itself. An important thing to

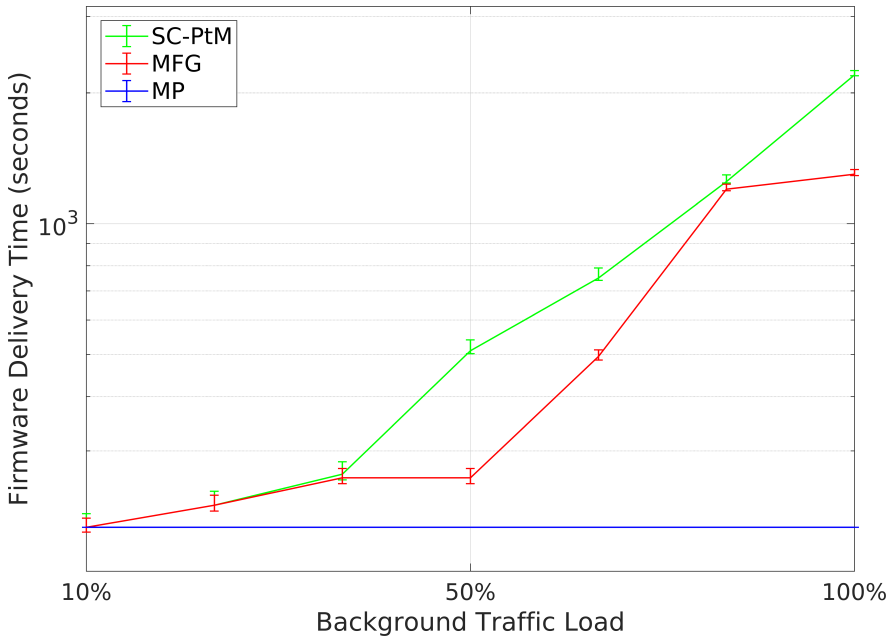


Figure 6.14: **Firmware delivery time with different background traffic loads:** The figure depicts the firmware delivery time with variable background traffic loads. The benefits of the minimum guarantee of MFG is apparent under high background load conditions (90%) where it is able to deliver the multicast content faster.

note however, is that the reduced firmware delivery time does not come at the expense of the background traffic. In fact, our method produces equal or better ACK average delivery times in all cases (figure 6.15). This is true even for the MP transmission strategy, that prioritises the multicast transmissions, as we are able to fully utilise all available resources for data transmission (either multicast or unicast).

6.4 Device Grouping and Synchronisation

Until this point we have assumed that devices that must receive the same multicast content are all awake and connected to the network at the time of the multicast transmission. However, this is not always the case as different devices may have different periodicities and DRX/eDRX cycles. Such synchronisation is essential in the context of NB-IoT due to the limited bandwidth, as multiple transmissions of the same context would severely deteriorate the overall performance of the system.

In this section we present three different mechanisms to achieve grouping and synchronisation of devices for multicast content reception, and we experimentally evaluate their performance under realistic operating conditions. Each mechanism makes differ-

Figure 6.15: **ACK delivery time with different background traffic loads:** The figure depicts the impact of the SC-MCCH in the delivery of the application ACK messages. Our proposed approach performs similarly to SC-PtM for low background traffic loads, but outperforms it as the background load increases.

ent trade-offs between three important aspects: (i) the device energy consumption, (ii) the resource usage, and (iii) the compliance with the existing NB-IoT standards.

Our first mechanism respects the DRX/eDRX cycle of the devices and aims to transmit the multicast data with the fewest transmissions required to cover all devices. This approach is standards compliant, and has the lowest energy consumption at the device side, at the expense of increased bandwidth usage due to multiple transmissions. Our second mechanism temporarily modifies the DRX/eDRX cycle of the devices in order to synchronise them at the time of multicast transmission, so that only a single transmission is needed. This approach minimises the bandwidth usage and is also standards compliant, but increases the energy consumption of the devices. Our third mechanism uses a small modification to the paging protocol to notify the devices well in advance of the time of the multicast transmission, so that the devices can wake up to receive it without the need for further signalling. This minimises both energy consumption and bandwidth usage, but is not compliant with the NB-IoT standard.

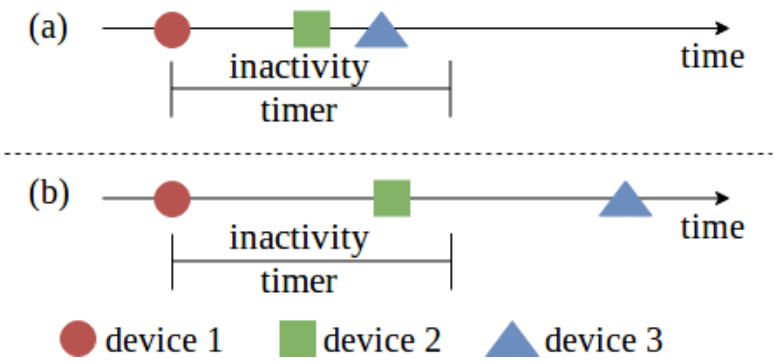


Figure 6.16: **Example of POs and inactivity timer:** The figure depicts three devices and their POs. In subfigure (a) the POs of devices 2 and 3 are within the range of the T_I from the PO of device 1 so one multicast transmission will cover all of them. In subfigure (b) only device 2 is within range from device 1 so a second multicast transmission is required for device 3.

6.4.1 Grouping Mechanisms

Ideally, if multiple devices could be grouped together and get synchronised so that they have a paging occasion (PO) within the inactivity timer (T_I) before a multicast transmission, they will all be able to receive the multicast data with a single transmission, minimising both the network resource usage and the device energy consumption. This is because none of the devices will go back to sleep before the others have been paged, and they will all be active at the multicast transmission time. Otherwise, the inactivity timer of some devices will expire before the other devices can be paged. An example is depicted in figure 6.16. In figure 6.16(a), devices 2 and 3 are within the range of the inactivity timer T_I from device 1 so one multicast transmission will cover all of them. However, in figure 6.16(b), only device 2 is within range from device 1 so either a second multicast transmission is required for device 3, or both devices 1 and 2 will need to remain connected until device 3 can be paged, thus increasing their energy consumption.

6.4.1.1 DRX Respecting - Standards Compliant (DR-SC)

In this mechanism, the DRX/eDRX cycle of the devices is respected, and multiple devices share a multicast transmission only if their POs happen to be closer in time than T_I. As such, the devices do not use any more energy than what they would have under normal operation, aside from the multicast data transmission itself. However, as there is no guarantee that multiple devices will coincide at the multicast transmission time,

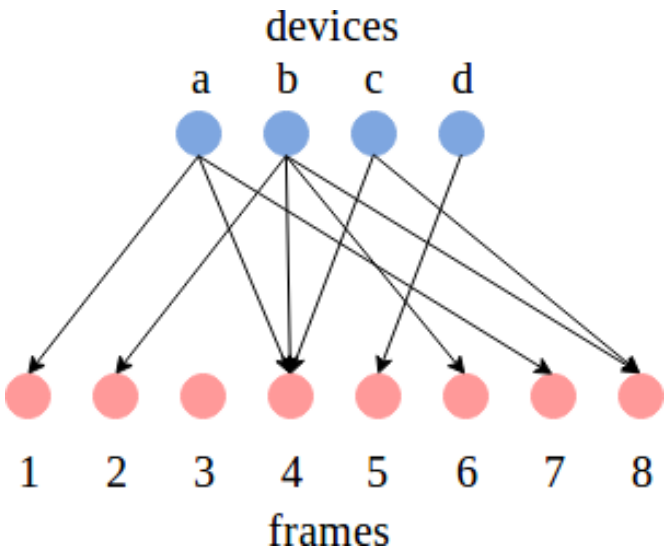


Figure 6.17: **Set cover problem:** The figure depicts the POs of devices as the set cover problem. We formulate the POs in time as a bipartite graph. Each edge from a device to a frame indicates that the device has a PO on that frame. For simplicity, and without loss of generality, we consider the inactivity timer T_I to be a single frame here. Finding the minimum set of frames that would cover all devices corresponds to the set cover problem, which is known to be NP-hard [213]. Here, the optimal solution would be frames 4 and 5.

numerous transmissions will most likely be needed to cover all devices (section 6.5.2), leading to higher network resource usage.

In this scenario, we would ideally like to find the minimum number of transmissions needed to cover all devices, so that the bandwidth usage is minimised. We can formulate the POs in time as a bipartite graph of devices and frames, where each edge indicates that the device has a PO on that frame (figure 6.17). Finding the minimum set of frames that would cover all devices corresponds to the set cover problem, which is a known NP-hard [213] problem. Therefore, we follow an approximate solution, given a greedy set selection approach [214].

More specifically, we begin by finding the period of time t_0 of length T_I that contains the maximum number of POs of different, non-updated devices. As each DRX/eDRX cycle is exactly twice as long as the previous one (section 2.1.3.2.7 and section 2.3.3), the PO occurrence patterns will start repeating after a period of time twice as long as the largest DRX/eDRX. Therefore, we only need to search this length of time for t_0, reducing the complexity. A multicast transmission is then scheduled at a time after the last frame of t_0, and the covered devices are then considered up-

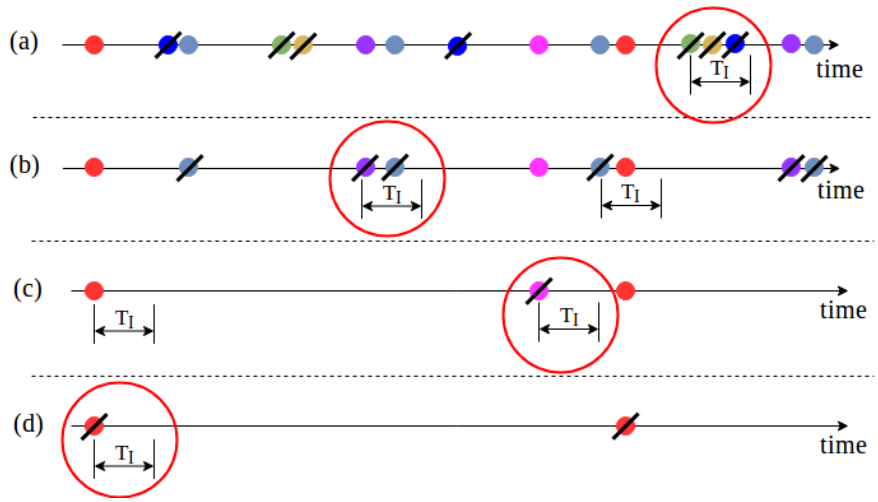


Figure 6.18: **DR-SC mechanism:** The figure depicts 7 devices (denoted with different colors) with different POs. Initially, none of the devices has received the multicast data. In step (a) we find the period T_I that contains the larger number of devices (3). We then remove these devices from our list and proceed to find the next T_I with the max number of devices. In step (b) we have 2 possible times so we pick one of them randomly. We follow the same process until all devices receive the data.

dated. The process is then iteratively repeated until no non-updated devices remain (figure 6.18).

6.4.1.2 DRX Adjusting - Standards Compliant (DA-SC)

Our second mechanism seeks to proactively change the DRX/eDRX cycle of some devices for a limited time, so that all devices will have a PO within T_I before the multicast transmission, and receive it simultaneously. Although the DRX/eDRX length is usually negotiated between the device and the BS, the BS can unilaterally decide on the cycle value, which is something that can be used to forcibly synchronise the devices. This has the benefit of minimising the number of required multicast transmissions to just a single one, making good use of the limited bandwidth available in NB-IoT. However, devices will need to use more energy, as they will likely be using a smaller DRX/eDRX cycle than their original. Since a DRX/eDRX value is exactly twice as long as the previous one (section 2.3.4), decreasing the DRX/eDRX cycle respects the original periodicity of the device.

In more detail, the BS chooses a time t to transmit the multicast data. The time t should be at least $2max_{DRX}$ where max_{DRX} is the longest DRX/eDRX cycle of the de-

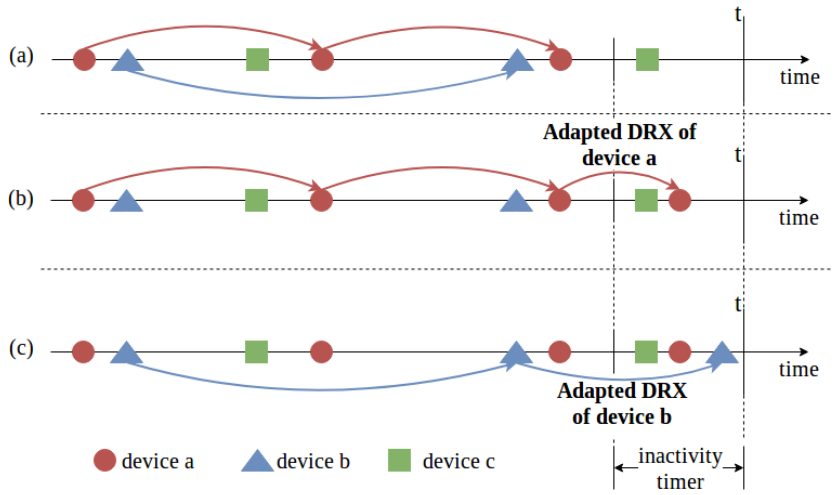


Figure 6.19: **DA-SC mechanism:** The figure depicts the adaptation of DRX cycles. Subfigure (a) shows the devices with their original DRX cycles. Device c does not need adapting as it has a PO within $[t - T_I, t)$. In steps (b) and (c) the DRX cycles of the devices a and b are adapted, as they dont have any POs in $[t - T_I, t)$. Please note that the adaptation happens on the last PO before $t - T_I$ so that the energy consumption introduced is kept to a minimum.

vices to receive the multicast data, so that there will be at least one PO of every device before t. The BS then finds each device that does not have a future PO within $[t - T_I, t)$, and decreases its DRX/eDRX cycle to the longest value that results in a PO within that time period. Since the shortest DRX cycle (0.32 seconds) is much shorter than T_I, such a DRX/eDRX cycle is guaranteed to exist. To keep the energy consumption introduced by the adapted DRX/eDRX cycle as low as possible, the adaptation happens in the last PO before $t - T_I$ (figure 6.19).

To enforce the new DRX/eDRX cycle, the BS pages the devices, which then proceed to connect to the network through the typical RA process (section 2.1.3, and receives the new DRX/eDRX value in the RRC Connection Reconfiguration message. The BS then instructs the device to switch back to sleep immediately (without waiting the inactivity timer to expire), using the RRC Connection Release procedure, to reduce the device uptime and resulting energy consumption. After the multicast transmission, the original DRX/eDRX cycles are restored with an additional RRC Connection Reconfiguration message.

6.4.1.3 DRX Respecting - Standards Incompliant (DR-SI)

Our third mechanism uses a new, non-critical extension to the existing paging message, named *mltc_Tx*, in order to notify the devices in advance about an imminent multicast transmission. This allows the devices to retain their preferred DRX/eDRX cycles as in DR-SC, maintaining the normal energy usage, while delivering the multicast content with a single transmission, as in DA-SC. However, although the extension to the paging message is simple, this solution is no longer compliant with the current NB-IoT standards.

In more detail, whenever the BS has multicast data to transmit it sends an extended paging message to the devices that do not have a PO within $[t - T_I, t)$. The paging message contains the new non-critical extension which comprises of the device identity and the time remaining until the multicast transmission. The device identity is only present in the non-critical extension and not in the *PagingRecordList* field of the paging message, so devices can distinguish between a paging to receive downlink unicast and multicast transmissions. As the device is not paged to receive downlink data, it does not need to wake up and connect to the network, keeping the energy consumption similar to that in normal operation.

Upon receiving the paging message, the device selects a random time value between $[t - T_I, t)$ and sets a new timer (T322) to expire at the selected time. When T322 expires, the device wakes up and connects to the network to receive the multicast data. Finally, to indicate that the connection is made for the multicast transmission and not for unicast downlink data, the device sets the *EstablishementCause* field of the RRC Connection Request message to the new value of *mltc_Rx*.

6.5 Grouping and Synchronisation Evaluation

6.5.1 Experimental Setup

To assess the impact of each grouping mechanism in terms of energy consumption and bandwidth usage, we conducted a thorough experimental evaluation, considering a single BS scenario serving a large number of NB-IoT devices. The effect on the bandwidth usage is dependent on the number of required transmissions to cover all devices for DR-SC. While the DA-SC and DR-SI approaches only need a single transmission, the DR-SC approach requires a variable number transmissions (section 6.5.2) depending on how many devices happen to be synchronised. Therefore, we use the number of

multicast transmissions as a proxy for the bandwidth utilisation. As the probability of devices being synchronised increases as the number of devices increases, we evaluated a varying number of devices (100 to 1000) to receive the same multicast content, and averaged the results over 100 runs.

Specific energy consumption values are hard to estimate, as they are device specific and may change as technology evolves. However, increased uptime will lead to increased energy consumption irrespectively of the type of the device. Therefore, we measure the relative increase of uptime compared to what would be required for unicast transmission (i.e. each device receiving the multicast data based on its own DRX/eDRX and without waiting for other devices) as a proxy for the energy consumption. Since unicast transmission would not introduce any additional processes, it is the most efficient way to receive the data in terms of energy consumption from the device perspective. Furthermore, we consider the uptime spent during the PO and during an active connection separately, as the energy usage in the latter is significantly higher [215, 216].

Similarly to the group communications approach, we simulate a single cell with 55000 NB-IoT devices following realistic traffic patterns based on [209]. For each device, we randomly generated an initial DRX/eDRX value. Furthermore, we show results for multicast data of three different sizes (100KB, 1MB and 10MB), which we believe covers the spectrum of typical firmware updates. All experiments were implemented on a custom simulator written in Matlab.

6.5.2 Results

6.5.2.1 Device Uptime

First, we assessed the increase in uptime that each approach incurs to a device. This increase can be due to additional paging (DA-SC), DRX/eDRX adjustment (DA-SC), or setting additional timers (DR-SI). Figure 6.20 shows the relative increase of uptime compared to unicast transmission for different sizes of multicast data. In particular, top-figure 6.20 shows the uptime while the device checks its POs for paging messages, while figure 6.20(b) depicts the relative uptime in the connected mode. As we can see, the DR-SC approach requires exactly the same uptime as the unicast approach, as no extra POs are needed. The DA-SC induces a minor increase as additional POs are used with the adapted DRX/eDRX, while the DR-SI introduces a negligible increase as only the reception of the paging message is required. Therefore, the energy consumption in

the DR-SI is kept similar to that of unicast but with a single multicast transmission.

Figure 6.20 (bottom) shows the uptime while the device performs the RA process, waiting for the multicast transmission to begin, and when receiving the multicast data. This is a more important metric than the PO uptime, as the energy consumed during these operations is an order of magnitude greater [215, 216]. In this case, both DR-SC and DR-SI have slightly higher uptimes than unicast transmission, as they need to wait for $T_I/2$ on average for the multicast transmission to start. DA-SC has the longest uptime, as it also needs to go through the RA process in order to connect to the BS and get the DRX/eDRX cycle adjusted. Overall however, compared to the actual time spent on receiving the multicast data, the relative increase in uptime is very low (bottom-figure 6.20). In practice, the overhead introduced by the signalling of DA-SC becomes practically negligible as the multicast data size gets above 1MB.

6.5.2.2 Number of Multicast Transmissions

Finally, we assess the number of multicast transmissions required to update all devices. While DA-SC and DR-SI mechanisms only require a single multicast transmission by design, DR-SC will typically require multiple transmissions as there is no guarantee that the devices will have synchronised POs. Figure 6.21 shows the average number of multicast transmissions needed as the number of devices to receive the multicast data increases. Larger numbers generally lead to a higher probability that multiple devices will be synchronised, so the number of required transmissions increases slower than the number of devices. However, the number of required transmissions is significant, and even for 1000 devices it is only 60% more bandwidth efficient than using unicast transmissions. Given that NB-IoT already operates on limited resources, this approach can seriously affect the existing traffic and result in significant performance degradation and thus, it is not deemed a practical grouping mechanism for multicast transmissions in NB-IoT.

6.6 Conclusions

In this chapter, we focused on group-communications in NB-IoT, and we presented a set of enhancements to the existing eMBMS framework, as well as new features to be supported by the existing procedures, with particular interest on customer-driven eMBMS group formation and content delivery. We also presented two transmission

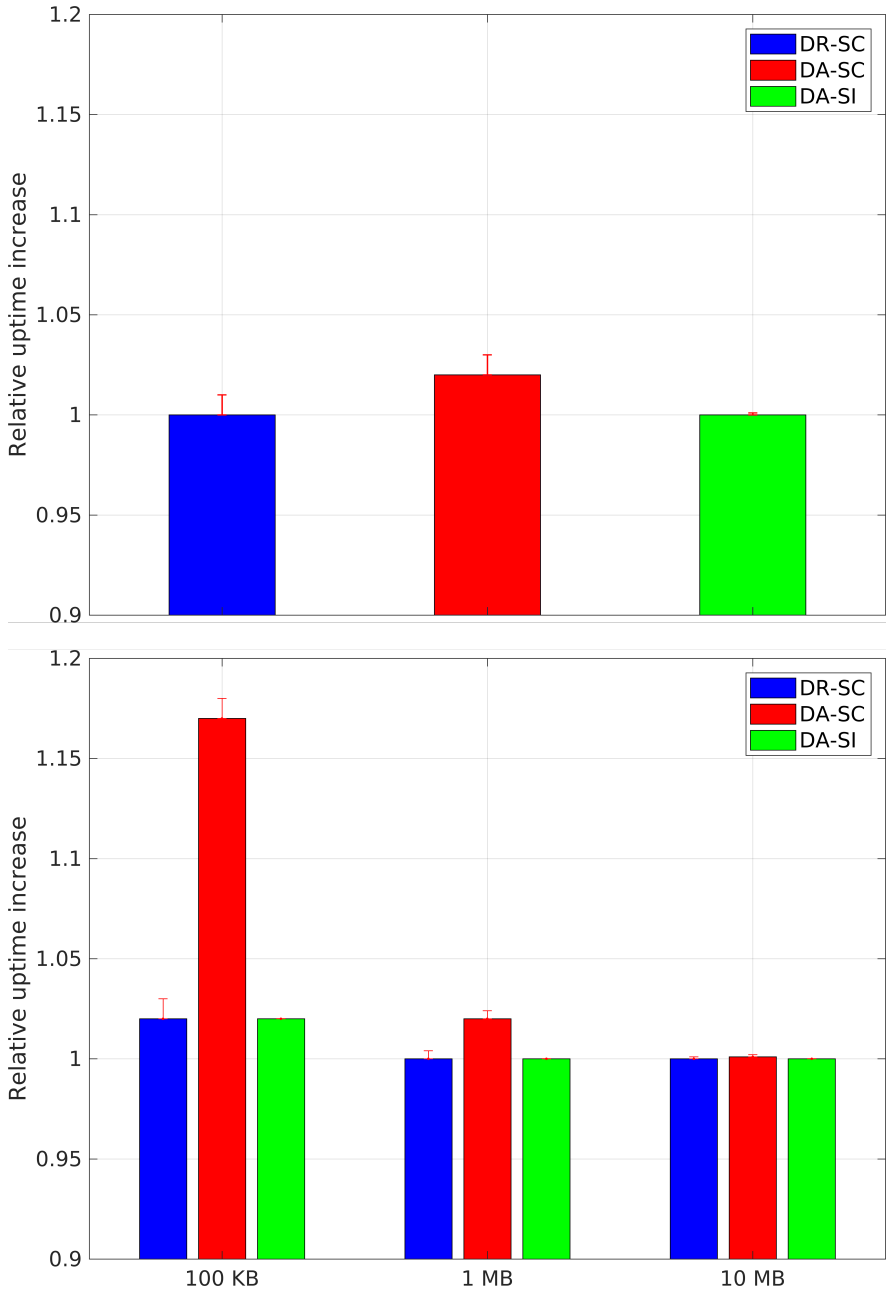


Figure 6.20: **Relative uptime increase of synchronisation techniques:** The figure depicts the relative uptime increase of the three synchronisation techniques compared to unicast transmission. The top subfigure depicts the uptime during the PO, while the bottom subfigure depicts the uptime in connected mode for different sizes of multicast data.

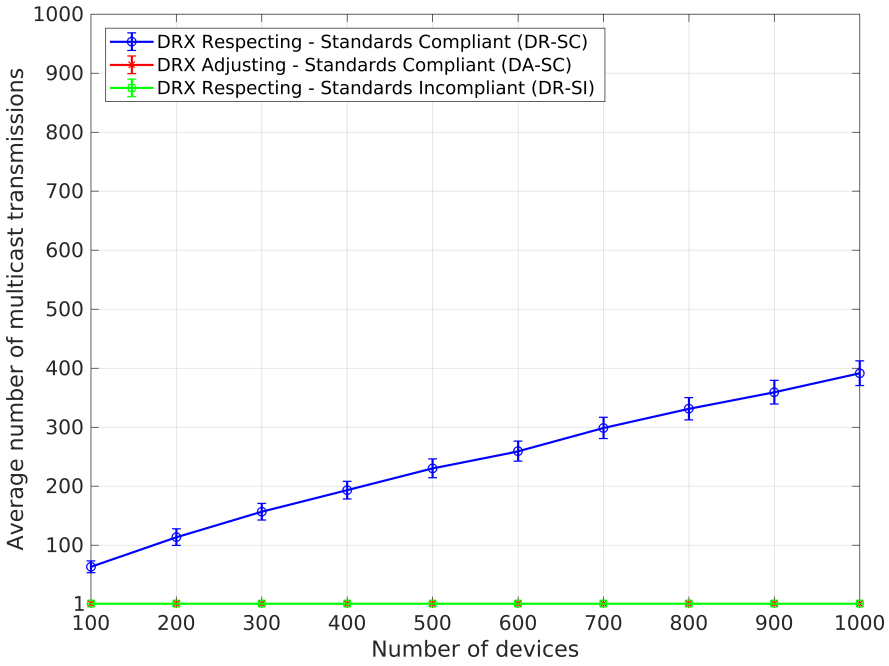


Figure 6.21: **Number of multicast transmissions:** The figure depicts the average number of multicast transmissions required to update all devices over 100 runs. When few devices receive the multicast data the number of multicast transmissions is around 50% of the number of devices. As the number of devices increase, the number of multicast transmissions falls to around 40% of the number of devices.

strategies (MFG and MP) aimed at limiting the impact of multicast traffic on unicast transmissions, and compared them against the SC-PtM approach currently proposed by 3GPP. Our results show that MFG is a valid approach when the network needs to support large numbers of multicast sessions, as it stops the unicast traffic from overloading the downlink channel. In contrast, the MP approach is more suitable for infrequent multicast sessions, as it allows the multicast content to be delivered quickly without drastically impacting the unicast traffic.

Further, we explored three different mechanisms to achieve device grouping and synchronisation for multicast transmissions in NB-IoT: DR-SC, DA-SC and DR-SI. Each of the mechanisms makes different trade-offs between the device energy consumption, resource usage, and standards compliance. To assess their performance we conducted a thorough experimental evaluation under realistic traffic conditions. Our results show that the DR-SC mechanism results in very high resource usage, and it is not much more efficient than delivering the data with unicast transmissions. Therefore it is not practical for NB-IoT deployments, where the available bandwidth is already limited. The DR-SI mechanism has excellent performance both in terms of energy consumption at the device side, as well as bandwidth utilisation at the network side. However, it requires protocol changes and may face deployment/adoption challenges. Finally, the DA-SC mechanism introduces slightly higher energy usage compared to DR-SI due to the adaptation of the devices' DRX/eDRX cycles, but this increase is very small compared to the actual time spent receiving the multicast data. Given the fact that it does not require any protocol changes, this mechanism offers the best trade-off among the three mechanisms for the target use case of distributing firmware updates.

Chapter 7

Energy Efficiency in NB-IoT

In this chapter we focus on the energy efficiency from the perspective of the IoT device. In particular, we explore NB-IoT devices specifically, as these have the strictest energy requirements (more than 10 years battery life on a single charge, and overall cost less than $5 [14]), and we show that, despite recent enhancements (e.g. eDRX, PSM - section 2.3.4), the current NB-IoT protocols are not efficient enough to achieve the desired battery life goal. We argue that many of the inefficiencies of the current protocols stem from the fact that they have been directly inherited by the 4G and 5G designs, which traditionally focused on HTC devices (e.g. phones, tablets), without major concern for energy efficiency, as they can be recharged often. Additionally, they have been optimised to cater to HTC traffic patterns and use cases (long connections uniformly spread over time, mainly downlink traffic). In contrast, NB-IoT devices typically exhibit very short connections in frequent, periodic intervals [14], and existing procedures can incur a disproportionate energy overhead in relation to the actual data communicated. It is noteworthy however, that a large part of our energy consumption analysis is common for other types of IoT devices, and thus, some of the results should generalise.

We begin by performing a thorough experimental measurement of the power consumption of each individual operation that a NB-IoT device performs under normal use (e.g., RA, Attach, data encryption, data transmission), using three different commercial NB-IoT devices. These operation-specific measurements offer significantly greater insight than prior works that only measure the total power consumption [21], or just the data exchange energy cost [217, 196]. As such, they allow us to unearth potentially inefficient areas of the NB-IoT protocols, and can guide us towards effective optimisations. They also show a large deviation from the energy consumption assump-

tions published by 3GPP [19], which can greatly affect studies that rely on them (e.g. [21, 20]) *To the best of our knowledge, this is the first work for NB-IoT devices that measures each operation in isolation at the time of writing.*

Building on the above energy characterisation, we present an NB-IoT energy consumption model, which we use to simumate the battery life of a device under realistic traffic conditions [14], factoring in collisions and signal degradation. This gives us an estimate of the battery capacity requirements to meet the 10 year goal for different coverage scenarios, and we find that it is far from realisable with current practices and previously proposed approaches, given the $5 cost constraint. We also use our model to compare against previous works that do not take all operations into account (e.g., [20]), and assess the overall energy consumption difference. We show that these operations ignored can have a significant energy cost, and it is imperative they are considered and optimised to lower the overall device energy consumption.

Finally, we propose two novel mechanisms that exploit the stationary and periodic nature of NB-IoT devices to substantially reduce device energy consumption, while at the same time free up network resources. Furthermore, we discuss a set of best practices under the existing protocols that device vendors and network operators should consider in order to maximise battery life.

The work presented in this chapter is under submission at the 27th IEEE International Conference on Network Protocols (ICNP) 2019.

7.1 Device Measurements

7.1.1 Experimental Setup

Initially, we measure the energy consumption of three popular NB-IoT development kits from Pycom (GPy) [218], Quectel (BC95) [219], and Sodaq (SARA-N2) [220]. Note that the Quectel and Sodaq devices are not complete NB-IoT solutions and require a separate Micro-Controller Unit (MCU) to operate. For this, we used an Arduino Uno board [221] as, at the time of writing, it was the one with the lowest power consumption among the various Arduino MCUs. The GPy device was used with the MCU unit proposed and manufactured by Pycom [222]. In order to get accurate power consumption measurements for the various operations, we switched off all LEDs at boot time (on all devices), and any WiFi/Bluetooth features if available (Pycom).

To measure the energy consumption of the different network processes (RA, Attach

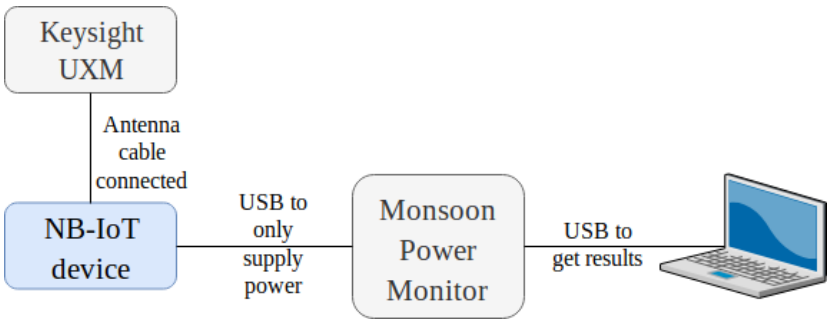


Figure 7.1: **Setup for energy measuring experiments:** The figure depicts the setup for measuring the energy consumption of different operations using the Power Monitor from Monsoon Solutions, and the E7515A UXM Wireless Test Set from Keysight. The devices were powered through the Power Monitor, using a USB cable which was configured to only supply power. To measure the energy of any network operations, we connected an antenna cable directly from the UXM box to the device.

etc.) we used the E7515A UXM Wireless Test Set [223] by Keysight Technologies, which implements a fully compliant NB-IoT BS. In terms of the NB-IoT configuration, the UXM box was set to use an in-band deployment over a 10MHz channel, with 15KHz subcarrier spacing, and QPSK/ BPSK modulation in the downlink/uplink respectively. For operations outwith the actual communication (e.g., message generation, key generation, encryption etc.), we used a Power Monitor (FTA22J) [224] from Monsoon Solutions. The devices were powered through the Power Monitor, using a USB cable which was configured to only supply power (figure 7.1).

In these experiments we measure the energy consumption while the devices are in any of the three performance states: (i) deep sleep, (ii) light sleep and (iii) working. The light sleep and deep sleep are states during which the device has limited or almost no energy consumption respectively, and correspond to the idle and PSM states of 3GPP [225] (section 2.1.3), while working state is the state during which the device generates data and communicates with the network. For this reason, we separately measure the following:

- RA process including the cell search and decoding of MIB/SIBs,
- Attach process including AKA and PDN connectivity processes (for cases when CP optimisation is not used),
- exchange of application data (including any required scheduling requests, reception of control data for ACK, encryption/decryption) (for cases when CP optimi-

sation is not used),

- *(d)* IMSI encryption (i.e. generate public/private keys, perform Diffie-Hellman, symmetrically encrypt IMSI, section2.1.5.1))

- active waiting with C-DRX of 10 and 30-second inactivity timer

For these measurements, we assume a NB-IoT specific traffic pattern, where devices send 200 bytes of application data and receive a 140 bytes acknowledgement every 5 minutes. We consider a broad range of traffic patterns in subsequent sections.

Strong encryption mechanisms can be very expensive for NB-IoT devices in terms of energy consumption, so the choice of security procedures can have a noticeable impact on their battery life. For symmetric encryption we measured the power consumption of the EEA1/EIA1, EEA2/EIA2 and EEA3/EIA3 algorithms which are recommended by 3GPP [226], and are based on the SNOW3G, AES, and ZUC algorithms respectively. For the IMSI encryption, we follow the approach of [39] and use the Diffie-Hellman key exchange algorithm implementing a Curve25519 elliptic curve. To generate asymmetric keys, we used public implementations of the popular RSA and El-Gamal algorithms [227, 228] with 1024-bit keys for both. Each data point in our results is the average of 10 different runs.

7.1.2 Results

First, we measure the average power consumption during the three performance states (figure 7.2), assuming a complete connection cycle (i.e. connection establishment, data exchange, active waiting, light sleep and deep sleep). Overall, all three devices have similar power usage in the working state, with the major differences occurring in the deep sleep state. The energy consumption in the working state can be orders of magnitude greater than either of the sleep states. Perhaps surprisingly, although the light sleep state is more efficient than the working state, it still uses 3 orders of magnitude more power than the deep sleep state. This indicates that in cases where latency for network originated data is not critical, there would be substantial gains if the devices switched to the deep sleep state earlier (section 7.3).

We then examine the energy consumption of the operations related to the security framework. As 3GPP allows for alternatives both for symmetric and asymmetric encryption [35], we examine the energy consumption of the three recommended symmetric encryption algorithms (figure 7.3), and two popular asymmetric encryption

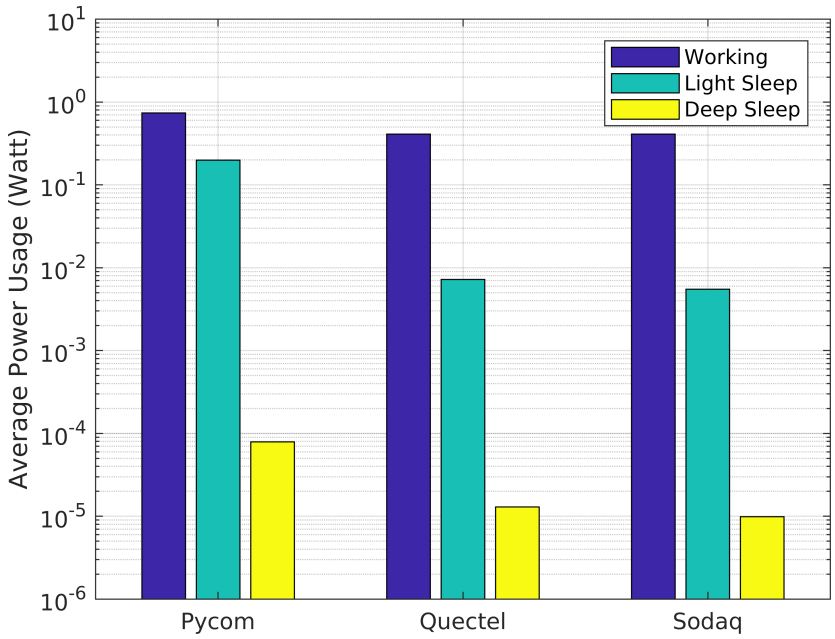


Figure 7.2: **Power usage in the different performance states and the different devices:** The figure depicts the power usage in Watts for the different performance states and the different devices. Surprisingly, the light sleep state still uses 3 orders of magnitude more power than the deep sleep state.

algorithms (El-Gamal, RSA) (figure 7.3) in isolation (section 2.1.5.1). For reference, we also measured the energy required for the Diffie-Hellman protocol.

We observe that among the symmetric algorithms (figure 7.3), EEA2/EIA2 is the most energy efficient, followed closely by EEA3/EIA3. Since they are also equally strong in terms of security, the choice between EEA2/EIA2 and EEA3/EIA3 has little impact on the energy consumption of the device. When comparing the asymmetric algorithms (figure 7.4), El-Gamal is more efficient for decryption and key generation but is significantly more expensive for encryption compared to RSA. As only the key generation part of these algorithms (sec. 2.1.5.1) is used, El-Gamal emerges as the most efficient choice.

Further, we investigate the energy consumption of the various operations in the working state (figure 7.5), to get insight on potential areas for improvement. In this experiment we used the EEA2/EIA2 algorithm for symmetric encryption and integrity protection, and the El-Gamal algorithm for the IMSI encryption, as these yielded the lowest energy cost. Notably, our results show that the energy consumption values on real devices are considerably greater compared to the values considered by 3GPP for NB-IoT [19], based on which the 10 year life goal was set.

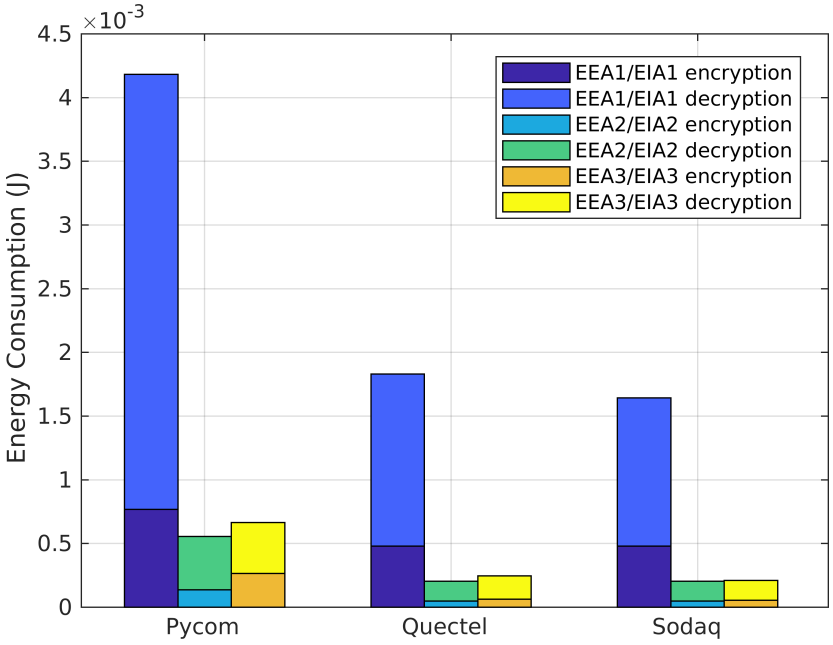


Figure 7.3: **Energy consumption of symmetric encryption algorithms:** The figure depicts the energy consumption in Joules of the three different symmetric algorithms currently proposed by 3GPP. EEA2/EIA2 is the most energy efficient symmetric encryption algorithm, followed closely by EEA3/EIA3. Since they are also equally strong in terms of security, the choice between EEA2/EIA2 and EEA3/EIA3 has little impact on the energy consumption of the device.

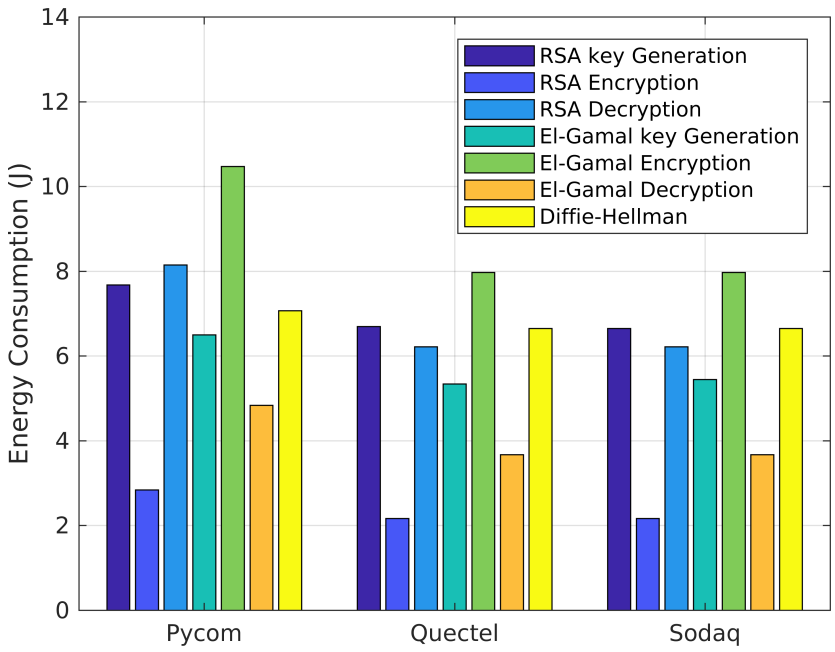


Figure 7.4: **Energy consumption of asymmetric encryption algorithms:** The figure depicts the energy consumption of RSA, El-Gamal and Diffie-Hellman algorithms in Joules. Key generation and decryption using El-Gamal is more efficient than RSA. However, RSA has a significant advantage in encryption. Diffie-Hellman also requires a lot of energy that is comparable to RSA key generation and decryption.

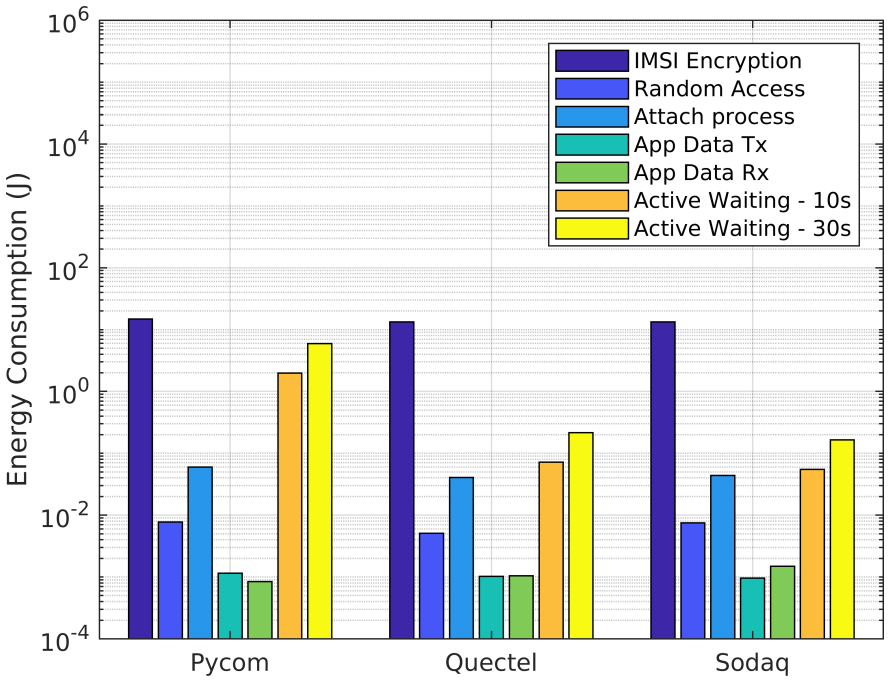


Figure 7.5: **Energy consumption for operations in working state:** The figure depicts the energy consumption in Joules for the operations in the working state. The encryption of the IMSI is the most expensive operation, but the RA, Attach and Active waiting also require significant energy. In active waiting we show the energy consumption when 10 and 30 seconds of inactivity timer are used. Note that the y-axis is in logarithmic scale for better clarity.

When looking at each operation individually, we see that the energy consumption for the actual data transmission and reception is orders of magnitude lower than for operations like the RA, Attach and Active Waiting. This is also reflected in the pie chart (figure 7.6) showing the proportion of time required for each of the operations in the working state with a $10s$ inactivity timer. In this experiment we have excluded the IMSI encryption as it is optional in NB-IoT and can dominate the time spent in the working state. Crucially, these results show that a holistic and fine-grained view of NB-IoT device energy consumption characterisation leads to significantly different battery life estimates compared to prior works [21, 20].

7.2 Battery Life Expectancy

In this section, we examine the expected battery life of NB-IoT devices according to our energy consumption measurements. To do so, we model the energy consumption

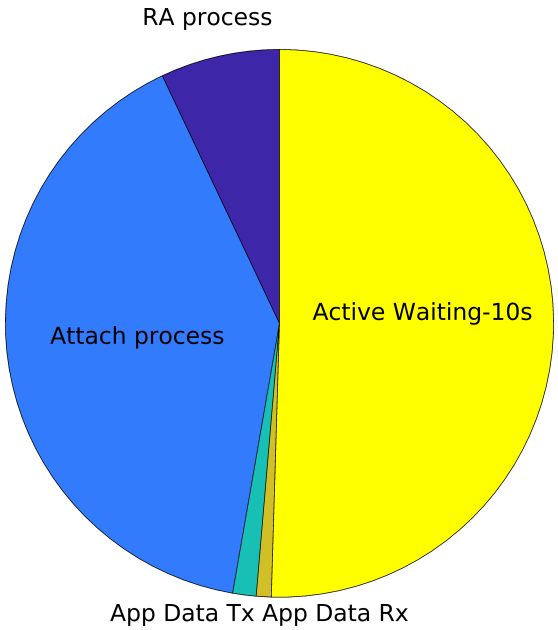


Figure 7.6: **Proportion of time spent for different operations in the working state:** The figure depicts the proportion of time that is spent for each operation in the working state, assuming an inactivity timer of 10 seconds.

of a NB-IoT device for a cycle of operation, under normal conditions. We then use our energy model along with realistic traffic patterns [14], to simulate the overall energy consumption. This allows us to take into account factors such as collisions during the RA process, which happen under congestion, and gives us a realistic estimate of the actual energy consumption of the device outside lab conditions. Using this simulation we can estimate the required battery capacity in order to achieve a 10 year lifetime [48], and assess whether this lifetime requirement is realistic for the current commercial consumer technology.

7.2.1 Energy Model

We define a period P as the time period between two successive instances that a device wished to transmit data. The total energy consumption of the device during a single period P is:

$$E_P = E_w + E_l + E_d \tag{7.1}$$

where E_w is the energy spent whenever the device wishes to transmit new data, E_l is the energy consumed during the light sleep state and E_d is the energy consumption

during the deep sleep state. Therefore, the energy consumption E of the device over a day:

$$E = \begin{cases} \frac{T}{P}E_P + E_{imsi} & P <= t_{renew} \\ \frac{T}{P}E_P + \frac{T}{P}E_{imsi} & P > t_{renew} \end{cases} \tag{7.2}$$

where T is the time within a day, E_{imsi} is the energy consumption to perform the IMSI encryption once, and t_{renew} is the deletion frequency of the security context (section 2.1.5). The energy consumption of the working state E_w is equal to:

$$E_w = E_{RA} + E_{At} + E_{app} + E_{AW} \tag{7.3}$$

where E_{RA} is the energy consumption for the completion of the RA process, and can be defined as:

$$E_{RA} = \varepsilon_{RA} * \overline{R} \tag{7.4}$$

with ε_{RA} being the energy consumption of an RA process without collisions, and $\overline{R}$ being the average number of connection attempts required to establish a connection.

E_{At} is the energy consumption required for completion of an attach process of average length[1] when the CP optimisation (section 2.3.3.3) is not used, and includes the AKA and the PDN connectivity procedures (section 2.1.3). E_{app} is the total energy consumed for the transmission and reception of application data, including the data generation, scheduling requests, generation and appendage of the required headers, and transmission/reception including all repetitions. Finally, E_{AW} is the energy consumed during the active waiting state.

E_L is the energy consumption during light sleep, and is equal to:

$$E_l = t_l U_l \tag{7.5}$$

where U_l is the power usage in the light sleep state and t_l is the time spent in the light sleep state during a period P. Similarly, E_d is the energy consumed during deep sleep during a period P, and equals:

$$E_d = (P - t_l - t_w) * U_d + E_{TAU} \tag{7.6}$$

where U_d is the power usage of the device during deep sleep and t_w is the time spent in the working state during a period P. E_{TAU} is the energy consumed during the Tracking

[1] According to [51] the Attach process must be completed within 15 seconds at most.

Area Update (TAU) process which is always executed at the end of the PSM cycle, and can be defined as:

$$E_{TAU} = E_{RA} + \varepsilon_{Tx} + \varepsilon_{Rx} \tag{7.7}$$

where ε_{Tx} and ε_{Rx} is the energy consumption of a single NPUSCH transmission and NPDCCH reception respectively including all repetitions, for the TAU request and TAU accept messages. The E_{RA} is also included, as devices have to go through the RA process before being able to update their location.

7.2.2 Simulation Setup

Our simulations are based on our energy model (section 7.2.1), and follow realistic traffic patterns for a NB-IoT cell in a dense urban environment [229, 16]. We simulated 50000 devices uniformly distributed in a cell of 500m radius, which is considered a typical-sized urban cell. We assume that 80% of the devices are periodic with variable application periodicities, ranging from 5 minutes to 24 hours. The remaining 20% are event-driven with no periodicity. In each connection, devices transmit a single application packet of 200 bytes, and receive an application ACK of 140 bytes. We also assumed that devices employ the CP optimisation (section 2.3). In terms of network configuration our parameters are summarised in table 7.1.

In all of our experiments we used the most optimistic configuration from the energy consumption perspective, to set up a lower limit for the required battery capacity. For symmetric encryption, we used the EEA2/EIA2 algorithm, as it is the most efficient one (section 7.1), while for the IMSI encryption we used the El-Gamal algorithm for the generation of ephemeral public/private keys, as it is the one that incurs the lowest energy consumption among the asymmetric algorithms (section 7.1).

We assumed that after the transmission of application data, devices remain in active waiting for 10 seconds (inactivity timer), during which, they employ the C-DRX (section 2.1.3.2.7). After the expiration of the inactivity timer, the devices switch to the light sleep state (idle) for 2 minutes, during which they apply I-DRX. We also assumed that the PSM feature is enabled. For all periodic devices with the exception of the ones with 5-min periodicity, we assumed that the PSM duration is $1/5$ of their application periodicity, while for event-driven devices we assumed a randomly generated PSM duration of up to 120 minutes, after which time the network releases a device's connection and requests it to re-attach on its next transmission.

Finally, for all devices we setup a new security context on their initial network at-

Variable	Value
UE transmit power on NPUSCH	Based on subclause 16.2.1.1.1 [49])
Number of subcarriers (M_{NPUSCH}) in uplink	12
UE transmit power on NPRACH	Based on subclause 16.3.1 [49]
Time spent in NPUSCH	0.93 ms
Time spent in NPDCCH	0.22 ms
C-DRX cycle	2.56 s
I-DRX cycle	5.12 s
PSM Duration	$app\_periodicity/5$ (periodic devices) random PSM (non-periodic devices)
Subcarrier spacing (KHz)	15
Coverage Levels	Normal / Robust / Extreme
NPDCCH repetitions	1 / 1024 / 2048
NPDSCH repetitions	1 / 1024 / 2048
NPUSCH repetitions	2 / 64 / 128
NPRACH repetitions	2
Modulation	QPSK
NPDCCH periodicity	$T_{NPDCCH} = R_{MAX} * G$ [49] $R_{MAX} = 1$ $G = 32$
RACH periodicity	40 ms [230]

Table 7.1: **Simulation configuration parameters:** The table details the network configuration parameters for our simulations.

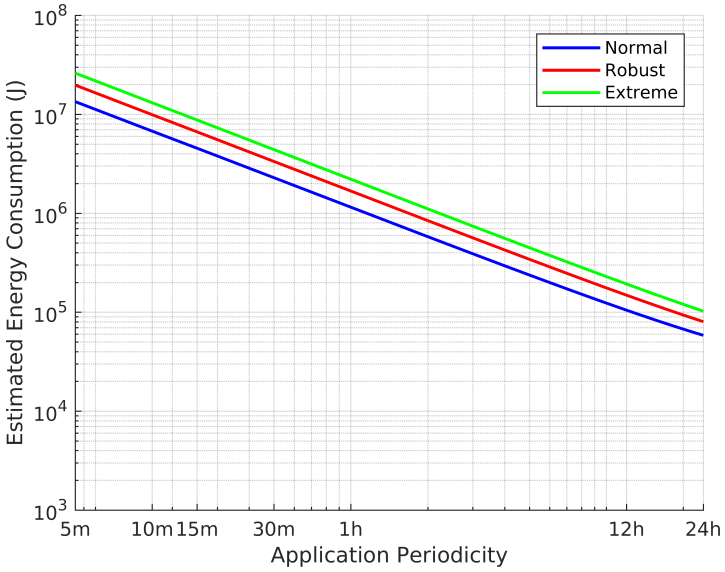
tach, using an encrypted IMSI (section 2.1.5), and we assumed that the security context was never deleted.

7.2.3 Results

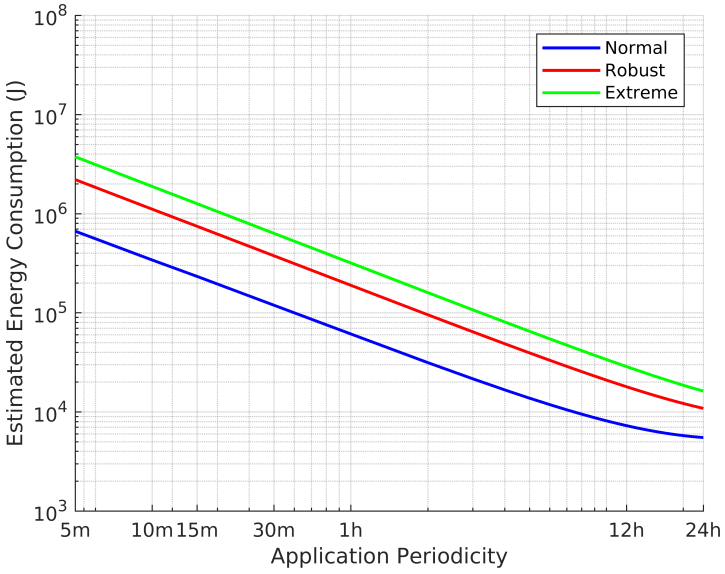
Figure 7.7 shows the estimated battery requirements for the three examined devices and the most common application periodicities [14, 16] for each coverage level. Our results also show that for 5-minute application periodicity, the minimum battery capacity needed is ≈$88 Wh, while a 30-min application periodicity would require a battery of ≈ 30 Wh capacity. If we consider the goal of 5 Wh capacity, then the application may have at most a 3-h application periodicity in order to achieve the 10-year life expectancy. Furthermore, these requirements exceed the overall cost goal of $5 per device set by 3GPP [48]. At the time of writing our research revealed that, the cheapest batteries of ≈ 90Wh and ≈ 30Wh capacity, cost ≈$45 and ≈$30 respectively.

Please note, that the above estimations do not include possible renewals of the security context that would require the generation of asymmetric keys for the IMSI encryption (section 2.1.5.1) in addition to the one performed during initial registration. This feature is currently optional for NB-IoT devices, as it can increase the energy consumption significantly depending on the deletion frequency. Including this would further increase the battery drainage, so our analysis should be considered as an upper limit of the battery life.

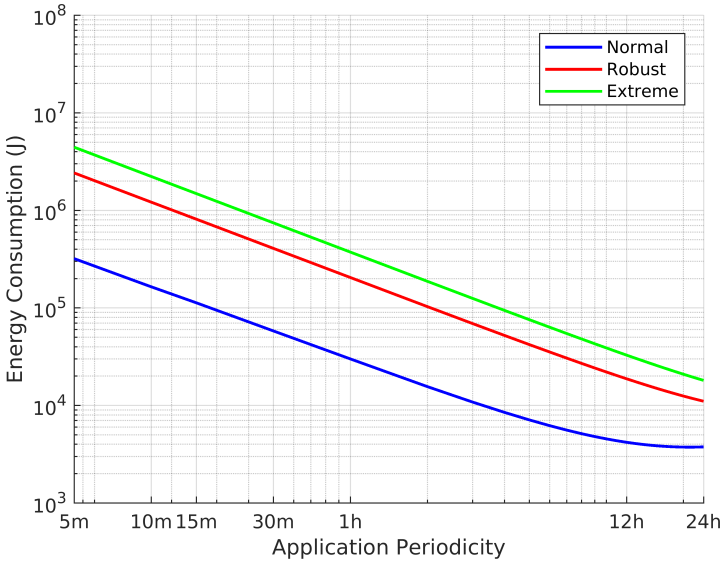
Overall, shorter application periodicities require frequent connections that consume more energy, as the devices need to frequently establish a connection (section 2.1.3) and get in the working state, which has a significant energy cost (section. 7.1). As the application periodicity increases, the devices spend increasingly more time in the more efficient deep sleep state (figure 7.8), which significantly lowers the overall energy consumption. During that time, the device mainly consumes energy to perform the TAU process at the end of each PSM cycle (section 2.3.4). As such it is imperative that this is taken into account in the energy model. In figure 7.9, we see that the model of [20] which does not include the TAU process, significantly underestimates the energy consumption of the device in the normal coverage level, which leads to a discrepancy of up to 44.5% for 24-h application periodicity. These results indicate that optimising the TAU process would have the greatest impact in the battery life of devices with long application periodicities. Conversely, for devices with shorter application periodicities, minimising the time spent in the working state and switching



(a) Pycom



(b) Quectel



(c) Sodaq

Figure 7.7: **Energy consumption for 10 years of operation for different application periodicities and coverage levels:** The figure depicts the average energy consumption for different application periodicities and different coverage levels.

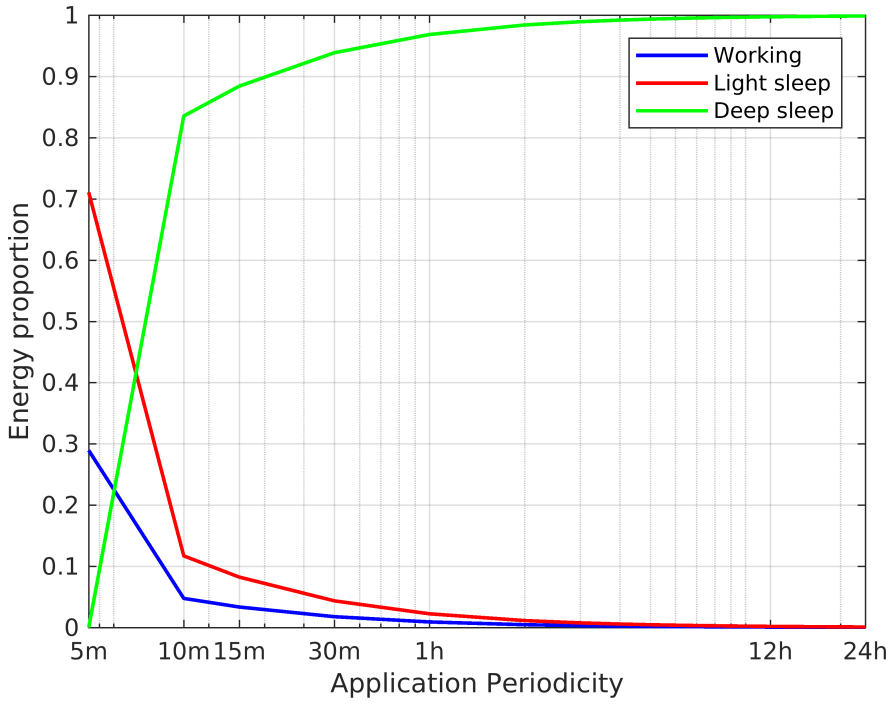


Figure 7.8: **Proportion of energy spent on each state, as a function of the application periodicity:** The figure depicts the proportion of energy spent in each of the three performance states in a single period P for different application periodicities.

faster to the more efficient light sleep state would bring a noticeable difference.

7.3 Energy Optimisation Mechanisms

Our device measurements (section 7.1) show that the most energy consuming operations in the working state are the active waiting, IMSI encryption and Attach/RA processes. Therefore, in this section we propose a set of novel mechanisms to optimise their energy consumption and achieve a substantial effect on the battery life expectancy. These mechanisms could also be applied to 4G/5G networks, but they are most crucial to energy constrained devices, such as NB-IoT. Furthermore, we present a guideline of best practices for device manufacturers and network operators, and quantitatively measure their impact to the overall energy consumption. For fair comparison among all proposals, we assume a baseline where no optimisation is applied, and the current 4G/5G procedures are followed. Please note that in the following evaluations we only consider the Sodaq device and normal coverage level for better clarity, however, the same trends can be observed for the other coverage levels.

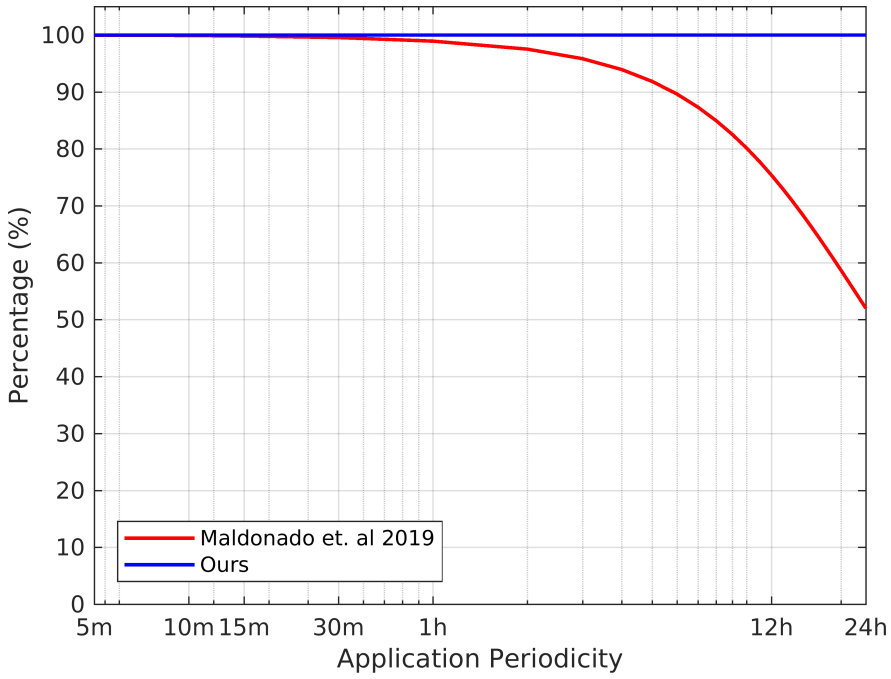


Figure 7.9: **Comparison of predicted energy consumption of our model and previous works:** The figure depicts the energy consumption predicted by [20] for different application periodicities relative to our model and [20], assuming normal coverage level and the use of CP optimisation. As the application periodicity increases, the energy for the TAU process at the end of the PSM cycle dominates total energy consumption, and as such it should be considered when estimating the battery life of NB-IoT devices.

7.3.1 Reduction of RA Connections and Security Context Renewal

As the RA process and IMSI encryption are two of the most energy consuming operations, reducing their frequency would bring significant benefit to the battery life expectancy of NB-IoT devices. Towards this end, we propose a mechanism that exploits the periodicity of IoT devices, to reduce the frequency of these operations, without limiting functionality nor breaking backwards compatibility for non-periodic devices. Since the majority of NB-IoT devices are expected to fall into that category [15], this would have a major impact in practice.

First, we propose reducing the number of RA processes by eliminating the TAU process at the end of a PSM cycle. The TAU process is mainly used to refresh the Tracking Area of the device, and to indicate availability to receive network originated data. While the first point is necessary in mobile, non-periodic devices as the network needs to be aware of their location, that is not the case for stationary and periodic devices, as there is communication with the network at fixed and predictable intervals. Therefore, by estimating their application periodicity, the network can assume that their Tracking Area remains unchanged as long as a transmission is not missed for more than n consecutive periods. To notify the device about network-originated data, we propose the use the paging procedure. As the PSM cycle is agreed between the device and the network, its exact ending time is known to the network in advance with millisecond accuracy, and can thus be used to page the device efficiently. Furthermore, paging the device is preferable in terms of energy consumption, as the RA process will need to be followed only when network-originated data really exist.

Second, we propose using the estimated application periodicities to decide when to delete the device's security context. Currently, a security context is deleted if it has not been used for a (relatively short) period of time. As such, devices with long application periodicities may be forced to perform IMSI encryption (if used) at frequent occasions or on every connection. A naive approach would be to simply increase the deletion threshold to a much larger value, but that is sub-optimal as the network would be forced to retain the security contexts of periodic devices with short application periodicities for longer than necessary. However, similar to the Tracking Area refresh, we can assume that a device is still operational as long as it does not miss more than n consecutive periods. A similar approach can also be adopted for non-periodic devices. While we cannot use estimated application periodicity in this case, these devices will perform the TAU process at the end of each PSM cycle. Therefore, we can use the

length of the PSM cycle, and only delete the security context if the device misses a TAU for more than n consecutive PSM cycles.

7.3.1.1 Mechanism

Our mechanism exploits the lack of mobility of NB-IoT devices, and reduces the number of costly and unnecessary procedures, thus decreasing their overall energy consumption. At the same time, it is able to accurately estimate when a security context needs to be deleted to prevent stalling contexts from being stored indefinitely, without increasing the energy consumption of the devices.

In our mechanism, devices need to inform the network whether they are periodic or not, but are not required to provide their actual application periodicity. At their initial connection devices register with the network using the existing RA and Attach processes without any modifications. During the Attach process, the network questions the device regarding its capabilities using the *Capability Enquiry* message. The device replies with the *Capabilities Enquiry Response* message, which includes a new field indicating whether the device is periodic or not. This prompts the network to estimate its application periodicity using either its future transmissions for periodic devices, or the agreed PSM cycle for non-periodic devices. The new field can either be included in one of the message's existing extensions to retain backwards compatibility, or can be added in its main body. On the device side, this can be accomplished with the use of AT commands which can be easily set by the application designer, or added with a firmware update. The remainder of the Attach process remains unchanged.

The network estimates the application periodicities as the running average of the last y connections. The value of y is defined by the network operator based on the storage capabilities and the average number of devices being served by the network. Specifically, the network estimates the application periodicity of the i_{th} device at the m_{th} connection as $s_i = \frac{1}{y} * (t_{m-y} + t_{m-y+1} + ... + t_m)$. Based on the estimated application periodicities the network can determine an approximate time of the device's next transmission/TAU, and adjust its security context deletion time as: $t_{deletion} = t_{now} + s_i * (n + offset)$ where n is the maximum number of missed periods allowed and is determined by the network, t_{now} is the current time, and $offset$ is the application periodicity percentage that can be tolerated as time offset. For this mechanism we consider that a value of $n = 0.5$ is adequate.

The aforementioned mechanism exploits the lack of mobility of NB-IoT devices and reduces the number of costly and unnecessary procedures, thus decreasing the

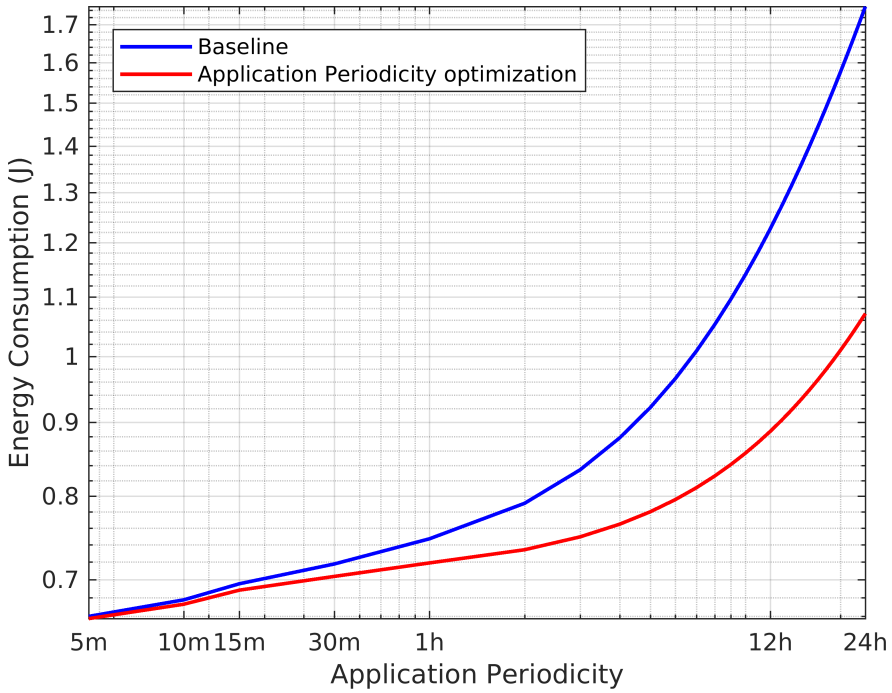


Figure 7.10: **Energy gains per period P for different application periodicities, with and without the periodicity estimation optimisation:** The figure depicts the energy gains per period P for the Sodaq device, when the current procedure is followed (no optimisation), and when our proposed periodicity estimation optimisation is applied.

energy consumption of the devices. At the same time, it is able to accurately estimate when a security context needs to be deleted to prevent stalling contexts being stored indefinitely, without increasing the energy consumption of the devices.

7.3.1.2 Results

Figure 7.10 shows the energy gains in Joules per period P (section 7.2.1) using the Sodaq device. We can see that the proposed optimisation significantly benefits devices with large application periodicites, resulting in energy gains of up to 37.8% per period P. As these devices spend most of their lives in the PSM state, the TAU process in these cases adds an unnecessary energy cost that can be avoided without impacting their operation.

7.3.2 Elimination of Random Access Process

The RA process is repeated every time the device connects to the network, and considering the significant energy cost of a single RA process, this accounts for a large part

of the overall battery consumption. The RA process serves two purposes: (i) receiving the TA information (section 2.1.3.2) in order to synchronise in the uplink, and (ii) receiving an uplink grant for the following transmissions.

For highly mobile devices, getting a new TA at each connection is necessary as it is likely to have changed from their last connection. As NB-IoT devices are mainly stationary, their TA does not significantly change between subsequent connections, and changes in their environment (e.g. a passing car) do not have a significant impact on the signal propagation delay, thus not affect their TA value. Therefore, we propose a mechanism that omits the RA process for stationary devices, and provisions an uplink grant for their next transmission. We do note however, that if uplink synchronisation is lost for any reason, it can still be detected and corrected using the existing TA update message. Further, in contrast to the two-step RA process currently discussed by 3GPP for Release 17 onward [231, 232] which aims to minimise the cost of an RA process, our mechanism can eliminate RA processes altogether (apart from the initial one), by exploiting the periodic nature of most IoT devices to pre-schedule appropriate grants in advance.

Our mechanism works as follows. On its very first connection, a stationary NB-IoT device performs the existing RA process without modifications to receive its corresponding TA value. During the Attach process that follows the initial RA, the device informs the BS whether it is stationary or not, using the existing Capability Response message. Similarly to our previous mechanism, this can be done either by introducing a new field in the existing message, or by using one of the existing extensions. After their application data exchange and at the end of their connection, stationary devices retain their TA value, to use it again the next time they wish to transmit data. The BS also retains the devices bearers (both DRBs and SRBs). Although, the TA is not expected to significantly change if devices are stationary, the BS can employ the existing TA adaptation procedures to modify the devices TA if it deems it necessary, and the device stores the latest TA value indicated by the BS.

In order to completely eliminate the RA process, stationary devices that re-use their TA value, need to also have an uplink grant for their next transmission. The BS can schedule the device's next transmission based on its periodicity. One possibility is that the device can directly inform the BS about its periodicity using the Capability Response message. Based on that information, the BS can calculate the correct SFN and HFN (section 2.1.2.1) when the device will wake up and pre-schedule sufficient resources then. If the device is unable to provide a periodicity value, our previously

described periodicity estimation mechanism can be used, adding a small number of frames to cover clock drifts at the device side. In either case, the scheduling of the resources is straight-forward, since the elimination of the RA process and the re-use of existing bearers, means that the BS does not need to consider extra delays that should be taken into account for an accurate scheduling. In case the device misses its scheduled resources, the existing RA process can be followed to request new resources. However, as the bearers exist, the Attach process can still be skipped.

An important feature of the proposed mechanism is that although the devices do not follow the RA and Attach processes, the data can still be securely exchanged. Specifically, as the scheduling of resources is device-specific, the BS station is aware of which device transmitted in what resources. This means that the device can reuse its previously established security context to encrypt its data, and the network will still be able to decrypt it. If required, the network can request the renewal of the security context, which can be applied either to the current transmission (i.e. the first transmission is rejected and the device needs to repeat it after the renewal of the security context), or to the devices next transmission (i.e. the security context is renewed after the data transmission, and is applied on the device's next transmission).

To assess the performance of this optimisation we examine the 10-year battery life using the energy consumption results of the Sodaq device (figure 7.11). We can see that eliminating the RA process further reduces the energy consumption regardless of the periodicity of the device, compared to the current procedure where no optimisation is applied. The greatest gains are observed when short application periodicities are used, due to the increased number of connections they have to do throughout their lives, with a maximum gain of 13.6%. Smaller, but still important energy gains are also observed for longer periodicities, with a 10.3% gain for a device with a 24-hour periodicity.

7.3.3 Best Practices for Energy Reduction

7.3.3.1 Inactivity Timer & Active Waiting

Active waiting requires the second greatest amount of energy after the IMSI encryption. This feature was initially introduced to reduce the number of connection establishments when devices transmitted data in close intervals. As such, it may not be required in NB-IoT where applications are expected to transmit all of their data in one go. Although using the C-DRX can decrease the energy consumption, forcing the devices to remain connected to the network for a period of time after their transmission

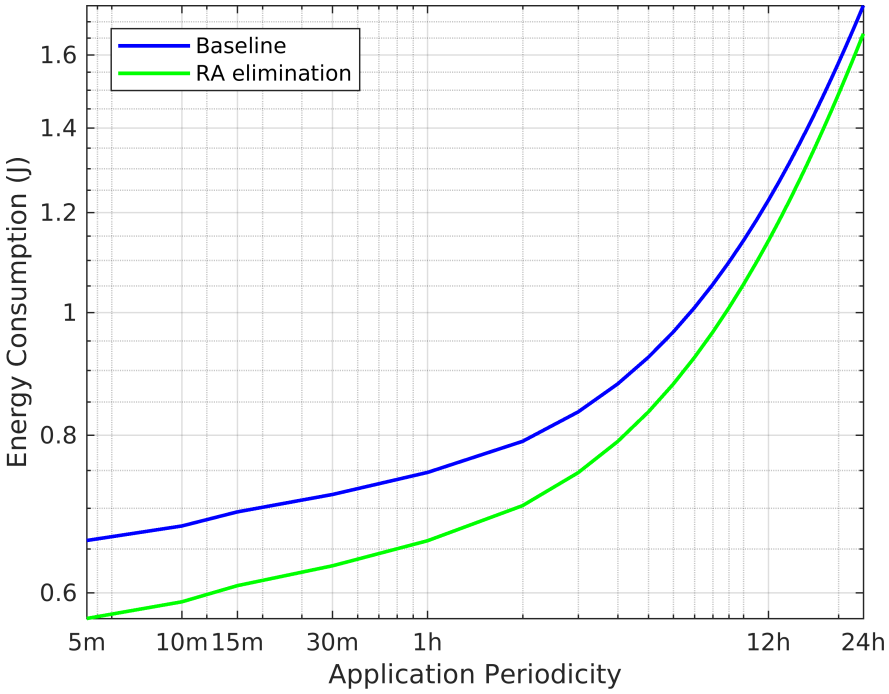


Figure 7.11: **Estimation of 10-year energy consumption for different application periodicities with elimination of the RA process:** The figure depicts an estimation of the total energy consumption in Joules for 10 years of operation for the Sodaq device when the current procedure is followed and when devices retain their TA value and skip the RA process.

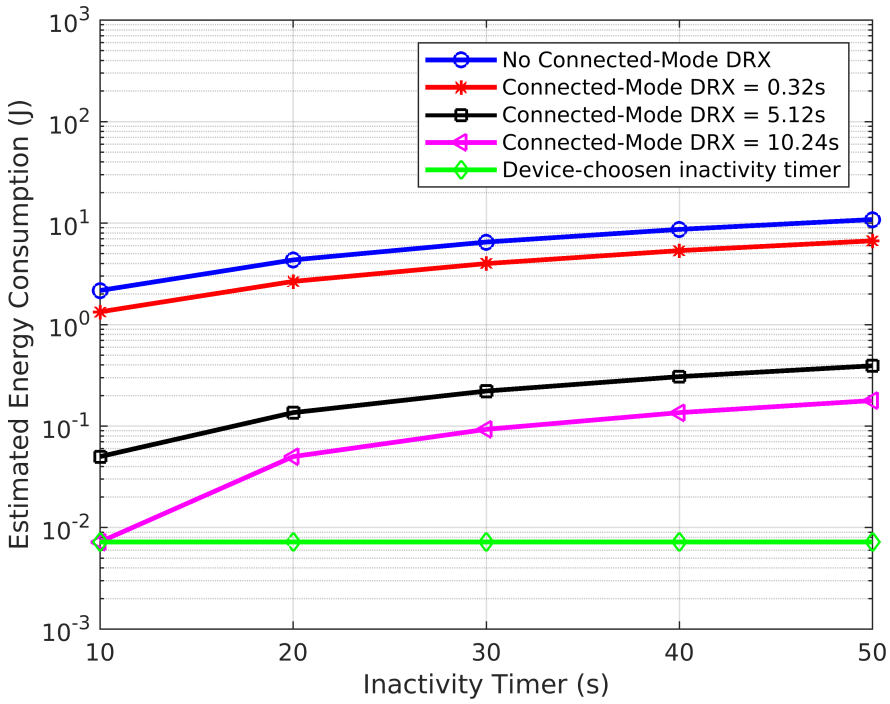


Figure 7.12: **Estimated energy consumption with different C-DRX cycles and self-selected inactivity timer:** The figure depicts the estimated energy consumption when different C-DRX cycles are used and when devices choose their own inactivity timer compared to the inactivity timer values used in commercial networks.

unnecessarily wastes energy. Our results showed that even the use of the shortest C-DRX cycle (0.32 seconds) can decrease the energy consumption by 22% and 38.2% for a 10-sec and 50-sec inactivity timer respectively. Further gains can be achieved as the C-DRX cycle increases.

3GPP has recognised the benefits of releasing a connection early, and introduced the *Release Assistance Indicator (RAI)* feature [28] as part of a series of optimisations in Releases 14 and 15. The RAI is a new field included in control messages to indicate that the device wishes to terminate its connection, so that the connection release procedure can be triggered immediately, reaching an energy gain of 98%.

It is important to note that this novel feature is not supported by older NB-IoT devices, and in fact our experiments showed that two of the tested devices did not include it. Although this feature can be added with a software update, existing applications also need to be updated in order to use it. Furthermore, there is currently no support to allow the application provider to inform the network whether outstanding data exist [188], to avoid paging the device shortly after it released a connection. However, the energy benefits are significant, and we believe that it is important that this feature is enabled

in both older and newer NB-IoT devices.

7.3.3.2 Attach process

Similarly, the Attach process is costly, and needs to be repeated at each network connection, incurring a significant impact on the battery life of the devices, especially for devices with short application periodicities. Therefore, 3GPP defines the *Control Plane (CP)* and *User Plane (UP)* optimisations [51] (section 2.3.3.3) to help reduce the energy consumption of NB-IoT devices. Although these optimisations have some drawbacks, we believe that they can contribute to the reduction of the energy consumption, and therefore, we estimate their energy gains if they are implemented (figure 7.13). Our experiments show that both approaches can decrease the energy consumption, especially for devices with short application periodicities. Greater energy gains can be achieved when the CP optimisation is used, provided that the data size and QoS requirements can be met by the SRBs. Specifically, the UP optimisation decreases the energy consumption by 2.9% while the CP optimisation results in an energy gain of 5.2% when a 5-min application periodicity is used. For a 24-h application periodicity, the energy gain drops to 1.2% and 2.2% for the UP and CP optimisations, respectively.

7.3.3.3 Security Deletion Frequency

Although our proposed periodicity optimisation mechanism simplifies the decision of when to delete a security context to prevent stale contexts from being stored indefinitely and without incurring extra energy consumption, it is a new proposal waiting to be deployed, and operators must make a decision on when to delete a security context. Therefore, here we provide an evaluation of the energy consumption with 1-h, 12-h and 24-h application periodicities, and a security context deletion frequency ranging from 1 to 24 hours (figure 7.14). Please note that the energy consumption include all operations within a period P. Due to the significant differences on the consumption, figure 7.14 does not depict the details for each application periodicity, therefore figures 7.15, 7.16 and 7.17 depict the energy consumption for the 1-h, 12-h and 24-h application periodicity in isolation.

We can observe a significant difference between the 1-h application periodicity and the other two, which is explained by the fact that for the 12-h and 24-h application periodicity the device spends the majority of its life in PSM. However, a device with

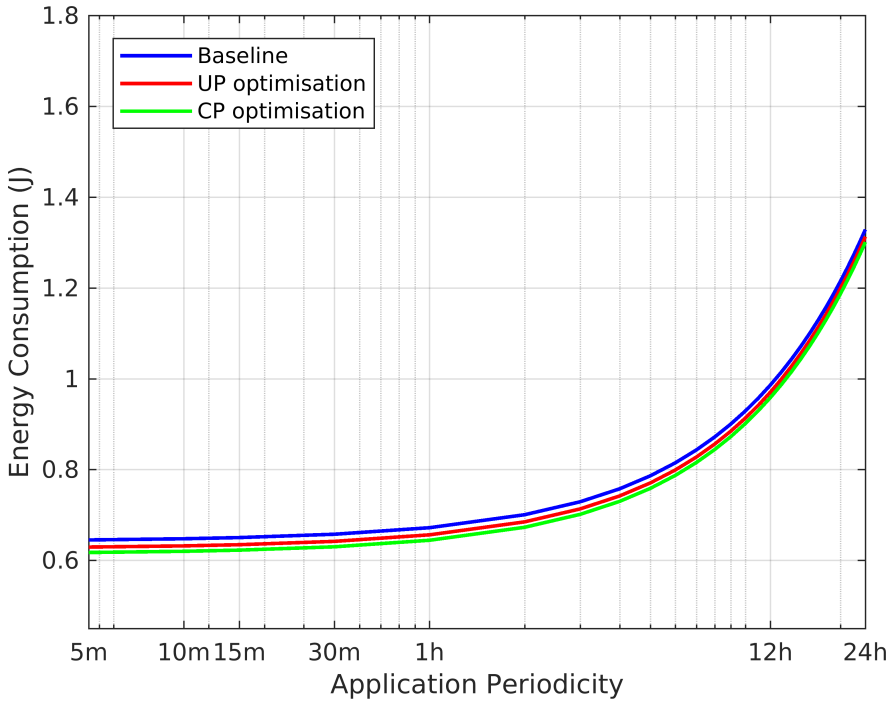


Figure 7.13: **Energy consumption per period P with and without optimisations during the Attach process for different application periodicities:** The figure depicts the energy consumption during a period P with the UP and CP optimisations, compared to the baseline.

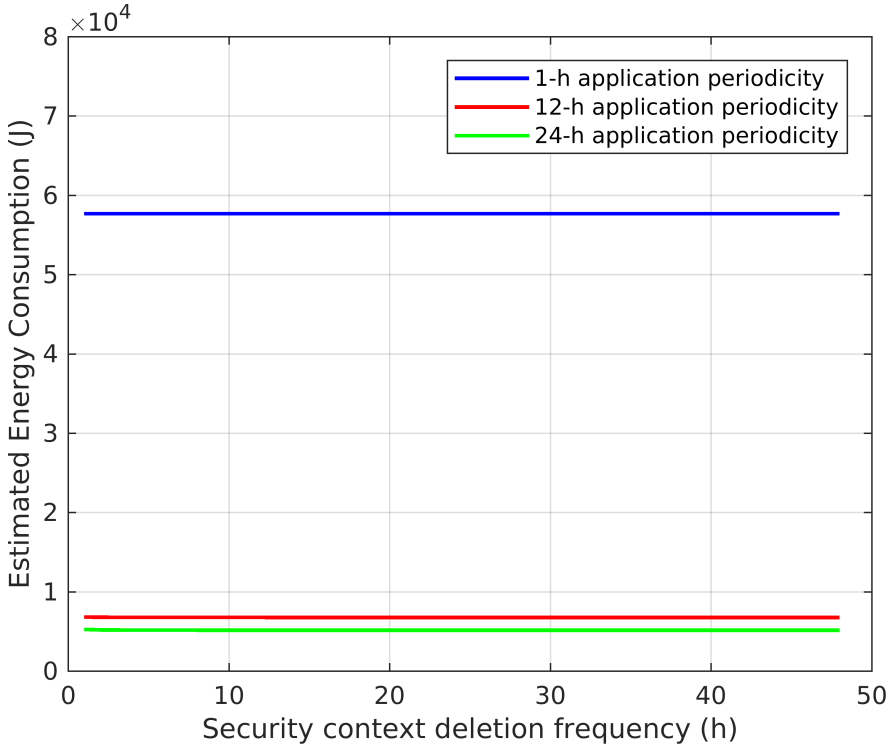


Figure 7.14: **Energy consumption based on security context deletion frequency:** The figure depicts the energy consumption introduced for 3 different application periodicities when the security context deletion frequency ranges from 1 to 48 hours.

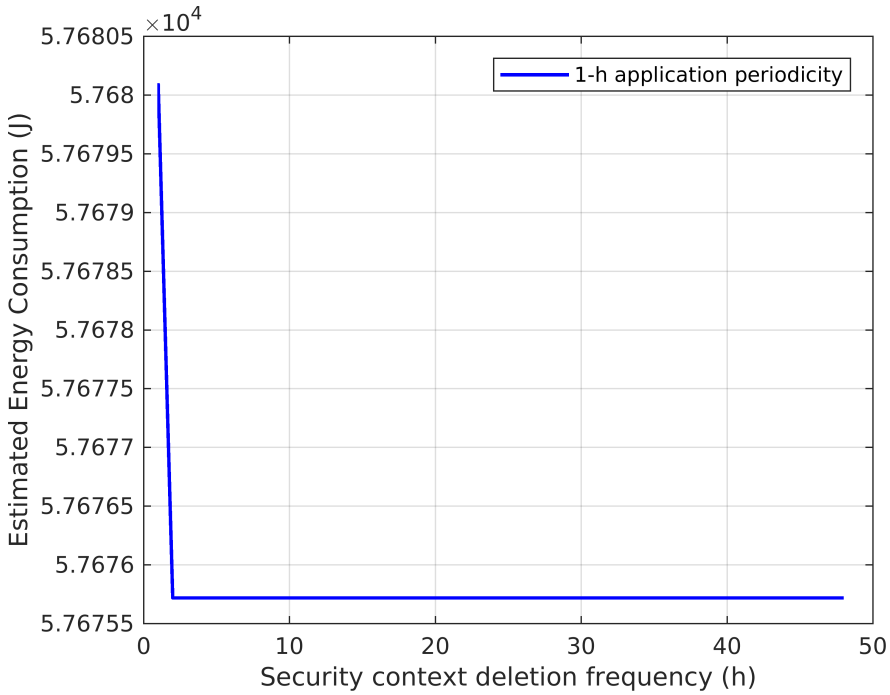


Figure 7.15: **Energy consumption of 1-h application periodicity and different security context deletion frequencies:** The figure depicts the energy consumption of the device with 1-h periodicity when the security context deletion frequency ranges from 1 to 48 hours.

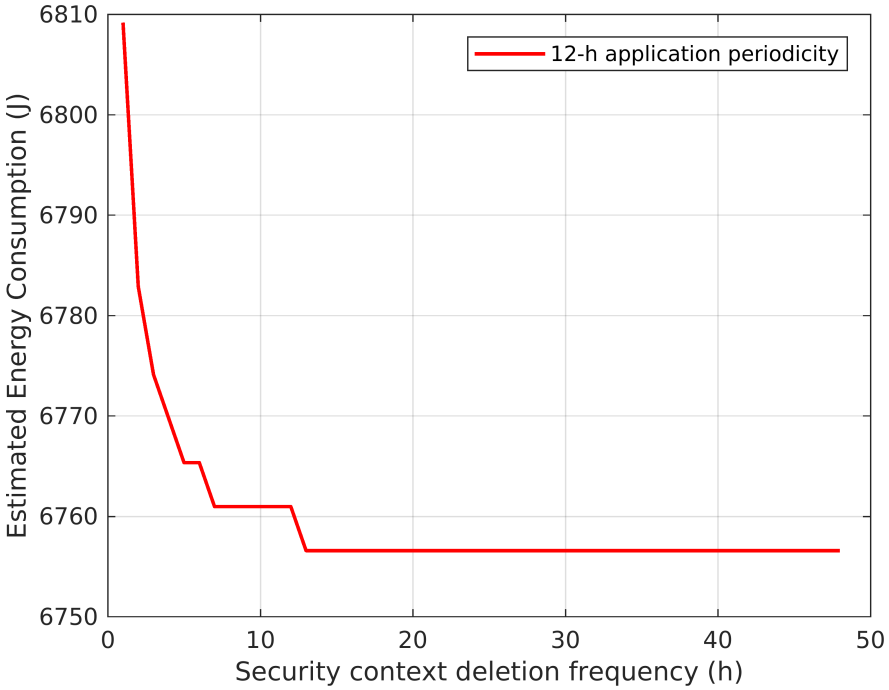


Figure 7.16: **Energy consumption of 12-h application periodicity and different security context deletion frequencies:** The figure depicts the energy consumption of the device with 12-h periodicity when the security context deletion frequency ranges from 1 to 48 hours.

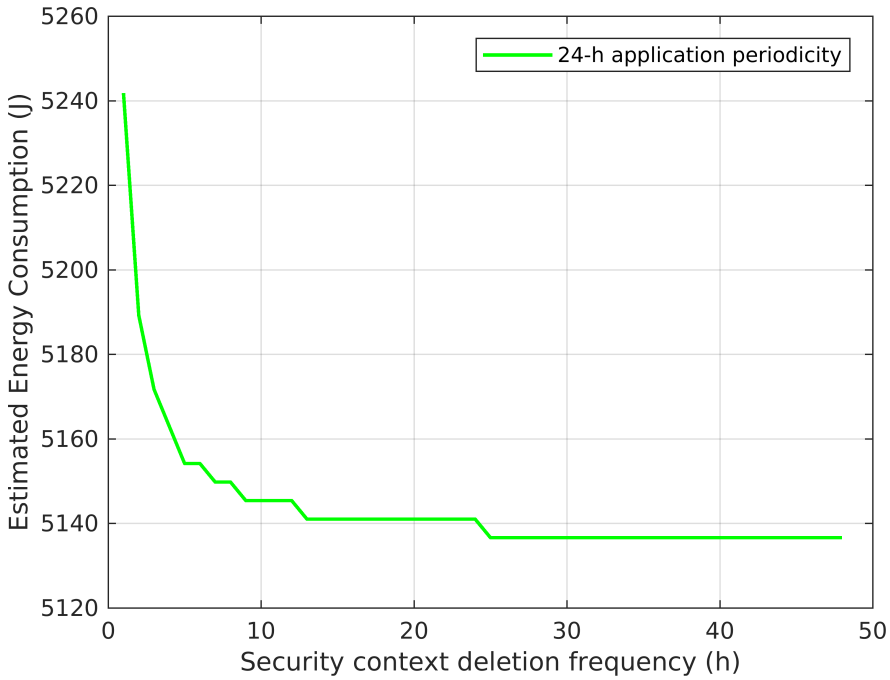


Figure 7.17: **Energy consumption of 24-h application periodicity and different security context deletion frequencies:** The figure depicts the energy consumption of the device with 24-h periodicity when the security context deletion frequency ranges from 1 to 48 hours.

1-h application periodicity will seldom be required to renew its security context, as usually the security context deletion times in commercial networks are larger than 1 hour. This is not the case for the 12-h and 24-h application periodicities that will be required to renew their security contexts more frequently.

7.3.4 Ablation Study

Finally, we compare the energy consumption per period P for different application periodicities, using each of the aforementioned optimisations (either proposed by us, or by 3GPP), and their combinations (figure 7.18). In this experiment we used a 10-second inactivity timer, with 5.12-second C-DRX cycles. Figure 7.18 shows the estimated energy consumption in Joules per period P for the different application periodicities. As the application periodicity decreases, the energy consumption of the RA and Attach processes becomes increasingly significant since these operations need to be repeated often. Therefore, for short application periodicities, the 3GPP proposed UP/CP optimisations and our RA elimination optimisation provide the greatest gain. As the application periodicity increases, however, the energy consumption of the device starts being dominated by the deep sleep state, which significantly reduces the effectiveness of these optimisations. Instead, our application periodicity optimisation that targets the PSM/security renewal provides the greatest reduction in energy consumption (up to 46.14%). As these various optimisations are mutually complementary, they can be combined to yield the greatest gains across the entire range of application periodicities.

7.4 Conclusions

In this chapter, we presented a detailed measurement-based characterisation of the energy consumption in real NB-IoT devices, and identified expensive network operations that can be the target for optimisations. Further, we presented an energy consumption model which we used in combination with our measurements in a simulation study to estimate the battery requirements of NB-IoT devices to achieve a battery life of 10 years. Our analysis showed that the current 3GPP specifications for NB-IoT, that are largely inherited from 4G and 5G, are not efficient in NB-IoT if used unchanged. We then proposed two optimisation mechanisms that exploit the traffic characteristics of NB-IoT devices to reduce the energy cost of unnecessary operations, thereby substantially decreasing the overall energy consumption.

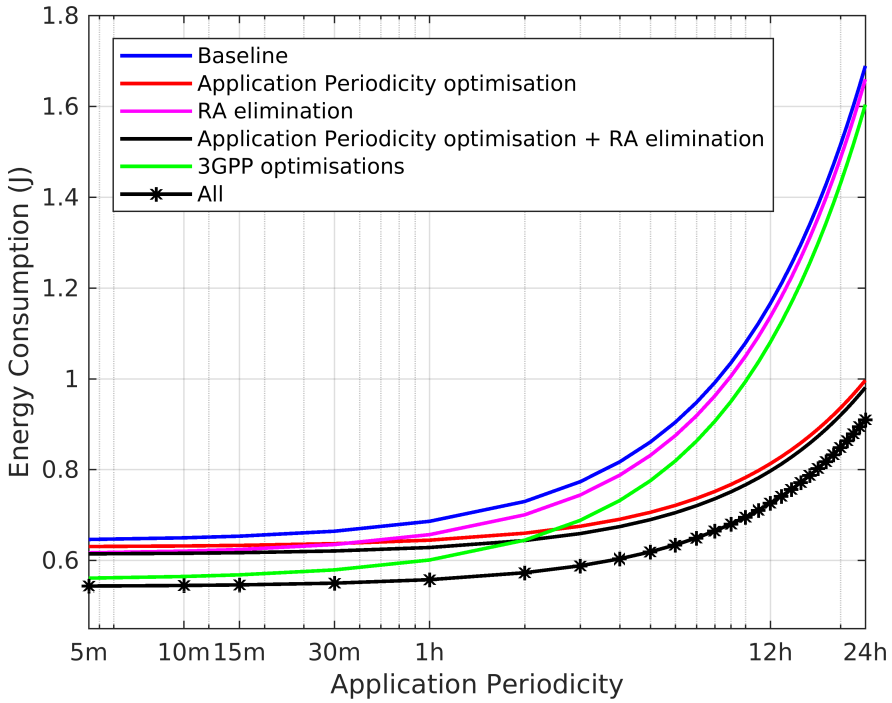


Figure 7.18: **Ablation study of the estimated energy consumption for different energy optimisations and different application periodicities:** The figure depicts an ablation study of the estimated energy consumption per period p of different optimisations and different application periodicities.

Chapter 8

Conclusions & Future Work

8.1 Conclusions

Internet of Things (IoT) has revolutionised multiple domains of our lives by extending network connectivity to everyday objects, allowing them to communicate with each other without human intervention. There are currently a wide range of applications that depend on IoT devices (e.g. smart homes, industry automation, autonomous driving), and new applications are being developed every day. Due to the freedom that IoT has brought, the number of IoT devices deployed has seen a tremendous growth in the recent years, and their numbers are only expected to grow in the near future, with forecasts reporting that ≈ 29 billion IoT devices will be operational by 2022 [1].

Cellular networks are considered a great candidate to support them, mainly due to their extensive deployments, increased area coverage, and flexible data rates. However, traditional cellular networks were historically dominated by HTC devices, that drove their design over the years. HTC devices present substantially different traffic patterns compared to IoT, that causes a series of inefficiencies, both at the network, as well as the device side, making existing cellular networks struggle as the number of IoT devices increases.

These problems are not specific on a singular area, but span a number of different operational areas of cellular networks, such as the connection establishment process, the network resource utilisation and the device energy consumption. In this thesis we focused on three key problematic areas for IoT devices in cellular networks, and provided solutions towards their efficient support, focusing on their unique communication patterns and requirements. Although each of these solutions focused on different areas, they can complement each other in providing a holistic framework for the ef-

ficient support of IoT devices in the current and future cellular network technologies. The high-level conclusions for each of these solutions is presented in the following subsections.

8.1.1 Connection Establishment

For our first contribution, we focused on the connection establishment process and the three major reasons that cause delays and network outages when large numbers of IoT devices try to establish a connection: (i) the increased number of collisions in the RACH when multiple devices attempt to access the network simultaneously, (ii) the increased signalling overhead in the RAN, and (iii) the increased signalling overhead in the core network. To address these issues we presented ASPIS, that exploits the periodic and predictable nature of the majority of IoT devices to control the RACH congestion, and reduce the signalling overhead for establishing a connection both in the RAN and the EPC parts of a cellular network. On top of its basic design, ASPIS features an improved RA process with fewer messages that also provisions for short packet transmissions (< 80 bytes) with even less signalling. Finally, ASPIS features a proactive preamble split scheme that is able to alleviate collisions while minimising the impact of IoT traffic on existing HTC.

A major advantage of ASPIS is its ease of implementation that does not require hardware changes, allowing to be incrementally deployed alongside legacy devices and infrastructure, while at the same time maintains security. We showed the practicality of ASPIS by implementing it on the OpenAirInterface platform, and demonstrated its effectiveness through extensive evaluations via a combination of small-scale experimental evaluation and large-scale, realistic simulations. Our experimental and large-scale simulation results confirmed that ASPIS outperformed the standard LTE, as well as other proposals at the time of publication.

Further, and focusing specifically on the RACH congestion, we presented a probabilistic preamble selection mechanism that based on the recent advances of the NORA approach. Our mechanism employs the NOMA approach to separate colliding preambles and split the coverage area of a cell in different logical zones. By splitting the cell coverage area, devices in different zones are able to use the same preambles without colliding, as preamble collisions can be resolved at the receiver side, effectively increasing the total number of preambles available in the cell. Then, we used a probabilistic selection scheme with Reinforcement Learning to decrease the number of colli-

sions of devices within the same zone. This scheme constantly observes the preamble usage in order to create zone-specific statistics about the preamble usage, which are then used by the devices to select the resources which are less likely to result in a collision.

The use of Reinforcement Learning allows the network to capture dynamic changes in the cell and update the preamble usage accordingly. Overall, our mechanism is able to decrease the number of collisions in the whole cell area, thus also decreasing the network access delay and allowing larger number of devices to connect to the network with fewer connection attempts compared to the currently used procedure. Most importantly, and similarly to ASPIS, our mechanism does not require hardware changes allowing it to be implemented as a software update, without breaking compatibility with existing devices.

8.1.2 Network Resource Utilisation

For our second contribution, we focused on the network resource utilisation and energy consumption of IoT devices during group communications. We argued that the currently used framework for group-communications is ill-suited for IoT devices and results in significant resource wastage in the network side, especially in resource limited technologies, such as NB-IoT.

To address these inefficiencies, we discussed the architectural and procedural enhancements needed to support the unique features of group communications in NB-IoT, and proposed a set of enhancements to the existing multicast framework. Our enhancements consist of fewer and shorter procedures, which are tailored to the demands of IoT devices, thus greatly simplifying the currently used schemes. To address both network resource utilisation and device energy consumption, we presented a novel on-demand model that forgoes the subscription-based model currently used in cellular networks, that requires periodic service announcements and monitoring. We also extended the NB-IoT frame to include channels for multicast transmissions, and proposed two transmission strategies for multicast content delivery. We evaluate their performance through extensive simulations, considering the impact of multicast transmissions on the downlink background traffic and the channel occupancy. Finally, to further improve the resource utilisation at the network side and the energy consumption at the device side, we presented three different mechanisms to achieve device synchronisation and grouping, with different trade-offs between network resource usage,

energy consumption and compliance with the NB-IoT standards.

8.1.3 Device Energy Efficiency

For our third contribution, we focused on the device energy efficiency to identify the operations that contribute the most to the overall energy consumption, and discuss optimisations. Towards this end, we started with a thorough experimental measurement of the power consumption of the individual operations under normal use, using three different commercial NB-IoT devices. Building on the our experimental energy measurements, we presented an NB-IoT energy consumption model, which we then used to simulate the battery life of a device under realistic traffic conditions. This gives us an estimate of the battery capacity requirements to meet the 10 year goal that has been set by 3GPP for IoT devices.

A major finding is that 3GPP significantly underestimates the energy consumption of the different operations, and its battery life goal is far from realisable with the currently used protocols and other previously proposed approaches, especially if we also consider the $5 cost constraint. As several works are based on the energy assumptions of 3GPP, we show that they also underestimate the overall battery usage, and do not take into account operations that can significantly contribute to the energy consumption. Having unearthed the operations with the higher energy usage, we proposed novel mechanisms that exploit the distinct characteristics of NB-IoT devices, to substantially reduce their energy consumption. Our results show that our optimisations can greatly reduce the energy consumption of IoT devices, especially when the application periodicities are long and devices spend most of their lives in sleep states. Finally, we discussed a set of best practices under the existing protocols, that device vendors and network operators should consider in order to maximise battery life.

8.2 Limitations & Future Work

This section summarises the limitations of the contributions presented in this thesis, and presents directions for future work.

8.2.1 Connection Establishment

Connection establishment is probably the most important step in the communication process with the cellular network, and can be the major bottleneck in high-load condi-

tions, due to the limited number of available preambles available for the RA process.

Support for event-driven IoT devices: In this thesis we proposed ASPIS that exploits the periodic and predictable nature of the majority of IoT devices in order to decrease the number of collisions in the RACH, thus allowing larger number of devices to connect to the network with fewer attempts. However, not all IoT devices are characterised by these traffic patterns patterns. As it is difficult to predict future transmissions of event-driven devices, our ASPIS mechanism reverts back to the existing scheme and uses historic data to adapt the preamble split, which can still lead to congestion in the RACH. To efficiently support event-driven IoT devices, artificial intelligence methods can be used in order to train the network to identify possible events early on, and provide accurate warnings about their locality and severity. Based on those predictions, the most appropriate action can then be taken, such as to adapt the preamble split or temporarily operate back-up cells in affected areas.

Limited mobility: In chapter 7 we discussed the idea of a-priori knowledge of the TA value in order to omit the RA process. Although this idea can have minimal impact on stationary devices, it cannot be extended to mobile IoT devices (e.g. fleet tracking), even when the devices do not move outside the coverage area of the cell. To suppose such limited mobility for IoT devices, we can extend the idea of logical zones presented in chapter 5 to the TA value. As the TA value is closely correlated with the propagation delay, different logical zones within the coverage area of a cell can have different TA values. To assist devices in determining their TA zone, measurement transmissions from the BS can be employed that will allow the device to compute its propagation delay, similarly to the transmission measurements already used for the calculation of the received signal strength.

Bearer establishment: Our ASPIS mechanism simplifies the connection establishment process by retaining the core network bearers of the device, to be immediately used in future transmissions, based on the idea that the communication parameters of an IoT device do not significantly change among subsequent transmissions. However, if these parameters do change a new bearer must be established in the core network to accommodate them. Furthermore, ASPIS still tears down any RAN bearers, which need to be re-established on each connection. Although ASPIS significantly decreases the overall signalling load, the repetition of bearer establishment in the RAN can further be optimised. To address these limitations, generic pre-established bearers can be employed both in the radio access and the core network, and devices can be assigned to a suitable radio access and core network bearer at the first steps of the RA process,

thus significantly decreasing the signalling overhead in all cases. However, as different device have different communication requirements, bearers with different parameter combinations need to exist throughout the network, to ensure that all requirements can be met. Furthermore, as large number of IoT devices can have similar communication requirements and may want to transmit data at the same time, multiple similar bearers need to be available to account for the upper limit of devices that can be supported in the same bearer. Therefore, the different parameter combinations for the bearers, as well as the number of bearers for each combination needs to be carefully investigated so that all devices can be accommodated efficiently, while at the same time avoiding stalling bearers from wasting resources.

Preamble splitting in zones: In our probabilistic preamble split work, we have assumed that preambles are evenly split among the different logical zones, as we assumed a uniform distribution of devices within the cell coverage area. However, in real deployments the device distribution may not be uniform, and the device density per zone may differ significantly among zones and among different times of the day. To account for the device density in each zone and on different times, a spatio-temporal model must be used to capture the dynamic changes in the cell in order to split the preambles between the different zones accordingly.

8.2.2 Network Resource Utilisation

Group formation: Our group formation scheme in chapter 6 assumes the presence of a coordination entity that is responsible for deciding what content should be delivered to which devices. Therefore, a new framework is required that will provide functionality to allow device vendors and network operators to (i) decide what content should be delivered, (ii) identify the devices that should receive the selected context, (iii) allow the network to retrieve the selected content from its source (e.g. vendor websites/databases) and (iv) deliver the content to the selected devices. Such a framework requires close collaboration between the network operators, application provides, device manufacturers and device owners. The new service-based architecture of 5G networks can serve as a starting point, but further research is required to design an efficient content-delivery framework that offers the aforementioned functionality.

Support for multicasting for devices in deep sleep: Our group communication and grouping mechanisms assume that devices are using the idle or inactive states and have a frequent or semi-frequent DRX cycle, and the multicast content delivery can

be planned and scheduled around those cycles. However, as we showed in chapter 7 IoT devices gain significant energy benefits by using the deep sleep state as much as possible. The drawbacks of this are that (i) devices are not reachable by the network while in PSM and (ii) the PSM cycles can be significantly longer than the DRX cycles, requiring the network to store the content for long periods of time. These drawbacks decrease the efficiency of the system as they require multiple transmissions of the same content, and delay the content delivery, especially for event-driven devices. Depending on the type of the multicast content, timely delivery of it may be important (e.g. security updates in the firmware). Therefore, new mechanisms are required to consider devices in deep sleep states and allow the network to contact them and deliver the multicast content in a timely manner, without increasing the resource usage.

Multicast bearers: As with any communication in cellular networks, multicast data needs to be transmitted using bearers, both within the core network and the RAN. Each bearer has different characteristics regarding the data rate and the QoS. As different devices have different capabilities, careful management of the multicast bearers is a major point that was not included in this work, but we strongly believe will improve the overall performance of group communications in NB-IoT. Efficient bearer grouping at the time of the random access and attach process can have significant benefits on the usage of RAN and core network resources and reduce the impact of multicast traffic on background traffic in the downlink (i.e., fewer bearers for multicast traffic thus more resources for background traffic). However, grouping dissimilar devices together in terms of QoS and transmission parameters may result in inefficient resource utilisation (i.e. inefficient transmission parameters even if only device does not have a good channel) and increased energy consumption.

8.2.3 Device Energy Efficiency

Assessment of energy consumption related to applications: In this work we have assessed the energy consumption of all the network-related operations, assuming a basic application of measurement reading running on the devices. As such, we have established a lower limit on the battery requirements. However, IoT devices are expected to run a large variety of applications extending beyond simple measurement readings, that might require data processing on the device side before transmission. Therefore, assessing the energy contribution of different applications is important in order to get more accurate estimations of the battery requirements for such devices.

Additionally, different applications have different requirements in terms of data rates, MCS, etc. In this work we have assumed a basic configuration for our considered application. However, both the data rate and the MCS that can result in different energy consumption estimations, and as such it is important to assess the energy consumption estimations under different transmission configurations in conjunction with the different applications running on the device.

Support for event-driven devices: In our energy performance work, we proposed the elimination of the RA process for periodic and stationary devices in order to reduce the overall energy consumption. However, this idea is not directly extendable to event-driven devices. Although the acquisition of the TA value for such devices can be implemented similarly to section 8.2.1 if devices are mobile, or can be retained from a previous connection for stationary devices, resource pre-scheduling is not possible, as there is no indication of the timing of the next transmission. In such cases, scheduling requests still need to be transmitted in a random access manner. However, as the TA value can be known before transmission of the scheduling request, there is no need to use the low-modulation and low-interference random access channel. Instead, the RACH can be replaced by the PUCCH/PUSCH, and be used for transmitting scheduling requests in a random access manner. Although collisions are still possible in such a scheme, the use of the PUCCH/PUSCH that has the ability to support higher MCS can result in increased resource availability for scheduling requests, and allow for more than 64 devices to access the network simultaneously, thus improving the overall system performance. As the capacity of the PUCCH/PUSCH is not unlimited, the feasibility of such an approach, as well as the quantitative gains need to be investigated.

Small cell deployments: In this thesis we have assumed typical macro cell deployments that provide coverage for a medium to large sized city, serving up to 55000 devices. Recent approaches however, move towards small cells that only cover parts of the city, or areas as small as a single building block. Although the different procedures and protocols do not change with the use of small cells, other factors such as the interference or the required transmit power can affect the functionality of such systems when large numbers of devices are deployed. Therefore, such scenarios are important to assess to determine whether they would be beneficial for IoT devices, the possible energy trade-offs be, and propose mechanisms to address them.

Bibliography

[1] Ericsson, "Internet of Things forecast," Tech. Rep., accessed: 2018-10-25. [Online]. Available: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast

[2] ——, "Population coverage," Ericsson, Review, 2018. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2018/population-coverage

[3] "Wi-Fi Alliance," https://www.wi-fi.org/, accessed: 2018-11-5.

[4] "BLuetooth Low Energy," https://www.bluetooth.com/news/pressreleases/2008/07/22/new-bluetoothstandard-built-for-the-universal-remote-control, accessed: 2018-11-5.

[5] "ZigBee Alliance," https://www.zigbee.org/, accessed: 2018-11-5.

[6] Qualcomm, "Leading the LTE IoT evolution to connect the massive Internet of Things," Qualcomm Technologies Inc., White paper, Jul 2017.

[7] S. K. Sandra, M. Dohler, and P. D. Prokar, "The Internet of Skills: use of fifthgeneration telecommunications, haptics and artificial intelligence in robotic surgery," *BJU INTERNATIONAL*, vol. 122, pp. 356–358, Sep 2018.

[8] K. Antonakoglou, X. Xu, E. Steinbach, T. Mahmoodi, and M. Dohler, "Toward haptic communications over the 5g tactile internet," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3034–3059, Jun 2018.

[9] S. Lien, K. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66–74, Apr 2011.

[10] M. Gerasimenko, V. Petrov, O. Galinina, S. Andreev, and Y. Koucheryavy, "Energy and delay analysis of LTE-Advanced RACH performance under MTC overload," in *2012 IEEE Globecom Workshops*, Dec 2012, pp. 1632–1637.

[11] M. Cheng, G. Lin, H. Wei, and A. C. Hsu, "Overload control for Machine-Type-Communications in LTE-Advanced system," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 38–45, Jun 2012.

[12] Ericsson, "Mobile subscriptions worldwide," Ericsson, Review, 2018. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2018/mobile-subscriptions-worldwide-q1-2018

[13] J. Schlienz and D. Raddino, "3GPP Low Power Wide Area Technologies," GSMA, Tech. Rep., 2017, available at: (https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf.

[14] Ericsson, "Cellular networks for massive IoT: Enabling low power wide area applications white paper," Stocholm, Tech. Rep., Mar 2016, accessed: 2018-11-5. [Online]. Available: https://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf

[15] J. Jermyn, R. P. Jover, I. Murynets, M. Istomin, and S. Stolfo, "Scalability of Machine to Machine systems and the Internet of Things on LTE mobile networks," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun 2015, pp. 1–9.

[16] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Large-Scale Measurement and Characterization of Cellular Machine-to-Machine Traffic," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1960–1973, Dec 2013.

[17] Cisco, "White paper: Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021," Tech. Rep., Mar 2017, accessed: 2018-10-25. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[18] J. Schlienz and D. Raddino, "Narrowband Internet of Things White Paper," Rohde & Schwarz, Tech. Rep., 2016, available at: (www.rohde-schwarz.com/appnote/1MA266.

[19] 3GPP, "NB-LTE - Battery lifetime evaluation," 3rd Generation Partnership Project (3GPP), RP 151393, Sep 2015.

[20] P. Andres-Maldonado, M. Lauridsen, P. Ameigeiras, and J. M. Lopez-Soler, "Analytical modeling and experimental validation of nb-iot device energy consumption," *IEEE Internet of Things Journal*, pp. 1–1, Mar 2019.

[21] M. Lauridsen, R. Krigslund, M. Rohr, and G. Madueno, "An Empirical NB-IoT Power Consumption Model for Battery Lifetime Estimation," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, Jun 2018.

[22] L. Feltrin, G. Tsoukaneri, M. Condoluci, C. Buratti, T. Mahmoodi, M. Dohler, and R. Verdone, "NarrowBand-IoT : A Survey on Downlink and Uplink Perspectives," in *IEEE Wireless Communications Magazine*, Feb 2019.

[23] G. Tsoukaneri, X. Foukas, and M. K. Marina, "ASPIS: A Holistic and Practical Mechanism for Efficient MTC Support over Mobile Networks," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2017.

[24] G. Tsoukaneri, M. Condoluci, T. Mahmoodi, M. Dohler, and M. K. Marina, "Group Communications in Narrowband-IoT: Architecture, Procedures, and Evaluation," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1539–1549, Jun 2018.

[25] G. Tsoukaneri and M. K. Marina, "On Device Grouping for Efficient Multicast Communications in Narrowband-IoT," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, Jul 2018.

[26] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN)," 3rd Generation Partnership Project (3GPP), TS 36.300, Apr 2017.

[27] ——, "NR; NR and NG-RAN Overall Description," 3rd Generation Partnership Project (3GPP), TS 38.300, Oct 2018.

[28] ——, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," 3rd Generation Partnership Project (3GPP), TS 23.401, Jul 2018.

[29] ——, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321, Jul 2018.

[30] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode," 3rd Generation Partnership Project (3GPP), TS 36.304, Jul 2018.

[31] ——, "Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description," 3rd Generation Partnership Project (3GPP), TS 23.246, May 2017.

[32] D. Lecompte and F. Gabin, "Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements," *IEEE Communications Magazine*, vol. 50, no. 11, Nov 2012.

[33] G. Araniti, M. Condoluci, P. Scopelliti, A. Molinaro, and A. Iera, "Multicasting over Emerging 5G Networks: Challenges and Perspectives," *IEEE Network*, vol. 31, Mar 2017.

[34] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)," 3rd Generation Partnership Project (3GPP), TS 36.331, May 2017.

[35] ——, "Security architecture and procedures for 5G System," 3rd Generation Partnership Project (3GPP), TS 33.501, Jul 2018.

[36] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector : A Systematic Approach for Adversarial Testing of 4G/LTE," 2017.

[37] R. P. Jover, "LTE security, protocol exploits and location tracking experimentation with low-cost software radio," *CoRR*, 2016.

[38] ——, "Some key challenges in securing 5G wireless networks," 2017.

[39] E. C. Jimenez, P. K. Nakarmi, M. Naslund, and K. Norrman, "Subscription identifier privacy in 5G systems," in *2017 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, May 2017.

[40] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J. P. Seifert, "Practical attacks against privacy and availability in 4G/LTE mobile communication systems," *CoRR*, 2015.

[41] R. Borgaonkar and S. Udar, "Understanding IMSI privacy," in *BlackHat*, Aug 2014.

[42] K. Nohl, "Mobile Self-Defense," https://events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf.

[43] 3GPP, "Study on the security aspects of the next generation system," 3rd Generation Partnership Project (3GPP), TR 33.899, Mar 2016.

[44] J. Jonsson and B. Kaliski, "Public-Key Cryptography Standards (PKCS): RSA Cryptography Specifications Version 2.1," United States, 2003.

[45] "Sigfox - The Global Communications Service Provider for the Internet of Things," https://www.google.com/search?client=ubuntu&channel=fs&q=sigfox&ie=utf-8&oe=utf-8, accessed: 2018-11-5.

[46] "LoRa Alliance," https://lora-alliance.org/, accessed: 2018-11-5.

[47] Y. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A Primer on 3GPP Narrowband Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, Mar 2017.

[48] 3GPP, "Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT)," 3rd Generation Partnership Project (3GPP), TR 45.820, Aug 2016.

[49] ——, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS 36.213, Apr 2017.

[50] ——, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS 36.211, Apr 2017.

[51] ——, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)," 3rd Generation Partnership Project (3GPP), TS 24.301, Jun 2018.

[52] C. Tseng, H. Wang, F. Kuo, K. Ting, H. Chen, and G. Chen, "Delay and Power Consumption in LTE/LTE-A DRX Mechanism With Mixed Short and ong Cycles," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, Mar 2016.

[53] N. Kouzayha, N. C. Taher, and Y. Ghamri-Doudane, "Towards a better support of Machine Type Communication in LTE-networks: Analysis of random access mechanisms," in *2013 2nd International Conference on Advances in Biomedical Engineering*, Sep 2013, pp. 57–60.

[54] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86–93, Jun 2013.

[55] F. Cao and Z. Fan, "Cellular M2M network access congestion: Performance analysis and solutions," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2013, pp. 39–44.

[56] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 4–16, Jan 2014.

[57] K. Zheng, S. Ou, J. Alonso-Zarate, M. Dohler, F. Liu, and H. Zhu, "Challenges of massive access in highly dense LTE-advanced networks with machine-to-machine communications," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 12–18, June 2014.

[58] F. Ghavimi and H. Chen, "M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 525–549, Oct 2015.

[59] E. Soltanmohammadi, K. Ghavami, and M. Naraghi-Pour, "A Survey of Traffic Issues in Machine-to-Machine Communications Over LTE," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 865–884, Dec 2016.

[60] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward Massive Machine Type Cellular Communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, Feb 2017.

[61] H. S. Dhillon, H. Huang, and H. Viswanathan, "Wide-area Wireless Communication Challenges for the Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 168–174, Feb 2017.

[62] Y. Mehmood, N. Haider, M. Imran, A. Timm-Giel, and M. Guizani, "M2M Communications in 5G: State-of-the-Art Architecture, Recent Advances, and Research Challenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 194–201, Sep 2017.

[63] C. Bockelmann, N. K. Pratas, G. Wunder, S. Saur, M. Navarro, D. Gregoratti, G. Vivier, E. D. Carvalho, Y. Ji, . Stefanovi, P. Popovski, Q. Wang, M. Schellmann, E. Kosmatos, P. Demestichas, M. Raceala-Motoc, P. Jung, S. Stanczak, and A. Dekorsy, "Towards Massive Connectivity Support for Scalable mMTC Communications in 5G Networks," *IEEE Access*, vol. 6, May 2018.

[64] 3GPP, "Study on RAN Improvements for Machine-Type Communications," 3rd Generation Partnership Project (3GPP), TR 37.868, Sep 2011.

[65] S. Sheu, C. Chiu, S. Lu, and H. Lai, "Efficient data transmission scheme for MTC communications in LTE system," in *2011 11th International Conference on ITS Telecommunications*, Aug 2011.

[66] C. Karupongsiri, K. S. Munasinghe, and A. Jamalipour, "Smart meter packet transmission via the control signal of LTE networks," in *2015 IEEE International Conference on Communications (ICC)*, Jun 2015.

[67] G. Hasegawa, T. Iwai, and N. Wakamiya, "Temporal load balancing of time-driven machine type communications in mobile core networks," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015.

[68] T. Terami, T. Ohta, and Y. Kakuda, "A Method of Mobile Core Network Load Reduction Using Autonomous Clustering-Based Two-Layered Structure for Information Dissemination in Wireless Networks," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Jul 2017.

[69] S. Abe, G. Hasegawa, and M. Murata, "Design and performance evaluation of bearer aggregation method in mobile core network with C/U plane separation," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, Jun 2017.

[70] K. Tanabe, H. Nakayama, T. Hayashi, and K. Yamaoka, "An optimal resource assignment for C/D-plane virtualized mobile core networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017.

[71] Y. Chen and W. Wang, "Machine-to-Machine Communication in LTE-A," in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, Sep 2010.

[72] C. Oh, D. Hwang, and T. Lee, "Joint Access Control and Resource Allocation for Concurrent and Massive Access of M2M Devices," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4182–4192, Aug 2015.

[73] C. Kahn and H. Viswanathan, "Connectionless access for mobile cellular networks," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 26–31, Sep 2015.

[74] D. T. Wiriaatmadja and K. W. Choi, "Hybrid Random Access and Data Transmission Protocol for Machine-to-Machine Communications in Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 33–46, Jan 2015.

[75] C. Karupongsiri, K. S. Munasinghe, and A. Jamalipour, "A novel communication mechanism for Smart Meter packet transmission on LTE networks," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2016.

[76] S. Oh and J. Shin, "A three-step data transmission scheme for machine type devices in 3GPP LTE systems," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2016.

[77] J. Bai, Y. Li, and X. Guo, "Resource Allocation in Non-Orthogonal Random Access for M2M Communications," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, Jun 2018.

[78] S. Andreev, A. Larmo, M. Gerasimenko, V. Petrov, O. Galinina, T. Tirronen, J. Torsner, and Y. Koucheryavy, "Efficient small data access for machine-type

communications in LTE," in *2013 IEEE International Conference on Communications (ICC)*, Jun 2013.

[79] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Reduced M2M signaling communications in 3GPP LTE and future 5G cellular networks," in *2016 Wireless Days (WD)*, Mar 2016.

[80] T. Kwon and J. Choi, "Multi-Group Random Access Resource Allocation for M2M Devices in Multicell Systems," *IEEE Communications Letters*, vol. 16, no. 6, pp. 834–837, Jun 2012.

[81] A. Tsai, L. Wang, J. Huang, and T. Lin, "Overload Control for Machine Type Communications with Femtocells," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, Sep 2012, pp. 1–5.

[82] T. Taleb and A. Ksentini, "An efficient scheme for MTC overload control based on signaling message compression," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 342–346.

[83] G. Farhadi and A. Ito, "Group-Based Signaling and Access Control for Cellular Machine-to-Machine Communication," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sep 2013, pp. 1–6.

[84] T. Chuang, M. Tsai, and C. Chuang, "Group-Based Uplink Scheduling for Machine-Type Communications in LTE-Advanced Networks," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, Mar 2015, pp. 652–657.

[85] T. Kim, H. S. Jang, and D. K. Sung, "An Enhanced Random Access Scheme With Spatial Group Based Reusable Preamble Allocation in Cellular M2M Networks," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1714–1717, Oct 2015.

[86] U. Tefek and T. J. Lim, "Clustering and radio resource partitioning for machine-type communications in cellular networks," in *2016 IEEE Wireless Communications and Networking Conference*, Apr 2016, pp. 1–6.

[87] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Random Access for M2M Communications With QoS Guarantees," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2889–2903, Jul 2017.

[88] Z. Shao, J. Zhang, X. Sun, and H. Zhu, "Low-power consumption spacial group based random access scheme for MTC devices," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Oct 2017, pp. 1–6.

[89] T. P. C. de Andrade, L. R. Sekijima, and N. L. S. da Fonseca, "A Cluster-Based Random-Access Scheme for LTE/LTE-A Networks Supporting Massive Machine-Type Communications," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[90] H. S. Jang, B. C. Jung, and D. K. Sung, "Dynamic Access Control with Resource Limitation for Group Paging-Based Cellular IoT Systems," *IEEE Internet of Things Journal*, Oct 2018.

[91] L. Li, X. Wen, Z. Lu, Q. Pan, and W. Jing, "Pre-Backoff Based Random Access with Priority for 5G Machine-Type Communication," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Dec 2017.

[92] J. Cheng, C. Lee, and T. Lin, "Prioritized Random Access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, Dec 2011, pp. 368–372.

[93] A. Larmo and R. Susitaival, "RAN overload control for Machine Type Communications in LTE," in *2012 IEEE Globecom Workshops*, Dec 2012.

[94] C. M. Chou, C. Y. Huang, and C. Chiu, "Loading prediction and barring controls for machine type communication," in *2013 IEEE International Conference on Communications (ICC)*, Jun 2013.

[95] Z. Wang and V. W. S. Wong, "Joint access class barring and timing advance model for machine-type communications," in *2014 IEEE International Conference on Communications (ICC)*, Jun 2014.

[96] Z. Zhang, H. Chao, W. Wang, and X. Li, "Performance Analysis and UE-Side Improvement of Extended Access Barring for Machine Type Communications in LTE," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, May 2014.

[97] T. Lin, C. Lee, J. Cheng, and W. Chen, "PRADA: Prioritized Random Access With Dynamic Access Barring for MTC in 3GPP LTE-A Networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2467–2472, Jun 2014.

[98] M. K. Giluka, N. Rajoria, A. C. Kulkarni, V. Sathya, and B. R. Tamma, "Class based dynamic priority scheduling for uplink to support M2M communications in LTE," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar 2014.

[99] M. Tavana, V. Shah-Mansouri, and V. W. S. Wong, "Congestion control for bursty M2M traffic in LTE networks," in *2015 IEEE International Conference on Communications (ICC)*, Jun 2015, pp. 5815–5820.

[100] Z. Wang and V. W. S. Wong, "Optimal Access Class Barring for Stationary Machine Type Communication Devices With Timing Advance Information," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5374–5387, Oct 2015.

[101] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9847–9861, Dec 2016.

[102] J. Moon and Y. Lim, "Adaptive Access Class Barring for Machine-Type Communications in LTE-A," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Jul 2016.

[103] I. Leyva-Mayorga, L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, and V. Casares-Giner, "Performance analysis of access class barring for handling massive M2M traffic in LTE-A networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016.

[104] D. Kim, W. Kim, and S. An, "Adaptive random access preamble split in LTE," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jul 2013.

[105] H. Wu, C. Zhu, R. J. La, X. Liu, and Y. Zhang, "FASA: Accelerated S-ALOHA Using Access History for Event-Driven M2M Communications," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1904–1917, Dec 2013.

[106] J. Choi, "On the Adaptive Determination of the Number of Preambles in RACH for MTC," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1385–1388, Jul 2016.

[107] M. Vilgelm, H. M. Grsu, W. Kellerer, and M. Reisslein, "LATMAPA: Load-Adaptive Throughput- MAximizing Preamble Allocation for Prioritization in 5G Random Access," *IEEE Access*, Jan 2017.

[108] W. Li, Q. Du, L. Liu, P. Ren, Y. Wang, and L. Sun, "Dynamic Allocation of RACH Resource for Clustered M2M Communications in LTE Networks," in *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*, Oct 2015.

[109] Y. Wu, N. Zhang, and G. Kang, "Dynamic Resource Allocation with QoS Guarantees for Clustered M2M Communications," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, Mar 2017.

[110] Z. Ding, P. Fan, and H. V. Poor, "Impact of User Pairing on 5G Nonorthogonal Multiple-Access Downlink Transmissions," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6010–6023, Aug 2016.

[111] K. S. Ko, M. J. Kim, K. Y. Bae, D. K. Sung, J. H. Kim, and J. Y. Ahn, "A Novel Random Access for Fixed-Location Machine-to-Machine Communications in OFDMA Based Systems," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1428–1431, Sep 2012.

[112] G. C. Madueo, C. Stefanovic, and P. Popovski, "Efficient LTE access with collision resolution for massive M2M communications," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014.

[113] M. Taherzadeh, H. Nikopour, A. Bayesteh, and H. Baligh, "SCMA Codebook Design," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sep 2014, pp. 1–5.

[114] H. Nikopour and H. Baligh, "Sparse code multiple access," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep 2013, pp. 332–336.

[115] X. Dai, S. Chen, S. Sun, S. Kang, Y. Wang, Z. Shen, and J. Xu, "Successive interference cancelation amenable multiple access (sama) for future wireless communications," in *2014 IEEE International Conference on Communication Systems*, Nov 2014, pp. 222–226.

[116] S. Chen, B. Ren, Q. Gao, S. Kang, S. Sun, and K. Niu, "Pattern division multiple accessa novel nonorthogonal multiple access for fifth-generation radio networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3185–3196, Apr 2017.

[117] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the Performance of Non-Orthogonal Multiple Access in 5G Systems with Randomly Deployed Users," *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1501–1505, Dec 2014.

[118] Y. Beyene, C. Boyd, K. Ruttik, C. Bockelmann, O. Tirkkonen, and R. Jantti, "Compressive sensing for MTC in new LTE uplink multi-user random access channel," in *AFRICON 2015*, Sep 2015.

[119] N. Zhang, J. Wang, G. Kang, and Y. Liu, "Uplink Nonorthogonal Multiple Access in 5G Systems," *IEEE Communications Letters*, vol. 20, no. 3, pp. 458–461, Mar 2016.

[120] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-Orthogonal Random Access for 5G Networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4817–4831, Jul 2017.

[121] J. Choi, "Layered Non-Orthogonal Random Access With SIC and Transmit Diversity for Reliable Transmissions," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1262–1272, Mar 2018.

[122] A. Alexiou, C. Bouras, V. Kokkinos, A. Papazois, and G. Tsichritzis, "Efficient mcs selection for mbsfn transmissions over lte networks," in *2010 IFIP Wireless Days*, Oct 2010, pp. 1–5.

[123] ——, "Spectral efficiency performance of mbsfn-enabled lte networks," in *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct 2010, pp. 361–367.

[124] S. Lu, Y. Cai, L. Zhang, J. Li, P. Skov, C. Wang, and Z. He, "Channel-aware frequency domain packet scheduling for mbms in lte," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, Apr 2009, pp. 1–5.

[125] G. Araniti, M. Condoluci, L. Militano, and A. Iera, "Adaptive Resource Allocation to Multicast Services in LTE Systems," *IEEE Transactions on Broadcasting*, vol. 59, no. 4, pp. 658–664, Dec 2013.

[126] J. F. Monserrat, J. Calabuig, A. Fernandez-Aguilella, and D. Gomez-Barquero, "Joint delivery of unicast and e-mbms services in lte networks," *IEEE Transactions on Broadcasting*, vol. 58, no. 2, pp. 157–167, Jun 2012.

[127] J. Park, J. Hwang, Q. Li, Y. Xu, and W. Huang, "Optimal DASH-Multicasting Over LTE," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4487–4500, May 2018.

[128] G. Araniti, V. Scordamaglia, M. Condoluci, A. Molinaro, and A. Iera, "Efficient frequency domain packet scheduler for point-to-multipoint transmissions in lte networks," in *2012 IEEE International Conference on Communications (ICC)*, Jun 2012, pp. 4405–4409.

[129] G. Araniti, M. Condoluci, A. Iera, A. Molinaro, J. Cosmas, and M. Behjati, "A Low-Complexity Resource Allocation Algorithm for Multicast Service Delivery in OFDMA Networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 358–369, Jun 2014.

[130] G. Araniti, M. Condoluci, A. Orsino, A. Iera, A. Molinaro, and J. Cosmas, "Evaluating the performance of multicast resource allocation policies over LTE systems," in *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, Jun 2015, pp. 1–6.

[131] O. Karimi, J. Liu, and Z. Wang, "Power-Efficient Resource Utilization in Cellular Multimedia Multicast," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Dec 2015, pp. 134–143.

[132] M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT Machine Age With 5G: Machine-Type Multicast Services for Innovative Real-Time Applications," *IEEE Access*, vol. 4, pp. 5555–5569, May 2016.

[133] R. Jagadeesha, J. Sheu, and W.-K. Hon, "User satisfaction based resource allocation schemes for multicast in D2D networks," in *2017 European Conference on Networks and Communications (EuCNC)*, Jun 2017, pp. 1–5.

[134] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[135] F. Corno, L. D. Russis, and J. P. Senz, "On the advanced services that 5g may provide to iot applications," in *2018 IEEE 5G World Forum (5GWF)*, Jul 2018, pp. 528–531.

[136] S. Pizzi, M. Condoluci, A. Molinaro, A. Iera, G. . Muntean, and G. Araniti, "Resource Balancing of Unicast and Multicast Wireless Multimedia Services in 5G Networks," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun 2018, pp. 1–5.

[137] J. Montalban, P. Scopelliti, M. Fadda, E. Iradier, C. Desogus, P. Angueira, M. Murroni, and G. Araniti, "Multimedia Multicast Services in 5G Networks: Subgrouping and Non-Orthogonal Multiple Access Techniques," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 91–95, Mar 2018.

[138] A. Bhardwaj, "Resource management for device-to-device multicast in LTE-a network," in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, Jan 2016.

[139] S. S. Moghaddam and M. Ghasemi, "A Low-complex/High-throughput Resource Allocation for Multicast D2D Communications," in *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, Sep 2018.

[140] H. Li and X. Huang, "Multicast Systems With Fair Scheduling in Non-identically Distributed Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8835–8844, Oct 2017.

[141] J. Zhao, W. Xu, X. Li, and J. Lin, "User-interest-aware multicast group formation in OFDM networks using matching theory," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct 2017, pp. 1–6.

[142] X. Huang, Z. Zhao, and H. Zhang, "Cooperate Caching with Multicast for Mobile Edge Computing in 5G Networks," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Jun 2017.

[143] B. Zhou, Y. Cui, and M. Tao, "Optimal Dynamic Multicast Scheduling for Cache-Enabled Content-Centric Wireless Networks," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2956–2970, Jul 2017.

[144] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting Caching and Multicast for 5G Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2995–3007, Apr 2016.

[145] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive Scheme for Caching YouTube Content in a Cellular Network: Machine Learning Approach," *IEEE Access*, vol. 5, pp. 5870–5881, Mar 2017.

[146] S. Mrad, S. Hamouda, and H. Rezig, "Graph Theory based multicast caching for better energy saving in dense small cell networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jun 2017, pp. 2015–2020.

[147] Y. Lu, W. Chen, and H. V. Poor, "Multicast Pushing With Content Request Delay Information," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1078–1092, Mar 2018.

[148] J. Hong, S. H. Chae, and K. Lee, "Network Throughput Gain of Multicast With User Caching in Heavy Traffic Downlink," *IEEE Access*, vol. 6, pp. 26 626–26 635, May 2018.

[149] S. Mrad, S. Hamouda, and B. T. Maharaj, "Energy-Efficient Multicast/Unicast Edge Caching for Dense Small Cell Networks with Graph Theory," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, Jun 2018, pp. 1–5.

[150] Z. Zhang, Z. Ma, M. Xiao, X. Lei, Z. Ding, and P. Fan, "Fundamental Tradeoffs of Non-Orthogonal Multicast, Multicast, and Unicast in Ultra-Dense Networks," *IEEE Transactions on Communications*, vol. 66, no. 8, pp. 3555–3570, Aug 2018.

[151] A. T. Koc, S. C. Jha, R. Vannithamby, and M. Torlak, "Device Power Saving and Latency Optimization in LTE-A Networks Through DRX Configuration," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2614–2625, May 2014.

[152] J. Liang, J. Chen, P. Hsieh, and Y. Tseng, "Two-Phase Multicast DRX Scheduling for 3GPP LTE-Advanced Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1839–1849, Jul 2016.

[153] M. K. Maheshwari, M. Agiwal, N. Saxena, and A. Roy, "Hybrid Directional Discontinuous Reception (HD-DRX) for 5G Communication," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1421–1424, Jun 2017.

[154] C. Chang and J. Chen, "Adjustable Extended Discontinuous Reception Cycle for Idle-State Users in LTE-A," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2288–2291, Nov 2016.

[155] N. M. Balasubramanya, L. Lampe, G. Vos, and S. Bennett, "DRX With Quick Sleeping: A Novel Mechanism for Energy-Efficient IoT Using LTE/LTE-A," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 398–407, Jun 2016.

[156] C. Chang and J. Chen, "UM Paging: Unified M2M Paging with Optimal DRX Cycle," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 886–900, Mar 2017.

[157] J. Liang, J. Chen, H. Cheng, and Y. Tseng, "An energy-efficient sleep scheduling with qos consideration in 3gpp lte-advanced networks for internet of things," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 13–22, Mar 2013.

[158] N. Kaur and S. K. Sood, "An Energy-Efficient Architecture for the Internet of Things (IoT)," *IEEE Systems Journal*, vol. 11, no. 2, pp. 796–805, Jun 2017.

[159] C. Wang, P. Li, C. Tsai, and K. Feng, "Load-balanced user association and resource allocation under limited capacity backhaul for small cell networks," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sep 2016.

[160] S. Hailu, P. Lunden, E. Virtej, N. Kolehmainen, O. Tirkkonen, and C. Wijting, "DRX-Aware Power and Delay Optimized Scheduler for Bursty Traffic Transmission," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015.

[161] P. Liu, K. Wu, J. Liang, J. Chen, and Y. Tseng, "Energy-Efficient Uplink Scheduling for Ultra-Reliable Communications in NB-IoT Networks," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep 2018, pp. 1–5.

[162] H. Ramazanali and A. Vinel, "Performance evaluation of lte/lte-a drx: A markovian approach," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 386–397, Jun 2016.

[163] K. Kanwal, G. A. Safdar, M. Ur-Rehman, and X. Yang, "Energy Management in LTE Networks," *IEEE Access*, vol. 5, Mar 2017.

[164] K. Samdanis, T. Taleb, D. Kutscher, and M. Brunner, "Self Organized Network Management Functions for Energy Efficient Cellular Urban Infrastructures," *Mobile Networks and Applications*, vol. 17, no. 1, Feb.

[165] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, Oct 2011.

[166] T. Chen, Y. Yang, H. Zhang, H. Kim, and K. Horneman, "Network energy saving technologies for green wireless access networks," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 30–38, Oct 2011.

[167] J. B. Rao and A. O. Fapojuwo, "A Survey of Energy Efficient Resource Management Techniques for Multicell Cellular Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 154–180, May 2014.

[168] K. Yang, L. Wang, S. Wang, and X. Zhang, "Optimization of Resource Allocation and User Association for Energy Efficiency in Future Wireless Networks," *IEEE Access*, vol. 5, Jun 2017.

[169] G. Liu, F. R. Yu, H. Ji, and V. C. M. Leung, "Energy-efficient resource allocation in shared full-duplex relaying cellular networks," in *2014 IEEE Global Communications Conference*, Dec 2014, pp. 2631–2636.

[170] A. Bousia, E. Kartsakli, L. Alonso, and C. Verikoukis, "Dynamic energy efficient distance-aware Base Station switch on/off scheme for LTE-advanced," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012.

[171] M. Feng, S. Mao, and T. Jiang, "Base Station ON-OFF Switching in 5G Wireless Networks: Approaches and Challenges," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 46–54, Aug 2017.

[172] L. Suarez, L. Nuaymi, and J.-M. Bonnin, "Energy-efficient BS switching-off and cell topology management for macro/femto environments," *Computer Networks*, vol. 78, 2015.

[173] P. Frenger, P. Moberg, J. Malmodin, Y. Jading, and I. Godor, "Reducing Energy Consumption in LTE with Cell DTX," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, May 2011.

[174] F. Cao and Z. Fan, "The tradeoff between energy efficiency and system performance of femtocell deployment," in *2010 7th International Symposium on Wireless Communication Systems*, Sep 2010, pp. 315–319.

[175] K. Davaslioglu, C. C. Coskun, and E. Ayanoglu, "Energy-Efficient Resource Allocation for Fractional Frequency Reuse in Heterogeneous Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5484–5497, Oct 2015.

[176] Y. Chiang and W. Liao, "Green Multicell Cooperation in Heterogeneous Networks With Hybrid Energy Sources," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 7911–7925, Dec 2016.

[177] C. C. Coskun and E. Ayanoglu, "Energy-spectral efficiency tradeoff for heterogeneous networks with QoS constraints," in *2017 IEEE International Conference on Communications (ICC)*, May 2017.

[178] H. Chang and M. Tsai, "Optimistic DRX for Machine-Type Communications in LTE-A Network," *IEEE Access*, vol. 6, Jan 2018.

[179] G. Szabo, G. Pongracz, I. Godor, R. Coster, and M. Sintorn, "Service aware adaptive DRX scheme," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014.

[180] A. Sehati and M. M. Ghaderi, "Online Energy Management in IoT Applications," in *INFOCOM*, 04 2018.

[181] M. Lauridsen, G. Berardinelli, F. M. L. Tavares, F. Frederiksen, and P. Mogensen, "Sleep Modes for Enhanced Battery Life of 5G Mobile Terminals," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–6.

[182] J. Brusey, J. Kemp, E. Gaura, R. Wilkins, and M. Allen, "Energy Profiling in Practical Sensor Networks: Identifying Hidden Consumers," *IEEE Sensors Journal*, Aug 2016.

[183] H. Malik, H. Pervaiz, M. M. Alam, Y. L. Moullec, A. Kuusik, and M. A. Imran, "Radio Resource Management Scheme in NB-IoT Systems," *IEEE Access*, Mar 2018.

[184] C. Chen, A. C. Huang, S. Huang, and J. Chen, "Energy-saving scheduling in the 3GPP narrowband Internet of Things (NB-IoT) using energy-aware machine-to-machine relays," in *2018 27th Wireless and Optical Communication Conference (WOCC)*, Apr 2018.

[185] A. H. Bassel, A.-H. Akram, G. C. Karina, C. Sathyanarayanan, and S. Kandeepan, "Energy-Efficient IoT for 5G: A Framework for Adaptive Power and Rate Control," in *The 12th International Conference on Signal Processing and Communication Systems*, Jan 2018.

[186] J. M. Liang, K. R. Wu, J. J. Chen, P. Y. Liu, and Y. C. Tseng, "Energy-Efficient Uplink Resource Units Scheduling for Ultra-Reliable Communications in NB-IoT Networks," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–17, Jul 2018.

[187] P. A. Maldonado, P. Ameigeiras, J. P. Garzon, J. J. R. Munoz, and J. M. L. Soler, "Optimized LTE Data Transmission Procedures for IoT: Device Side Energy Consumption Analysis," *CoRR*, 2017.

[188] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Narrowband IoT Data Transmission Procedures for Massive Machine-Type Communications," *IEEE Network*, vol. 31, no. 6, pp. 8–15, Nov 2017.

[189] E. Azoidou, Z. Pang, Y. Liu, D. Lan, G. Bag, and S. Gong, "Battery Lifetime Modeling and Validation of Wireless Building Automation Devices in Thread," *IEEE Transactions on Industrial Informatics*, Jul 2018.

[190] B. Martinez, M. Montn, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sensors Journal*, Oct 2015.

[191] J. Finnegan, "An Analysis of the Energy Consumption of LPWA-based IoT Devices," in *IEEE International Symposium on Networks, Computers and Communications (ISNCC)*, Jun 2018.

[192] E. Morin, M. Maman, R. Guizzetti, and A. Duda, "Comparison of the Device Lifetime in Wireless Networks for the Internet of Things," *IEEE Access*, Apr 2017.

[193] R. S. Sharan, Y. Wei, and S. H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, Mar 2017.

[194] M. E. Soussi, P. Zand, F. Pasveer, and G. Dolmans, "Evaluating the Performance of eMTC and NB-IoT for Smart City Applications," in *2018 IEEE International Conference on Communications (ICC)*, May 2018.

[195] H. Bello, J. Xin, Y. Wei, and M. Chen, "Energy-Delay Evaluation and Optimization for NB-IoT PSM with Periodic Uplink Reporting," *IEEE Access*, vol. PP, 12 2018.

[196] C. Y. Yeoh, A. bin Man, Q. M. Ashraf, and A. K. Samingan, "Experimental assessment of battery lifetime for commercial off-the-shelf NB-IoT module," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Feb 2018.

[197] "Mobile Af Hoc and Sensor Systems (MASS) 2017," https://mass2017.engineering.osu.edu/, accessed: 2018-10-25.

[198] 3GPP, "Study on Machine-Type Communications (MTC) and other mobile data applications communications enhancements," 3rd Generation Partnership Project (3GPP), TR 23.887, Dec 2013.

[199] ——, "Study on Small data transmission enhancements for UMTS," 3rd Generation Partnership Project (3GPP), TR 25.705, Jul 2015.

[200] "OpenAirInterface-5G software alliance for democratising wireless innovation," http://www.openairinterface.org/, accessed: 2018-10-25.

[201] "Abstract Syntax Notation One (ASN.1)," https://portal.etsi.org/CTI/ApproachToTesting/SpecLanguages/ASN.1.htm, accessed: 2018-11-5.

[202] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-Orthogonal Multiple Access (NOMA) for Cellular Future Radio Access," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, Jun 2013, pp. 1–5.

[203] L. Dai, B. Wang, Y. Yuan, S. Han, C. I, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, Sep 2015.

[204] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "Successive Interference Cancellation: A Back-of-the-envelope Perspective," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, no. 17, Oct 2010, pp. 1–6.

[205] "IEEE International Symposium on Personal, Indoor and Mobile Radio Communications," http://pimrc2019.ieee-pimrc.org/, accessed: 2019-03-19.

[206] 3GPP, "Access class barring and overload protection," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.898, Mar 2005.

[207] ——, "Study on 3D channel model for LTE," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.873, Jan 2018.

[208] M. Condoluci, G. Araniti, T. Mahmoodi, and M. Dohler, "Enabling the IoT Machine Age With 5G: Machine-Type Multicast Services for Innovative Real-Time Applications," *IEEE Access*, vol. 4, May 2016.

[209] Ericsson, "Massive IoT in the city," Stocholm, Tech. Rep., 2016, accessed: 2018-11-22. [Online]. Available: https://www.ericsson.com/en/mobility-report/massive-iot-in-the-city

[210] J. Calabuig, J. F. Monserrat, D. Gozalvez, and D. D. Gomez-Barquero, "AL-FEC for streaming services in LTE E-MBMS," *EURASIP Journal on Wireless Communications and Networking*, Mar 2013.

[211] C. Bouras, N. Kanakis, V. Kokkinos, and A. Papazois, "AL-FEC for streaming services over LTE systems," in *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Oct 2011.

[212] G. M. IoT, "3GPP Low Power Wide Are Technologies," White Paper, Oct 2016.

[213] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. [Online]. Available: http://mitpress.mit.edu/books/introduction-algorithms

[214] V. Chvatal, "A Greedy Heuristic for the Set-Covering Problem," *Math. Oper. Res.*, vol. 4, no. 3, pp. 233–235, Aug 1979.

[215] "3GPP TSG-RAN WG2 Meeting #57bis, R2-071284, St. Julian's, Malta, Agenda item:5.2.3, Source: Nokia, Title: DRX parameters in LTE."

[216] "3GPP TSG-RAN WG2 Meeting #57bis, R2-071285, St. Julian's, Malta, Agenda item:5.2.3, Source: Nokia, Title: DRX parameters in LTE."

[217] M. Lichtman, R. P. Jover, M. Labib, R. Rao, V. Marojevic, and J. H. Reed, "LTE/LTE-A jamming, spoofing, and sniffing: threat assessment and mitigation," *IEEE Communications Magazine*, Apr 2016.

[218] "GPy," https://pycom.io/product/gpy/.

[219] "Quectel LTE BC95 NB-IoT Module," https://www.quectel.com/product/bc95.htm.

[220] "The first NB-IoT shield for Arduino: supported by T-Mobile," https://www.kickstarter.com/projects/sodaq/the-first-nb-iot-shield-for-arduino-supported-by-t.

[221] "Arduino Uno Rev3 - Arduino Starter Kit," https://store.arduino.cc/arduino-uno-rev3.

[222] "Expansion Board 2.0," https://pycom.io/hardware/expansion-board-2-0-specs.

[223] "E7515A UXM Wireless Test Set," https://www.keysight.com/en/pd-2372474-pn-E7515A/uxm-wireless-test-set.

[224] "High Voltage Power Monitor," https://www.msoon.com/online-store.

[225] 3GPP, "Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 25.331, Jul 2018.

[226] ——, "3GPP System Architecture Evolution (SAE); Security architecture," 3rd Generation Partnership Project (3GPP), TS 33.401, Jul 2018.

[227] "Crypto++ Library 7.0 - Free C++ Class Library of Cryptographic Schemes," https://www.cryptopp.com/.

[228] "Python Cryptography Toolkit," https://pypi.org/project/pycrypto/.

[229] S. Landstrom, J. Bergstrom, E. Westerberg, and D. Hammarwall, "NB-IoT: A Sustainable Technology for Connecting Billions of Devices," Ericsson, Review, 2016.

[230] 3GPP, "Radio transmission and reception; Part 3: Radio Resource Management (RRM) conformance testing ," 3rd Generation Partnership Project (3GPP), TS 36.521, Apr 2017.

[231] ——, "2 step RACH procedure consideration," 3rd Generation Partnership Project (3GPP), TDoc R1-1700792, for Discussion.

[232] ——, "Discussions on 2 Steps RACH Procedure," 3rd Generation Partnership Project (3GPP), TDoc R1-1700668, for Discussion.