Comparative Analysis of Different Methods for Reducing the Energy Consumption and latency in NB-IoT systems

by

Mohammad Ibteda Ali Swianto (160021113)
Shahabuddin Akhand (160021094)
Nassak Saif (160021009)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

February 2021

Comparative Analysis of Different Methods for Reducing the Energy Consumption and latency in NB-IoT systems

Approved by:

nafizibhamid

Nafiz Imtiaz Bin Hamid

Supervisor and Assistant Professor
Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Boardvazar, Gazipur-1704.

Date: March 10, 2021

Contents

List of Tables

List of Figures

List of Acronyms

ACK Acknowledgement.

ARQ Automatic Repeat Request.

AWGN Additive White Gaussian Noise.

BER Bit Error Rate.

BLER Block Error Rate.

CoAP Constraint Application Protocol.

CPU Core Processor Unit.

CRC Cyclic Redundancy Check.

DCI Downlink Control Information.

DOCSIS Data Over Cable Service Interface Specification.

DS Dynamic Scheduling.

DTLS Datagram Transport Layer Security.

DWDM Dense Wavelength Division Multiplexing.

eDRX Extended Discontinuous Reception.

EEACC Energy-Efficient and Adaptive Channel Coding.

eNB E-UTRAN NodeB.

EPS	Evolved Packet System.
FDMA	Frequency Division Multiple Access.
FEC	Forward Error Correction.
FFT	Fast Fourier Transform.
GFEC	Generic FEC.
HARQ	Hybrid Automatic Repeat Request.
HD-FEC	Hard Decision FEC.
IoT	Internet of Things.
LDPC	Low Density Parity Check.
LPWAN	Low Power Wide Area Network.
LTE	Long-Term Evolution.
MCS	Modulation Coding Scheme.
mDNS	Multicast Domain Name System.
MME	Mobility Management Entity.
NACK	non-Acknowledgement.
NAS	Non-Access Stratum.
NB-IoT	Narrow Band Internet of Things.
NCG	Net Coding Gain.
NPDSCH	Narrowband Physical Downlink Shared Channel.
NPRACH	Narrowband Physical Random Access Channel.

NPUSCH Narrowband Physical Uplink Shared Channel.

OFDM Orthogonal Frequency Division Multiplexing.

PBESM Prediction Based Energy Saving Mechanism.

PDN Packet Data Network.

PIE Packet Inspection Entity.

PPE Packet Prediction Entity.

PSM Power Saving Mode.

RNTI Radio Network Temporary Identifier.

RRC Radio Recourse Control.

RS Radio System.

RX Receive mode.

SD-FEC Soft Decision FEC.

SG Scheduling Grant.

SMS Short Message Service.

SPS Semi-Persistent Scheduling.

SR Scheduling Request.

TAU Tracking Area Update.

TCP Transmission Control Protocol.

UE User Equipment.

VoIP Video over Internet Protocol.

Acknowledgement

*First, we would extend our deepest gratitude to the Almighty Allah, our Creator, Who had created us and instilled within us, the intellect to educate ourselves with the knowledge of the physical world and thus conduct our research on this thesis success-fully. We would like to express our gratitude for **Asst. Prof. Nafiz Imtiaz Bin Hamid**, for his guidance in moments of turmoil and confusion during our research. We would like to thank our parents and all our family who were there for us in moments of frustration. They gave us the motivation to continue our work through thick and thin. Finally we would like to show our appreciation and gratitude toward IUT for providing this opportunity, which will prove to be one of the defining moments in our lives.*

Abstract

The internet of things or IoT is a new and rising technology. Which compromises of a new kind of UE with different kinds of needs. This opens up an entirely new field of communication technology, Low Power Wide Area Network or LPWAN for short. To compete with rising LPWANs like LoRa and Sigfox, 3gpp, on release 13, froze the specifications for NB-IoT. Inheriting many features from LTE, NB-IoT was developed with ample opportunites for improvement. In this paper, we compare the various methods that were used in NB-IoT to reduce energy consumption and latency with novel methods researchers have proposed in recent years. To ultimately come up with an optimized NB-IoT with the most conservative amounts of energy consumption and latency.

Chapter 1

Introduction

1.1 IoT

The Internet of Things (IoT), also called the Internet of Everything or the Industrial Internet, is a new technology paradigm envisioned as a global network of machines and devices capable of interacting with each other. The IoT is recognized as one of the most important areas of future technology and is gaining vast attention from a wide range of industries.[1]

With the digital revolution, every aspect of our lives is monitored and maintained by machines. The integration of IoT devices into the mix was only inevitable. It is one of the major drivers of the next generation of network-integrated devices.

As identified by Atzori et. al.[2], Internet of Things comprises three paradigms – internet-oriented (capable of communicating), things oriented (capable of monitoring) and semantic oriented (capable of gathering useful data). Although this type of description is required since the subjects in question consists of various disciplines, IoT can fully realize its potential only in an application field where the three paradigms intersect.[3]

This establishes that a connection to the internet is one of the major focuses of this rapidly growing field. The IoT however is comprised of unique devices with specific requirements:

- The ability to connect large numbers of heterogeneous IoT elements
- High reliability
- Real-time awareness with low latency
- The ability to secure all traffic flows
- Programmability for application customization
- Traffic monitoring and management at the device level
- Low cost connectivity for large number of devices/sensors

1.2 LPWANs

For this reason researchers introduced a technology, low-power wide area network (**LPWAN**), which is:

- **Long range:** The operating range of LPWAN technology varies from 1-5km in urban areas to 10-40 km in rural settings. It can also enable effective data communication in previously infeasible indoor and underground locations.
- **Low power:** Optimized for power consumption, LPWAN transceivers can run on small, inexpensive batteries for up to 20 years
- **Low cost:** LPWAN's simplified, lightweight protocols reduce complexity in hardware design and lower device costs. Its long range combined with a star topology reduce expensive infrastructure requirements, and the use of license-free or licensed bands reduce network costs.[4]

Figure 1.1 Illustrates LPWANs in terms of range and Data rate. It demonstrates how LPWAN sits perfectly between other communication technologies to meet IoT requirements. Many LPWAN technologies have risen in the licensed as well as unlicensed frequency bandwidth. Among them, Sigfox, LoRa, and NB-IoT are today's leading emergent technologies. Yet they have many technical differences.[4] The

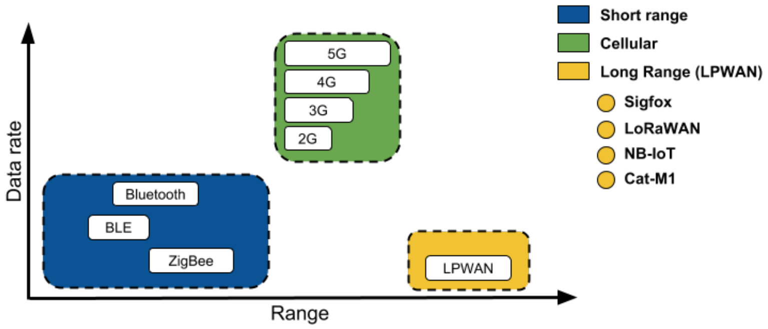


Figure 1.1: IoT services according to range and Data Rate

major difference being NB-IoT uses licensed bands, while Sigfox and LoRaWAN use unlicensed ISM bands. NB-IoT also has a higher maximum data rate and payload length.

Sigfox is a company in France, it is an LPWAN operator which operates and commercializes its own IoT solution in 31 countries and is still under rollout worldwide owing to the partnership with various network operators. Although LoRa was first developed before Sigfox, in France, it was standardized much later LoRa-Alliance and deployed to 42 countries. To compete with these LPWAN technologies 3GPP standardized NB-IoT in release 13. It was based on the previous cellular infrastructure of LTE and modified to use narrow-band radio technology. And provide equal or better communication for IoT devices using a cellular network.

1.3 Motivation & Scopes

Since NB-IoT is one of the only major LPWAN technologies which uses licenced LTE spectra for communication (Sigfox and LoRa uses unlicensed spectra) it provides the best Quality of service. It doesn't have to deal with interference, it also provides lower latencies, at the cost of energy consumption. Along with being able to carry a larger payload, when compared to LoRa and Sigfox, it is also very easily scalable. But all of these advantages come with a major drawback, energy consumption. UEs in NB-IoT networks consume more energy due to synchronous communication and its OFDM/FDMA access modes use more peak current.[6]

Thus one of the major challenges for NB-IoT is to reduce energy consumption while also maintaining its advantage of latency.

1.3.1 Thesis Outline

Our thesis strives to optimize NB-IoT to be more suitable for modern IoT needs. We first analyze NB-IoT and discuss the various mechanisms already present in IoT in chapter 2. Chapter 3 describes novel mechanisms that researchers have proposed to improve aspects of NB-IoT. The inner workings of these novel mechanisms are then discussed in chapter 4. And finally chapter 5 explores how we went about our research to decide the optimal mechanisms to use for NB-IoT.

Chapter 2

Overview of NB-IoT

2.1 Legacy (Existing) NB-IoT

NB-IoT was designed for much simpler devices when compared to devices which use LTE. Thus NB-IoT uses a simpler form of LTE to reduce complexity and power consumption. It uses only one HARQ for both uplink and downlink. Scheduling in NB-IoT is either Dynamic scheduling (DS) or Semi Persistent Scheduling (SPS). For error correction NB-IoT uses FEC channel coding method with less retransmissions than LTE. It uses eDRX and PSM to save power during the RRC connected mode, and gives the user the ability to change eDRX timers (on the UE side) to a certain degree. In this section we will discuss all these methods and finally introduce novel methods which researchers have proposed.

2.1.1 Semi-Persistent Sscheduling

In NB-IoT, data transmissions are typically scheduled using dynamic scheduling (DS), in which UEs send scheduling requests (SR) to the eNB and then collect scheduling grants (SG) for uplink data transmissions. The complex scheduling scheme is communicated by these signalling signals, which increases latency. As seen in Fig. 2.1, in SPS, the eNB preallocates resources to the UEs so that they can send their data without using the traditional SR-SG procedure [5]. For UEs that send routine and periodic data transmissions, this technique is very efficient.

When UEs don't have any data to send, they use their scheduled resources to send padding information.

This results in spectral inefficiency and higher energy use. SPS can be used to schedule devices with intermittent traffic patterns to improve spectral performance, while DS can be used to schedule devices with irregular traffic.

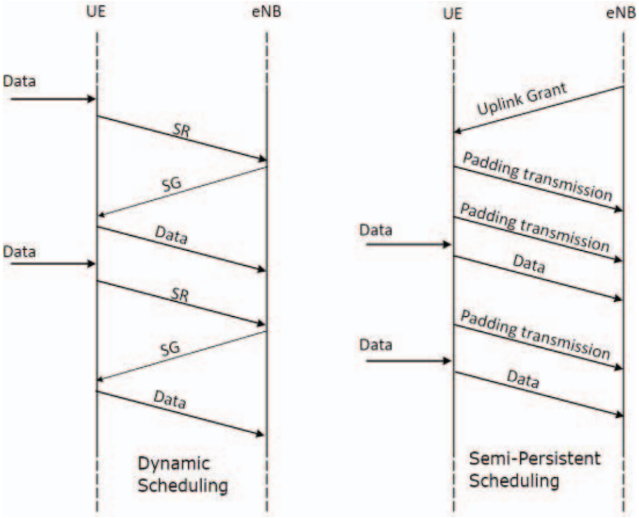


Figure 2.1: Illustration of the two scheduling types (i.e. Dynamic and Semi-Persistent Scheduling) in NB-IoT

2.1.2 Automatic Repeat Request (ARQ)

Automatic Repeat Request (ARQ) is a data transmission error-control mechanism that employs acknowledgements (or negative acknowledgements) and timeouts to ensure reliable data transmission over a shaky communication link. In an ARQ scheme, the receiver detects whether the received packet is in error using an error detection code, typically a Cyclic Redundancy Check (CRC).The transmitter is notified by sending a positive acknowledgement if no errors are detected in the received data. If an error is detected, the receiver discards the packet and sends the transmitter a negative acknowledgement, requesting a re-transmission.

A short message sent by the receiver to the transmitter to indicate whether it has correctly or incorrectly received a data packet is known as an Acknowledgment

(ACK) or Negative Acknowledgement (NACK).If the sender does not receive an acknowledgment before the timeout, it typically re-transmits the packet until it receives an acknowledgement or completes a predetermined number of retransmissions. The ARQ protocol is divided into three groups.

- Stop-and-Wait ARQ is the fundamental frame of ARQ convention where the sender sends one packet at a time and after that holds up for an ACK or NACK flag from the collector . The collector sends an ACK flag after receipt of a good bundle. In case the ACK does not reach the sender some time recently the timeout, the sender re-sends the same bundle.
- The sender continuously sends a number of packets (determined by the length of the transmission window) without receiving an ACK signal from the receiver in the Go-Back-N ARQ protocol. The receiver system keeps track of the sequence number of the next packet it expects to receive and sends it along with each ACK.If it's a repeat of a packet it's already acknowledged or a packet with a sequence number higher than the one planned, the receiver will ignore any packet that doesn't have the same sequence number it expects.When the sender has sent all of the packets in its transmission window in a sequential order, it will check to see if all of them have been acknowledged before proceeding with the next sequence number after the one that was last acknowledged.
- Selective Repeat ARQ is a type of the ARQ protocol for transmitting and acknowledging packets or fragments of packets in which the transmission mechanism continues to send a specified number of packets even after a packet is lost. Unlike Go-Back-N ARQ, after an initial mistake, the receiving mechanism will continue to accept and acknowledge packets sent.The receiver system keeps track of the sequence number of the first packet it hasn't received and sends it with every ACK. If a packet from the sender fails to reach the recipient, the sender continues to send packets until the window is empty.The receiver continues to fill its receiving window with subsequent packets, responding with an ACK containing the sequence number

of the first missing packet each time. After sending all of the packets in its window, the sender re-sends the packet number indicated by the ACKs and then starts where it left off.

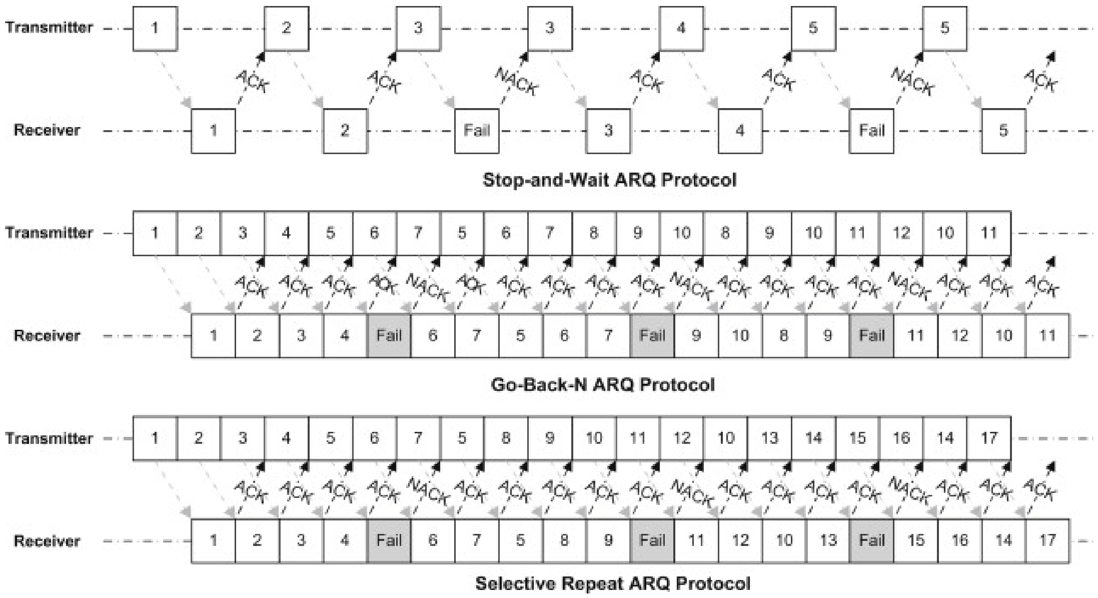


Figure 2.2: Illustration of different ARQ schemes

The ARQ variants are shown in the Figure 2.2 as an example. It demonstrates how different ARQ schemes use the communication channel, as well as the use of the ARQ buffer in the transmitter and receiver. ARQ has a significant advantage over Forward Error Correction (FEC) schemes in that error detection requires far fewer decoding mechanisms and far less redundancy than error correction.ARQ is also adaptive, in the sense that information is only re-transmitted if there are errors. FEC, on the other hand, could be preferable to error detection, either instead of or in addition to it, for the following reasons:

1. there is no input channel available or the ARQ latency is intolerable.

2. the re-transmission scheme is inconvenient to implement.

3. The predicted number of errors, if not corrected, would necessitate a significant number of re-transmissions.

For each link between the MS and the BS, the ARQ can be used. Since the use of ARQ may increase latency due to higher reliability standards, it's typically disabled for time-sensitive applications like VoIP or immersive gaming.The ARQ parameters are specified and negotiated during link setup if the ARQ mechanism is allowed. There is no combination of ARQ and non-ARQ traffic on a network. A single unidirectional flow is the domain of a single instance of ARQ.[6]

2.1.3 Hybrid-ARQ

High-rate forward error correction (FEC) and automatic repeat request (ARQ) error-control are combined in hybrid automatic repeat request (hybrid ARQ or HARQ). Using an error-detecting (ED) code like a cyclic redundancy scan, redundant bits are applied to data to be transmitted in standard ARQ (CRC). Receivers who detect a compromised message will ask the sender to send a new message.The original data is encoded with an FEC code in Hybrid ARQ, and the parity bits are either sent along with the message or are only sent upon request when a receiver detects an erroneous message. When a code, such as a Reed–Solomon code, can perform both forward error correction (FEC) and error detection, the ED code can be omitted.The FEC code is used to fix a subset of all possible errors, whereas the ARQ approach is used as a fallback to correct errors that are uncorrectable using only the redundancy sent in the initial transmission.As a result, hybrid ARQ outperforms ordinary ARQ in poor signal conditions, but at the cost of substantially lower throughput in good signal conditions in its most basic form. Simple hybrid ARQ is usually better below a signal quality cross-over point, whereas basic ARQ is better above it.

HARQ is the protocol in charge of the PHY layer's default approach for causing retransmissions. Retransmissions in HARQ are conditionally activated, as opposed to automatically repeating a message, as with the downlink and uplink repetitions mentioned above. This results in dramatically improved resource performance.When an ACK is not received in HARQ, a retransmission is triggered; however, since we strive for high reliability, the majority of packets will receive an ACK, so retransmissions will be rare. HARQ can thus be viewed as a method of starting with a higher code rate and gradually lowering it to the necessary level.For the purpose of combining retransmissions into one longer code, incremental redundancy with a circular buffer is used (as in LTE). Although this is a very resource-efficient method, it has the disadvantage of a longer latency and the reliance on receiving feedback.

A HARQ process in LTE can be

- Synchronous:The retransmissions happen after a set amount of time has passed.
- Asynchronous:Due to scheduling constraints, retransmissions can occur at any time. In this case, adequate signaling is needed to inform the transmitter of the HARQ process we are considering.

We can distinguish also between other two kinds of HARQ process:

- Adaptive:: The transmission parameters are decided on the fly by the scheduler.
- Non-adaptive:The transmission parameters remain the same as they were at the time of the first transmission attempt. Subsequent retransmissions can use progressive Redundancy Version indexes throughout this situation.

Depending on the direction of the transmission, in LTE we can state that,

- The downlink HARQ is asynchronous and adaptive.
- The uplink HARQ is synchronous and non-adaptive.

2.1.4 Forward Error Correction

FEC is a digital signal processing system that increases the bit error rate of communication links by introducing redundant information (parity bits) to the data at the transmitter side, with the redundant information being used by the receiver side to identify and correct errors inserted in the transmission link.The signal encoding that occurs at the transmitter must be properly decoded by the receiver in order to extract the original signal information, as shown in the diagram below. To prevent misinterpretation of the information by the receiver decoding the signal, the encoding rules must be specifically specified and enforced.Successful inter-operability will only take place when both the transmitter and receiver follow and implement the same encoding and decoding rules.As you can see, FEC is a crucial component that must be identified in order to allow the creation of inter-operable optical transceivers over point-to-point links. When operators support more transparent and dis-aggregated transport in high-volume short-reach applications, market developments are currently shifting toward eliminating proprietary aspects and becoming inter-operable.

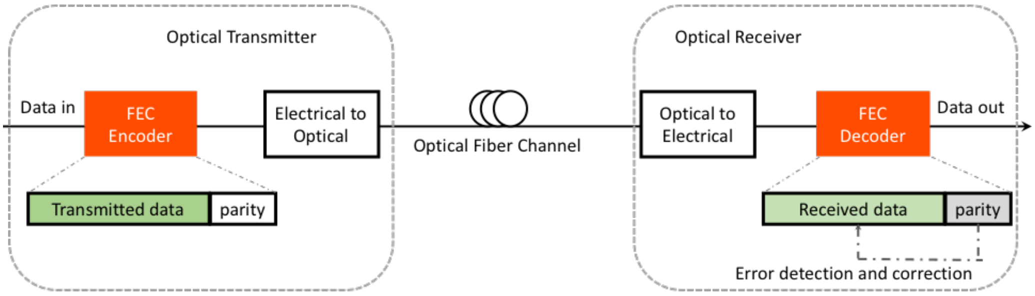


Figure 2.3: FEC working process

- Coding overhead rate is the ratio of the number of redundant bits to information bits
- Net coding gain (NCG) is the improvement of received optical sensitivity.
- Pre-FEC BER threshold is a predefined threshold for error-free post-FEC transmission determined by NCG

Hardware complexity, latency, and power usage are also variables to consider.

The option between HD-FEC and SD-FEC is a significant one in FEC coding and decoding. HD-FEC uses exact thresholds to decide whether a bit has occurred,

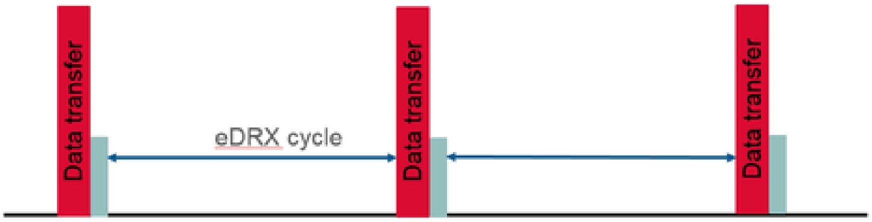


Figure 2.4: eDRX cycle

whereas SD-FEC uses probabilities to determine whether a bit has occurred. SD-FEC will have a higher NCG in order to get closer to the ideal Shannon limit, but this also means an increase in complexity and power consumption.

2.1.5 Extended Discontinuous Receprion (eDRX)

Extended Discontinuous Reception (eDRX) is an expansion of an existing LTE feature that IoT devices may use to save power. eDRX can be used independently of PSM or in combination with it to save even more fuel. Developed by mobile network standardization body 3GPP and implemented in 3GPP Rel.13, eDRX allows application developers to set, and later adjust how long an edge system stays in low-power sleep mode until it wakes up to listen for any network signals for pending data.

With eDRX, the amount of time a system is not listening to the network can be greatly expanded. It may be appropriate for a computer to be unreachable for several seconds or longer in a Massive IoT application. eDX may give a reasonable compromise between system reachability and power consumption for certain applications, but it does not provide the same levels of power reduction as PSM. When devices will sleep, networks and devices negotiate. For a set period of time, the system's receiver circuitry is turned off; the device is not listening for paging or downlink control channels during this time. The receiver will listen for the Physical Control Channel when the system wakes up.

The system will use eDRX to listen for pending data indications without requiring a complete network link. eDRX uses less power than if it made a complete network link by only listening for a pending data indication, so this step helps save the device's power. This listening process takes a fraction of the time it takes to establish a complete network connection.

Discontinuous reception (DRX) is now used by many smartphones to prolong battery life between recharges. The smartphone will save power by temporarily turning off the receive part of the radio module for a fraction of a second.The network cannot reach the smartphone if it is not listening, so if the amount of time is held to a minimum, the smartphone user will not feel any service loss. If the smartphone is called, for example, it may simply ring a fraction of a second later than if DRX is disabled.

Only the active process is affected by eDRX timers. The eDRX effect is reduced the shorter an active step is configured. eDRX is ideal for devices that must be reachable from the network for an extended period of time, or even the entire time. At this time, the device's receiver will only be available for a particular time interval (PTW), which will be repeated per eDRX (PCL) period. The mobile network knows which time periods the system will listen to pagings during because of the timers set, and will only submit a paging for this device during those times. This will also help the network (eNodeB) save money.

For devices using LTE-M LPWA networks, the maximum sleep period ranges from 43 minutes to three hours, while for devices using NB-IoT LPWA networks, the maximum sleep time ranges from 43 minutes to three hours. LTE-M can have a minimum sleep period of 320 milliseconds (ms) and NB-IoT can have a minimum sleep time of 10.24 seconds. The amount of time a system slept before waking up was previously determined by the network (e.g., with eDRX's predecessor, DRX) (typically 1.28 seconds or 2.56 seconds). With eDRX, the system, not the network, decides how long it will sleep for, a duration known as the eDRX cycle. Since a computer cannot be reached while it is asleep, the time it takes to reach it is determined by the length of the eDRX period set by the application developer.

When it comes to balancing a device's reachability against its battery consumption, the ability to set the duration of the eDRX cycle gives IoT application developers a lot more versatility.When a cloud service sends the system a command (e.g., report your position), it could take up to 82 seconds for the device to receive the command until it reports back its location if the application prefers an eDRX interval of 82 seconds. The word "mobile terminated latency" is used to describe this lag.

The application's sensitivity to mobile terminated latency is highly dependent on the application. For example, a pet tracking app may be willing to wait a few minutes to get a location fix for a pet if it means the battery would last a year rather than a week. Other applications, such as a remote light, which only require a maximum of 5 seconds for the remote battery-powered light to respond to a "ON" request, requiring the eDRX cycle to be set to 5 seconds. Another advantage of eDRX is that it is not static; the program can modify it at any time. For example, the application may set the original eDRX cycle for an hour to prolong the life of the device's battery, and then reduce the eDRX cycle to a couple of minutes later when the application needs to be more sensitive (using more battery power, but allowing the device to be reached with a much shorter delay).

eDRX is well designed for IoT application use cases where you want to save the battery power of a low-cost edge system and the device's reachability does not have to be instantaneous, ranging from 5 seconds to 6 hours. A smart gas meter monitoring IoT program is one example of such a use case. Government regulations require that the company responsible for the meter be able to immediately cut off the gas supply in a house, for example, after being notified of a fire nearby in certain smart gas meter applications. Although the company is not required to turn off the gas immediately, it must do so within a certain amount of time, such as two minutes after receiving notification from the fire department that the fire has spread to the location where the smart gas meter is located. The company will use eDRX to program the smart gas meter to request an 82-second eDRX loop. This eDRX cycle ensures that the meter will respond to a shut-off request in the allotted two minutes. In this way, the device's battery life can be increased, the device's cost can be reduced (because it requires a smaller battery), and the application's data transfer costs can be reduced (since the device only connects and transmits data when it is told to by the network). This while adhering to government regulations and assisting in ensuring the safety of the building's occupants in the event of a fire.

Another popular IoT use case for which eDRX is well equipped is "finding" IoT applications – for items such as pets, bikes, computers, expensive equipment, and similar properties. Unlike asset monitoring apps, which involve periodic, near-real-time updates, the user of these applications would typically only be interested in knowing where the asset is when it has been misplaced. Of course, when they do need to locate the asset, they'll want it quickly — 1 or 2 minutes is normally appropriate if it means the unit will have an additional year of battery life. Fido can't get into too much trouble in a few minutes, but give him an hour alone and he can get into a lot of trouble with the neighbor's cat. Using a compact, low-cost, low-power device attached to the asset, users can learn the current location of their pet, bike, laptop, tool, or other asset in a minute or two with eDRX. Furthermore, since the system would only link to the network when instructed, the application's data transmission costs would be virtually zero.

eDRX support varies between carriers. The network responds with the actual applied eDRX cycle and paging time window after the device requests a specific eDRX cycle (PTW). Enabling eDRX has no effect on the module's ability to send data, but it does switch off reception for the period interval specified, saving power. T-Mobile supports all known values for the narrowband eDRX cycle, NB-S1 mode, ranging from 20s (20.48s) to 3hrs values for the narrowband eDRX cycle, NB-S1 mode (10485.76s). Bit strings of the form: are used to represent values.

”0010”	20.48s
”0011”	40.96s
”0101”	81.92s
”1001”	163.84s
”1010”	327.68s
”1011”	655.36s
”1100”	1310.72s
”1101”	2621.44s
”1110”	5242.88s
”1111”	10485.76s

The requested eDRX period can be saved in non-volatile memory and continued between sessions, depending on which NB-IoT module your product uses. Disabling eDRX with an AT command and the option to restore the default value is common in modules.

2.1.6 Power Saving Mechanism (PSM)

Power Save Mode (PSM) and Extended Discontinuous Reception (eDRX), two of these optimization features, are listed in this guide to help you decide if they can support your Massive IoT application — and, if so, how to use them.

Huge IoT devices usually send and receive data infrequently. The unit can be put to sleep between cycles of data transmission and reception to reduce power consumption and thus optimize battery charge.

Power Save Mode (PSM) is a cellular modem feature that switches off the device's radio and puts it to sleep without forcing it to reconnect to the network when it wakes up. Although the re-attach operation consumes a small amount of energy, the cumulative energy consumption of re-attachments over the lifespan of a system may be large. As a result, if re-attachment could be prevented, battery life could be extended. PSM meets this condition. In Power Saving Mode, the UE informs the network that it will be switched off indefinitely. When the UE host system determines that it is time to transmit based on some logic or timer, it wakes up and transmits to the network, staying in RX mode for four idle frames so that it can be reached if necessary.

PSM is a device-side mechanism for conserving radio energy. The computer tells the network how often and for how long it needs to be active in order to send and receive data. The network, on the other hand, decides the final values. This mode is close to turning off the device, but the device stays connected to the network. There is no need to reconnect or re-establish PDN (Packet Data Network) connections once the system is working again. PSM was first introduced in 3GPP Release 12 and is now available for all LTE system types. The system requests PSM by providing a timer in the connect, tracking area update (TAU), or routing area update with the desired value.

If you use PSM, the system goes into a power-saving mode, which means it uses the least amount of power possible.The device remains connected to the network and retains its link settings in PSM. When a system leaves the PSM, it does not need to reconnect to the network; instead, it reestablishes the previous link, resulting in lower signaling overhead and lower device power consumption.However, since it does not listen to the paging time windows when in PSM, the system is unreachable from the network. For a mobile originated event, mobile disconnected services must be interrupted before the system reconnects to the network. Tracking Area Updates (TAU) often cause the system to stop using PSM and reconnect to the network.The system listens to paging time windows and thus queued downlink transmissions when executing a TAU. The timer T3412 specifies the maximum time between connections. The device must wake up from PSM and perform a TAU after T3412 has expired.

PSM time is the difference between two preferred timers (T3324 and T3412) given by a system when it initiates PSM with the network (T3412 minus T3324). These values may be embraced by the network, or they may be modified. The network then maintains track of the system's current state, and the device stays connected to the network. A re-attachment procedure is not necessary if a computer wakes up

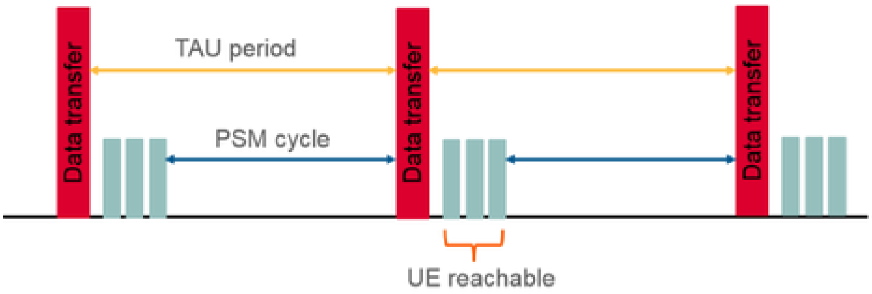


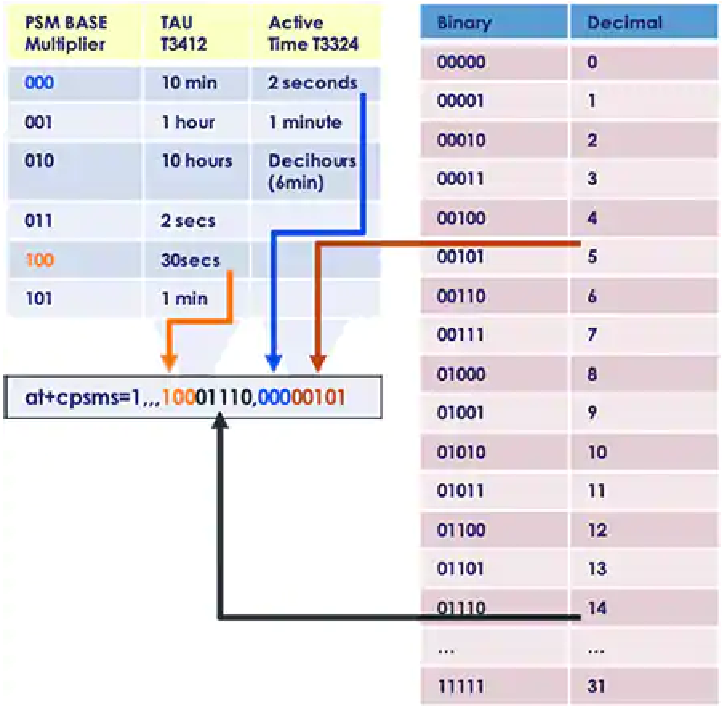
Figure 2.5: TAU (Tracking Area Updating) period and PSM cycle

and sends data before the time period agreed upon with the network has expired. From this figure we can see The PSM's value is limited by the monitoring area update that is used (TAU). The system may also request a periodic TAU during the attachment procedure by supplying a T3412 value.

For example, in a monitoring application, an application might configure a device's radio module to allow PSM, negotiate a 24-hour time interval with the network, and send a regular status update to a centralised monitoring point. If the device's monitoring program senses an alert state, regardless of any agreed-upon sleep period, the radio module will be immediately woken up and critical data will be transmitted to the centralised monitoring point without the need for a reattach procedure. However, much like a radio module that has been switched off, a radio module in PSM that is sleeping cannot be contacted by the network. Due to the inability to be contacted while sleeping, PSM may not be ideal for all applications.

An operator can choose to store incoming packets or SMS (if supported) while the device is sleeping and forward them to the device once it awakens. The MNO should set aside storage for at least the last packet of 100 bytes, according to this guide, so that the customer can send simple messages to the computer, such as a clock update. Any storage limits must be conveyed to the customer in order to achieve an understanding of the operator's store and forward strategy for UE using PSM. As the packets and SMS are stored in the home network, any restriction on downlink information retention will continue to be enforced continuously while the system is roaming.

The AT+CPSMS command was used to set the optimal periodic TAU (T3412) interval and active time for the PSM (T3324). The value must be entered in an 8-bit binary format, with the first three bits reflecting a 5-bit binary number's base multiplier. This is as defined by the 3GPP and can be found in the TS 24.008 specification.



PSM BASE Multiplier	TAU T3412	Active Time T3324
000	10 min	2 seconds
001	1 hour	1 minute
010	10 hours	Decihours (6min)
011	2 secs	
100	30secs	
101	1 min	

at+cpsms=1,,,10001110,00000101

Binary	Decimal
00000	0
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
01100	12
01101	13
01110	14
...	...
11111	31

Figure 2.6: Calculation of peridodic TAU & active time form 3GPP TS 24.008

The module will be configured to send a Monitoring Area update message to the network every 7 minutes as an example function. Periodic TAU will be set to 7 minutes or 420 seconds as a result. A multiplier of 1 Minute (101) with a value of 7 (00111) or a multiplier of 30 secs (100) with the binary value 14 (01110) can be used for 7 minutes. The Active Time operates in the same way, but with different starting values. For a 10 second active time, the values of 000 as the 2 second base multiplier and 00101, which equals 5, will result in the command being 000101, which equals 5.

at+cpsms=1,,,10001110,00000101

2.1.7 EPS Optimization

User data or SMS messages are transmitted to the IOT services via MME by encapsulating them in NAS messages in order to minimize the total number of control plane messages when handling a short data transaction.

For CIoT EPS optimization, there are two paths messages can take to the control panel:

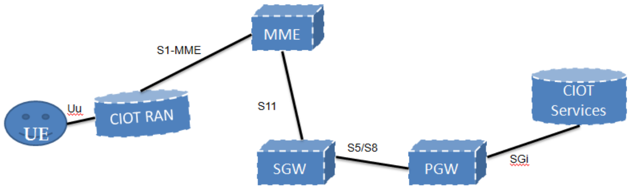


Figure 2.7: Control Panel CIoT EPS Optimization using S11 interface

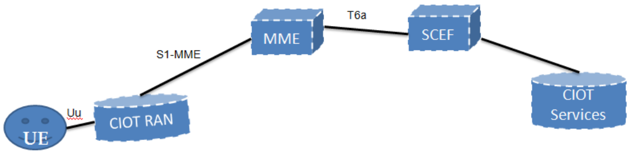


Figure 2.8: Control Panel CIoT EPS Optimization using non-IP DATA packet path

In both cases the DL data travels in the reverse direction.

This means that for small packets an RRC connection does not have to be established every time. Thus the system energy consumption and latency reduces.

For small data transmission between the MME and S-GW, a new interface named S11-U is introduced as a part of control plane CIoT EPS optimization.

The SCEF, a new node designed especially for machine type data, is used for delivery of non-IP data over control plane and also provides an abstract interface for the network services like authorization and authentication, discovery and access network capabilities.

In Control Plane CIoT optimization, the data transfer of both IP and non-IP data takes traditional paths. It is transmitted to the CIoT application providers through data radio transmitters via the SGW and the PGW. The NB-IoT network's structure is the same as the LTE network.

Data interchange between the UE and the eNB takes place at the RRC stage with the Control Plane CIoT EPS optimisation. In the RRC Connection Setup message or the complete messages for ULRDs can be piggy-backed in the DL data packets here. DL Information Transfer and UL Information Transfer can also be used for data transmission if this isn't sufficient.

All these messages contain a byte array which has NAS information, which corresponds to the NB-IoT data packets in this case. A S1-MME module exchanges dedicated Information NAS between the eNB and the MMME. The content of the received data transmission to its upper layer will thus be transmitted directly to the eNB by the UE's RRC. Security at AS stage is not applied to the data distribution process. Due to the transmission of data through NAS, there is no reconfiguration of the RRC link. This will start after or after the RRC connection setup or the resumption protocol for RRC Connection due to the end of the RRC connection following release of the RRC connection.

2.2 Improvement Opportunities

NB-IoT inherits many of its protocols from LTE, which makes for an easier and faster RnD period. But LTE was designed for UEs with a relatively longer battery life, with often more capable CPUs. LTE ensures reliability by repeating transmission and using UE resources to calculate when to retransmit. The CPU expends copious amounts of energy for these calculations, and the system becomes less energy efficient. For traditional UEs like smartphones this was a manageable loss of energy. But most IoT devices do not have the luxury of frequent charge cycles and thus need a more efficient network to prolong battery life. The most energy consumed (from UEs) is found to be during the transmission and reception.[7] [8]

$$E_{trans} = \frac{(P_{T_x} + P_{R_x}) \times M}{B \times P_{st}}$$

where,

- E_{trans} is the energy consumed, on a single wireless link, by the transceiver,
- P_{T_x} is the Transmission power,
- P_{R_x} is the reception power,
- M is the uplink packet size in bits,
- B is the transmission bit rate, and
- P_{st} is the probability of successful transmission.

Improvements were made to the system to make NB-IoT more suitable for its use-case.For the initial RRC connections phase NB-IoT sanctions a small packet in the application layer with the non-access stratum (NAS). When the UE tries an RRC connection again, it does not require a security grant every time. This significantly reduces overhead of the message, which reduces energy consumption. Known as the evolved packet system (EPS) optimization this method was introduced to reduce energy consumption and modem complexity. Whereas in LTE control signals and application data would be sent separately through the control and user planes, respectively, and application data requires a radio bearer setup each time.

Researchers have identified use of a scheduler as a potential mechanism for enhancing energy efficiency of NB-IoT networks.[9],[10] Other potential methods include repeated transmissions along with efficient bandwidth allocation

According to our research PBESM[11] is the most efficient novel scheduling method. Which proves to be more energy efficient when compared to legacy NB-IoT while also improving latency.

Another method which effectively halves the energy consumption of legacy NB-IoT is a novel channel coding method called EEACC.[7]

Chapter 3

Novel Mechanisms

3.1 Prediction Based Energy Saving Mechanism (PBESM)

3.1.1 Problems with Legacy scheduling request

The uplink packets cannot be transmitted without a scheduling request process, even when the radio bearer is identified and the RRC is in the connected state.The radio resources for packet transmission in NB-IoT networks are shared among UEs, similar to LTE networks, and an eNB scheduler dynamically assigns these resources, based on the scheduling policy, to each UE. DCI transmitted through the NPDCCH is represented by scheduling commands, which consist of assigned time, resource, and decoding information. When the connection is established, the UE uses a radio network temporary identifier (RNTI) to decode the NPDCCH information at specific times, with the decoding time configured when the connection is established. This RNTI-based decoding decides if the UE was already allocated the NPUSCH radio resource.

In the situation where the UE does not have any allocated radio services for uplink, Figure 3.1 portrays the uplink packet transmitting process, including the physical control channel. NB-IoT networks, unlike LTE networks, lack dedicated radio infrastructure for scheduling requests. As a result, only a random access procedure

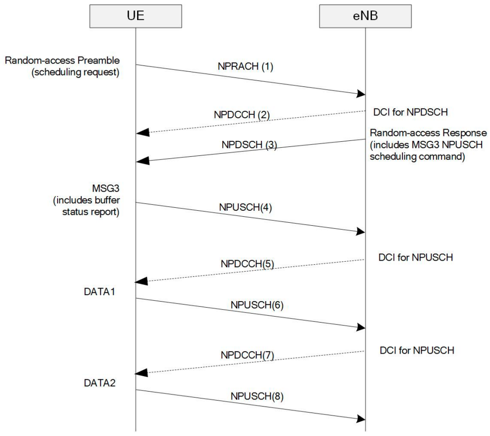


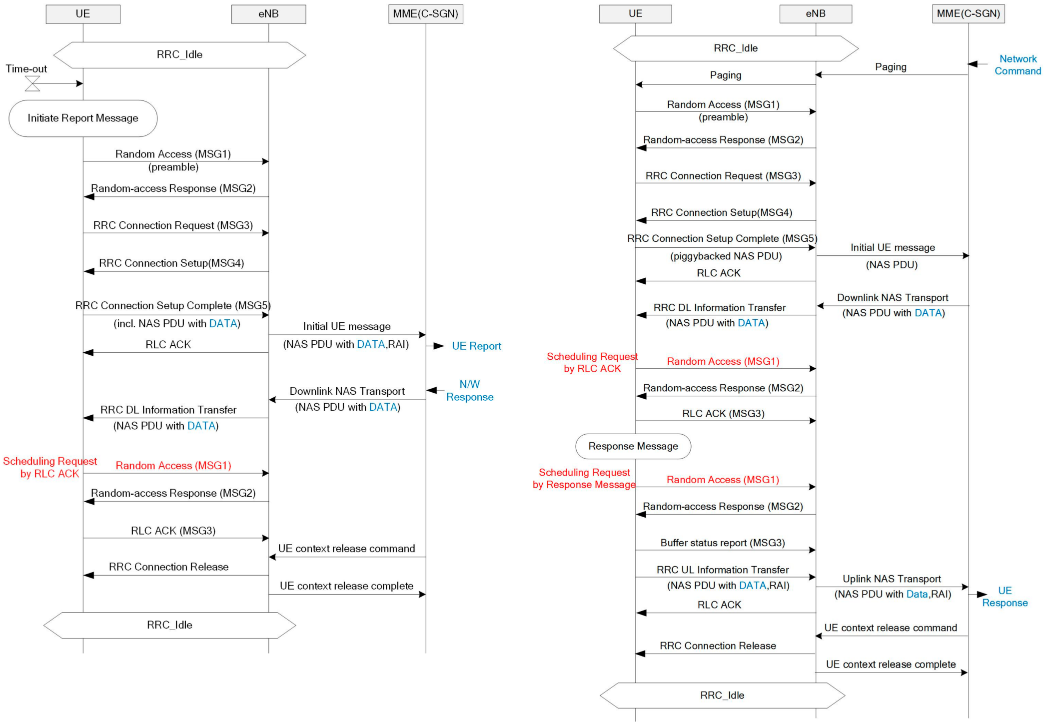
Figure 3.1: The Scheduling request procedure (with random access)

can be used to request uplink radio services. Since this protocol competes with that of other instruments, contention resolution may fail, and the UE can retry random access after a period of time has elapsed.

The preamble is transmitted through a narrowband random-access channel, as seen in phase (1) of Figure 3.1. (NPRACH). The eNB then uses the NPDSCH and NPDCCH to give the random-access answer. As seen in steps (2) and (3) of Figure 3.1, the NPDCCH contains the DCI for decoding the NPDSCH, and the NPDSCH contains an identifier for contention resolution and a scheduling command for the next NPUSCH. The sum of data in the UE buffer is unclear to the eNB scheduler. As a result, it first allocates a tiny radio resource to receive UE buffer status results. The UE then sends the buffer status report via the scheduled NPUSCH when it receives the uplink scheduling instruction, as seen in step (4) of Figure 3.1. Finally, once the data transfer corresponding to the recorded size is done, the eNB accepts the UE buffer status and begins to delegate uplink radio services. This is shown in steps (5) through (8) in figure 3.1

As a consequence, even after the radio bearer has been identified, the scheduling request protocol necessitates additional communications. Since the random-access procedure can also be used to send an RRC link request in addition to the scheduling request, the answer message is usually an 88-bit scheduling instruction, which is the appropriate size for that type of message. As a result, even small packets are difficult to transmit in MSG3 (as shown in step (4) of Figure 3.1), and another uplink transmission, similar to step (6) of Figure 3.1, is needed to complete the packet transmission. Because of the NB-IoT repeating transmission for coverage expansion, as the number of uplink application packets expands, so does the number of uplink application packets.[12]

If the UE requires a handshake with the network energy consumption increases further, as shown in Figure 3.2. As active time increases due to random access, both battery life and cell capacity reduces.



(a)

(b)

Figure 3.2: Both scenarios represent the handshake protocol. (a) Where the UE has a network response and (b) where the UE receives a network command with the network response

The energy consumption issue emerges from the transport protocol stack used in the IoT network, such as restricted application protocol (CoAP), datagram transport layer protection (DTLS), and multicast domain name scheme, in addition to the handshake caused by application packets (mDNS). Additional handshakes are required for security-context development and resuming in the DTLS protocol, which is being investigated extensively for the IoT. Given the server's memory specifications for managing various protection contexts from IoT devices, a refresh operation is unavoidable. An inaccurate timeout value in this situation will result in further scheduling request procedures and increased energy consumption.Figure 3.3 depicts the CoAP/DTLS protocol's initial 6-way handshake for defining security background. A scheduling request protocol is necessary for each transmission for uplink DTLS and RLC ACK packets.[13][14]

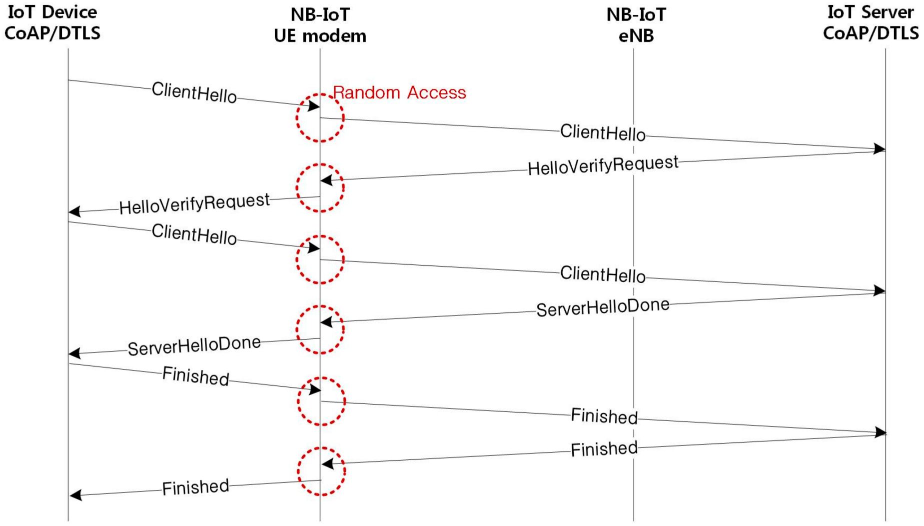


Figure 3.3: to make initial security context

3.1.2 Network Architecture for PBESM

NB-IoT network architecture with PBESM looks similar to legacy NB-IoT with the exception of two new entities. These are the packet inspection entity (PIE) and packet prediction entity (PPE). This new architecture is represented in the figure below:

The PIE is logically located on the MME. It inspects the packet header and uses it to determine the session type e.g protocol type, IP address and port number. The PIE then estimates the occurrence of the uplink return message using the same technique as the PSM's paging wait.

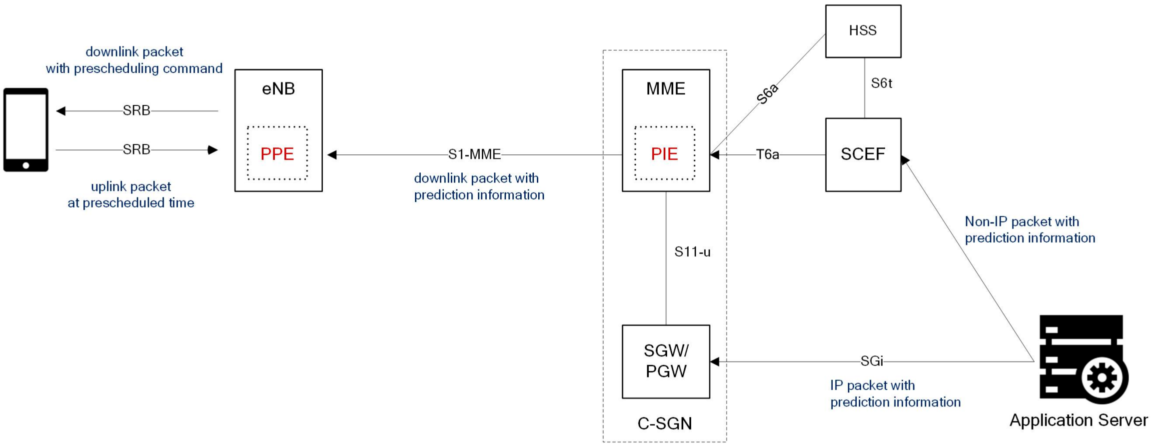


Figure 3.4: Network architecture with PBESM

Furthermore, it estimates the time taken to respond for each session and transmits this data to the PPE in the base station. The PIE is on the MME because the PSM's paging delay will allow the predictive processing delay to head in the opposite direction, which the MME can avoid. The projected data is transmitted in downlink packets over the current S1-MME protocol. Though the NAS response message and transport protocol can be predicted in the PIE, message security and customized protocols make it difficult to predict the application response. As a consequence, the application server can provide assistance.To take advantage of the PBESM, the application server must support the estimation of uplink events. Because of the paging delay in the PSM, the knowledge about packet occurrence from the application server is only reliable for prediction.

For each session, the PPE organizes the collected uplink incident information and response time. It estimates the processing delay, which is the time gap between downlink transmission and uplink packet generation, using these data and the proposed algorithm. The eNB then produces a pre-scheduling order containing the uplink transmission time and modulation scheme, which is sent to the UE via the NPDSCH in a downlink packet. When the UE receives the instruction, it retains the uplink packet for the specified period of time without the need for a scheduling order.

RLC ACK, RRC and NAS signaling messages, transport layer response (e.g.,

Packet Type	**Prediction Entity**	**Strategy**
RLC protocol	PPE	Prediction of RLC ACK packet through poll bit transmission
RRC protocol	PPE	Prediction of RRC response message through a REQUEST message type
NAS protocol	PIE	Prediction of NAS response message through a REQUEST mesage type
Transport layer protocol	PIE	Prediction for DNS, DTLS, etc. through packet header inspection
Application response	Application server, PIE	Application server can analyze application message type

Table 3.1: Predictable packet types

TCP ACK and DTLS response), and application response are the five types of predictable packets. For each packet type, Table 1 lists the prediction entities and strategies.

3.2 Energy-Efficient and Adaptive Channel Coding (EEACC)

The goal of the EEACC approach is an appropriate link adaptation scheme for NB-IoT systems that's also integrated with a proper repetition number and MCS selection mechanism in order to achieve higher energy efficiency, transmission distance, and throughput while retaining strong transmission reliability

3.2.1 Problems with Legacy Channel Coding Methods

Error correction is one of the most critical aspects of NB-IoT system design. The channel coding technique for NB-IoT, if well developed, will help save a large amount of energy by reducing the number of necessary retransmissions. This is why many research studies have suggested channel coding strategies as a way to save energy. The NB-IoT uplink baseband processing can currently be separated into two parts: channel coding and modulation. In the case of the NB-IoT uplink, channel coding involves generation and attachment of Cyclic Redundancy Checks (CRCs), turbo or convolutional coding, and rate matching.

In any mode of wireless communication, including the NB-IoT, reliability is a key performance criterion. As a result, any NB-IoT architecture must ensure that end-to-end connectivity between terminals is reliable. The effort to improve one output always comes at an expense in terms of the other, as in most engineering designs. Most communication networks use channel coding to ensure resistance to channel impairments and secure connection transmissions. To ensure durability, previous research has suggested different channel coding approaches.

The standard repetition approach and the Narrowband Link Adaptation (NBLA) approach [15] are the two major approaches that have been established. In their efforts to improve information efficiency, the methods have been found to have high energy costs. This is the subject of the current research project. It aims to strike a balance between maintaining energy quality and ensuring stable connectivity on NB-IoT uplinks.

3.2.2 Proposed channel coding method, EEACC

The latest study proposes a channel coding which is a 2-dimensional (2D) link adaptation approach, which evaluates to a dual-objective optimization problem that aims to improve the NB-IoT network coverage and transmission latency without sacrificing its energy efficiency performance. There are two aspects to the proposed adaptive channel coding technique. It consists of an inner loop and an outer loop adaptation scheme, both of which aim to improve the network's energy efficiency and throughput. The inner loop adaptation scheme, in particular, is designed based on channel conditions to ensure transmission reliability and, as a result, increase the network's data rate. The outer loop scheme, on the other hand, is based on the Modulation Coding Scheme (MCS) number and also the transmission repetition number.

The proposed approach introduces an inner loop link adaptation procedure that focuses on dynamically changing the transmission repetition number based on channel conditions (periodically sampled and updated) in the Presence of Random Phase Noise in NB-IoT Systems proposed by [16], the current BLER performance is used to predict the channel conditions each time on the next transmission. Although the coherent-time of the fading channel is assumed to be fairly long due to the assumed low mobility of NB-IoT user-equipments, the main consideration in this channel estimation model is that (UEs). As a mechanism to improve the accuracy of the approach, phase noises are taken into account before combining the channel estimates over repetition.

The time-domain baseband received signal at the nth sampling time of the lth orthogonal-frequency-division-multiplexing (OFDM) symbol can be expressed as with phase noise $l[n]$ caused by oscillator fluctuations and a residual FO f_e normalized by the sub-carrier frequency. This is expressed below:

$$S_{l(Estimated)} = \epsilon^{j\phi|n|}(\frac{1}{\sqrt{N}}\sum_{k=-\frac{N}{2}}^{(N/2)-1} S_1[k]\epsilon^{j2\pi n(f_e+kN)}) * h_1[n] + w(n)$$

where,

- “*” denotes the linear convolution
- $S_l[k]$ is the transmit symbol on the lth OFDM symbol and the kth sub-carrier,
- $h[n]$ is discrete fading channel taps,
- $w(n)$ is additive-white Gaussian-noise (AWGN) and
- N is the Fast-Fourier-Transform (FFT) size

The inner loop link adaptation’s purpose is to ensure that the transmission BLER achieves the target. The former, which includes MCS level selection and repetition number determination, is referred to as outer loop link adaptation. The following is a description of the proposed link-adaptation method.

Chapter 4

Methodology

4.1 Prediction Based Power Saving Mechanism (PBESM)

In this chapter we will discuss how pbesm works. Previously described network architecture, describes two new entities, PIE and PPE. In this chapter we will elaborate on the usage of these new entities from a network point of view. And consequently explain how the inner algorithms are designed to work.

4.1.1 Pseudo Algorithm

As described in chapter 3.1.2 the PIE, located in the MME, determines the session type by using the information in the message header. This session type is used to determine the paging delay during PSM.The session type is sent to the PPE, located in the eNB, which decides paging delay and sends it to the UE.By reducing the time for which the UE waits in idle mode (during PSM), PBESM aims to dramatically reduce energy consumption.

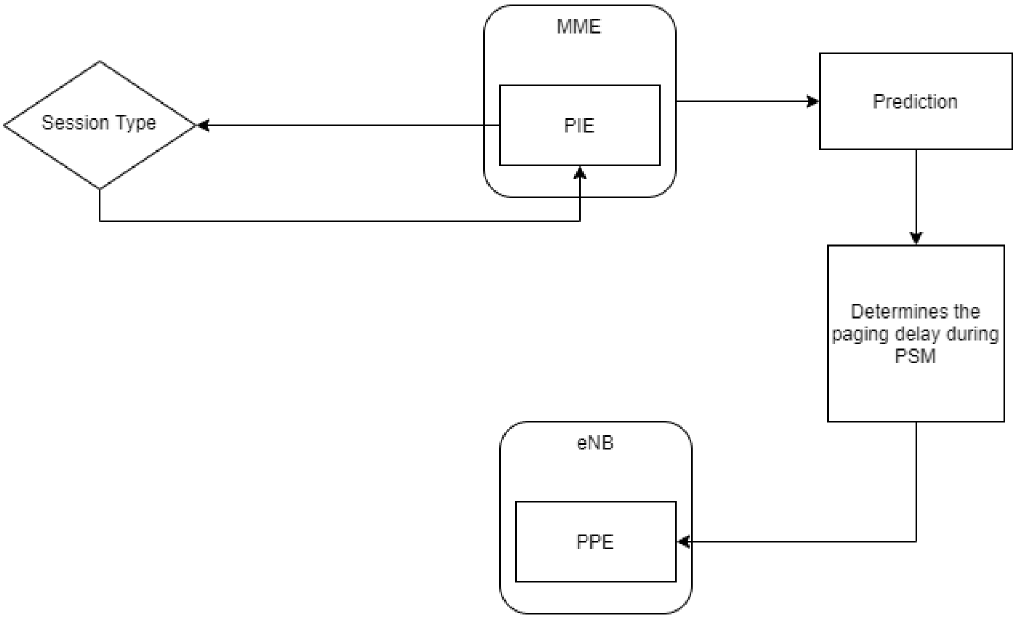


Figure 4.1: PBESM flow chart

Proposed Prediction Algorithm

As described in the flowchart below the PIE uses the header information to predict a processing delay, $T_{CID,PID}$. It is uniquely calculated for every individual packet (PID = Packet ID) per cell (CID = Cell ID). Thus for the k_{th} session the processing delay would be $T_{CID,PID}(k)$, i.e. for the k_{th} UE & Packet. Information from the MME is also factored into this calculation to get $T'_{CID,PID}$ defined as:

$$T'_{CID,PID}(k) = \alpha T_{CID,PID}(k) + (1 - \alpha) \times T_{info}$$

where,

- α is used to factor in T_{info} and
- T_{info} is the delay calculated with the information from the MME

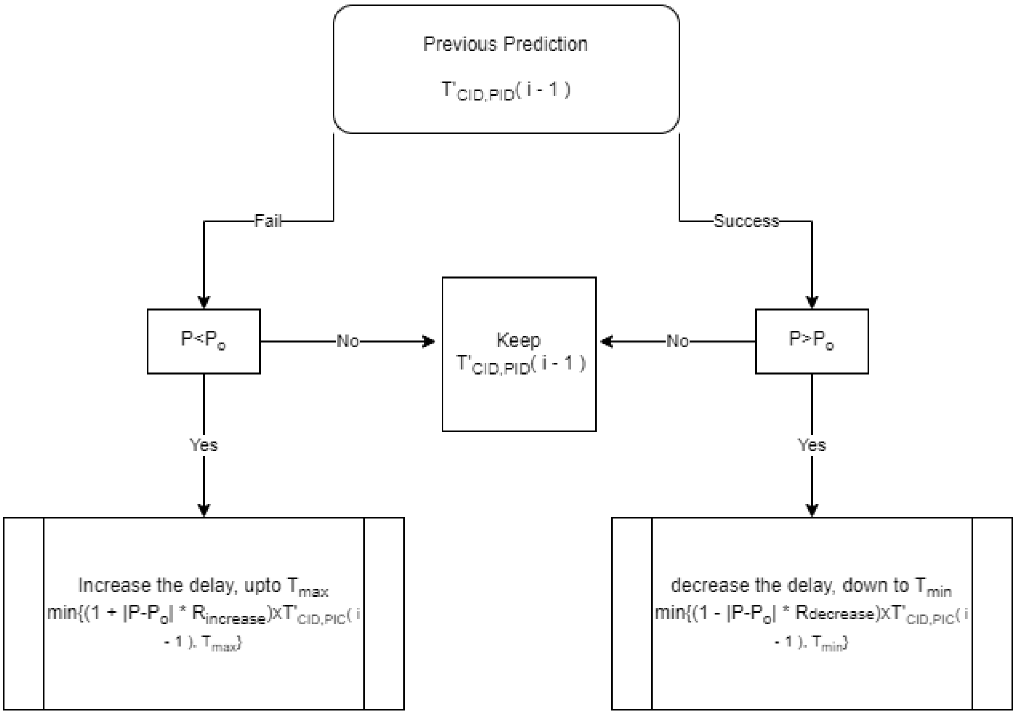


Figure 4.2: PBESM Algorithm

UE side implementation

The UE side implementation is simple, where it checks if there is any new scheduling information or not. If there is none, the UE reverts to using legacy scheduling methods.

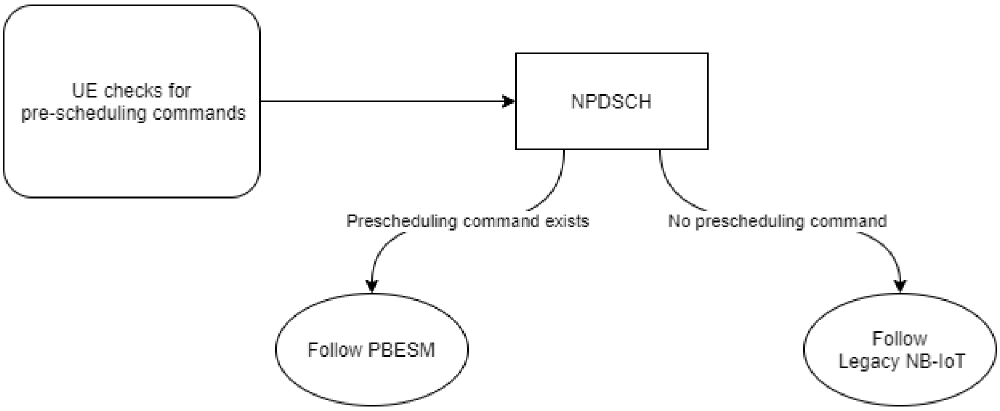


Figure 4.3: PBESM in the UE side of things

4.1.2 Experimental Setup

PBESM was tested against Legacy NB-IoT in Five different scenarios. These scenarios are described in the table below:

Scenario	**1**	**2**	**3**	**4**	**5**
UE report	x				
Network connam		x	x	x	x
Non IP	x	x			
IP			x	x	x
CoAP/DLTS/UDP			x	x	
Initial security context Creation			x		
Resuming security context				x	
TCP					x

Table 4.1: Different simulation scenarios to test PBESM

In the following chapters we will discuss the results obtained when PBESM is compared with scheduling methods of Legacy NB-IoT, in the context of these five scenarios.

4.2 Energy-Efficient and Adaptive Channel Coding Method

For a specific Period, T, The ACK/NACK transmissions are monitored. Using these transmissions MCS level is either increased or decreased, and BLER is calculated. The channel condition is then catagorized according to the BLER value. A Bler of 0% to 7% is a good channel, between 7% and 13% is considered to be medium and a BLER above 13% would mean the channel condition is bad. If the channel is good then the number of re-transmissions per NACK is reduced, and if it is bad then re-transmissions are increased. To make sure that the MCS level does not change too frequently, a compensation delay ΔC is used. This targets a BLER of 10% which is in accord with 3gpp standards.The working principle of both the inner and outer loop is described in the flow chart below.

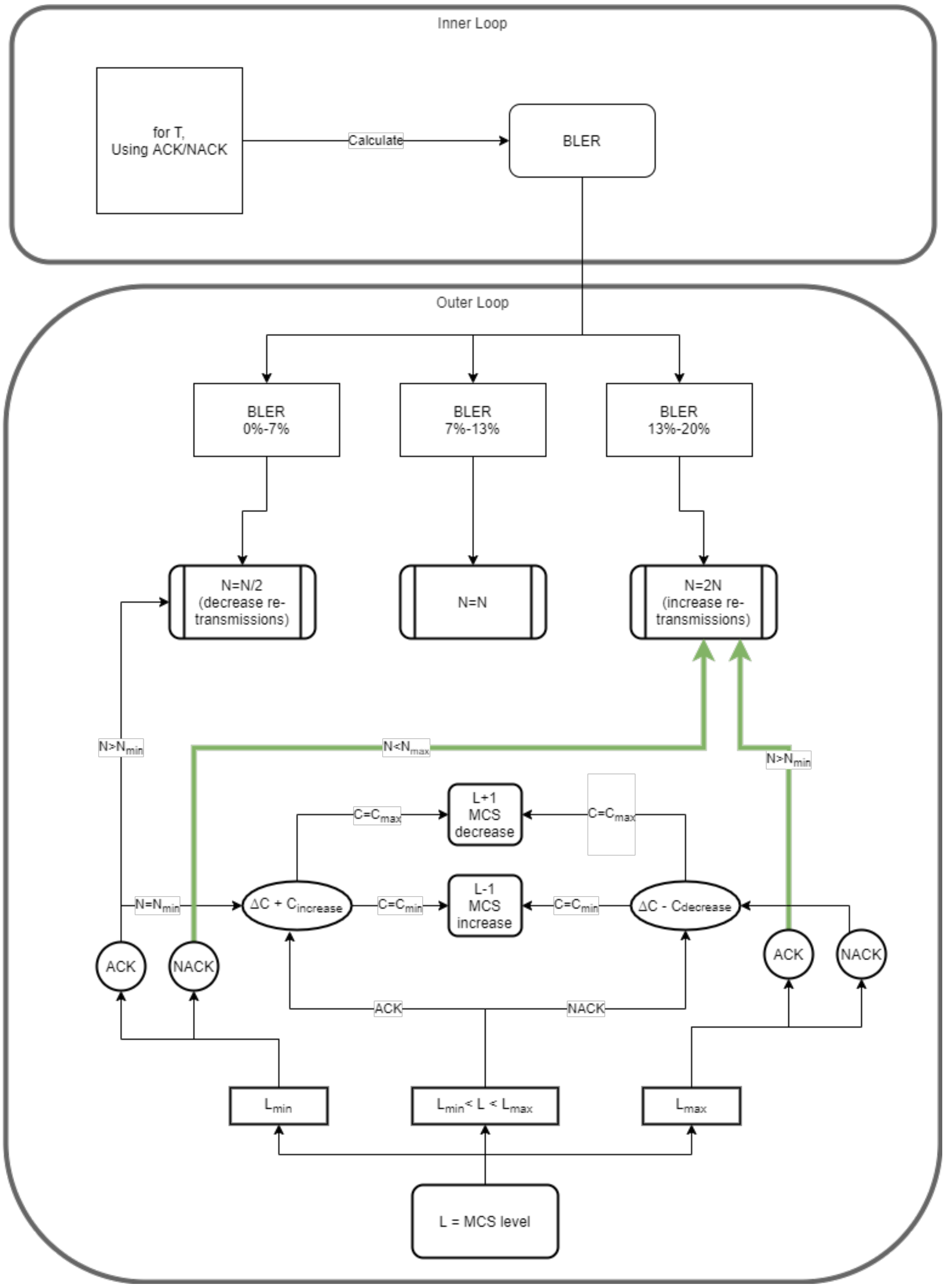


Figure 4.4: A flowchart illustrating the pseudo algorithm for EEACC

Chapter 5

Analysis and Results

NB-Iot is a new and rising technology. As such, most of research conducted in the field focuses on improving the legacy methods, by tweaking the parameters and trying to find an optimal value for said parameters. However novel mechanisms have been proposed to replace old ones, which are not exactly designed with IoT devices as primary targets.

The most crucial issue with old mechanisms in legacy NB-IoT is to reduce energy consumption while also maintaining latency. Although NB-IoT introduces the EPS mechanism to contort itself to IoT needs, it leaves a lot more to desire.

The following figure 5.1 describes how we compare the various novel mechanisms, proposed to convert NB-IoT into a more suitable network for IoT devices, with legacy mechanisms. To finally design an NB-IoT network which not only reduces energy consumption of the UEs in the network, but also reduces latency.

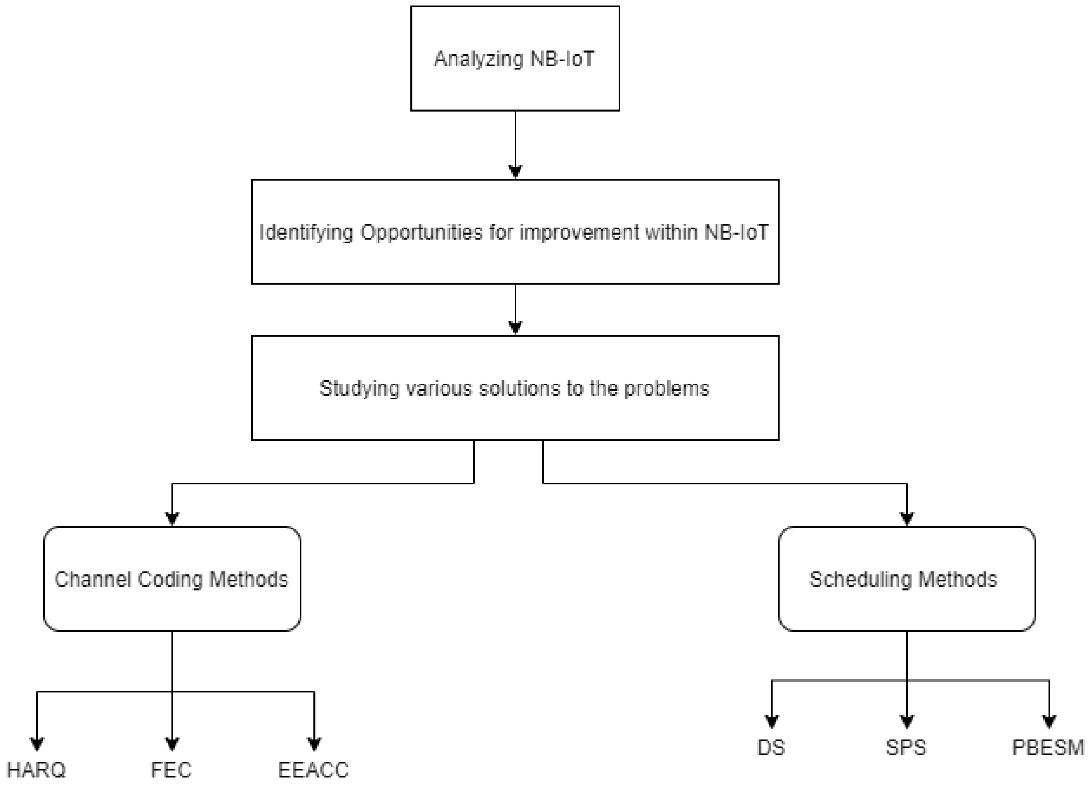


Figure 5.1: Comparison and analysis approach to choose optimal NB-IoT

These comparisons are further described in the tables below. Table 5.1 is a comparison between legacy channel coding methods and EEACC and table 5.2 compares legacy scheduling methods with PBESM.

	ARQ	**FEC**	**HARQ**	**EEACC**
How it works	Uses ACK/NACK to repeat transmission if needed.	A preamble used to correct errors after reception, is added to the message.	Uses ACK/NACK to determine when to send a preamble with FEC code.	Predicts channel condition using ACK/NACK. Then assigns appropriate MCS level and retransmissions accordingly
Pros	Ensures reliability and the message doesn't contain redundant data for channel coding.	A fewer number of retransmissions are required to ensure reliability	Type I: Re-transmissions occur when error correction fails. Type II: Only messages that fail to properly transmit are retransmitted with an FEC preamble	Messages do not contain any redundant data for channel coding and number of re-transmissions are also decreased.
Cons	A lot of re-transmissions are needed to ensure reliability. This increases energy consumption.	Every message has a preamble with error correction data. This decreases the datarate, and effectively increases energy consumption for a given datarate.	Energy consumption is not dramatically reduced. As both Type I and Type II implementations have similar problems to FEC and ARQ.	It is impossible to know how the network will behave practically. As this is a novel mechanism and hasn't been implemented yet.

Table 5.1: Comparison between Legacy Channel Coding Methods and EEACC

	Dynamic Scheduling	Semi-Persistent Scheduling	PBESM
How it works	Using the ACK/NACK transmissions a dynamic schedule is created for the UE. This schedule is updated and sent with every message.	A predetermined schedule is sent to the UE. To allow the UE to transmit and receive during the time a dynamic schedule is made. The rest of the process is similar to DS.	The initial Uplink transmissions are used to determine an optimal schedule for the UE. Which is only sent once within a certain time frame.
Pros	Works ideally when UE transmissions are sporadic.	Works ideally when UE transmission are periodic.	Theoretically performs well in all types of situations.
Cons	When UE transmissions are periodic, wasting bandwidth to send scheduling information increases energy consumption.	If the UE transmissions are sporadic it takes time to the new scheduling requirements. This means that the UE does not get to transmit or receive when it wants to. Thus wasting energy and time, reducing latency.	It is impossible to know how the novel network will behave in a practical scenario. As it has not been implemented yet.

Table 5.2: Comparison between Legacy Scheduling Methods and PBESM

As evident from the analysis above EEACC & PBESM out perform Legacy channel coding methods and Scheduling methods respectively. And are a clear choice for optimizing NB-IoT.

Bibliography

[1] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.

[4] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[5] E. Dahlman, S. Parkvall, and J. Sköld, "Chapter 9 - scheduling and rate adaptation," in *4G LTE-Advanced Pro and The Road to 5G (Third Edition)* (E. Dahlman, S. Parkvall, and J. Sköld, eds.), pp. 249–267, Academic Press, third edition ed., 2016.

[6] S. Ahmadi, "Chapter 7 - the ieee 802.16m medium access control common part sub-layer (part ii)," in *Mobile WiMAX* (S. Ahmadi, ed.), pp. 281–319, Oxford: Academic Press, 2011.

[7] E. Migabo, K. Djouani, and A. Kurien, “An energy-efficient and adaptive channel coding approach for narrowband internet of things (nb-iot) systems,” *Sensors*, vol. 20, no. 12, 2020.

[8] M. E. Migabo, K. Djouani, A. M. Kurien, and T. O. Olwal, “A stochastic energy consumption model for wireless sensor networks using gbr techniques,” in *AFRICON 2015*, pp. 1–5, 2015.

[9] R. Boisguene, S. Tseng, C. Huang, and P. Lin, “A survey on nb-iot downlink scheduling: Issues and potential solutions,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 547–551, 2017.

[10] B.-Z. Hsieh, Y.-H. Chao, R. Cheng, and N. Nikaein, “Design of a ue-specific uplink scheduler for narrowband internet-of-things (nb-iot) systems,” pp. 1–5, 04 2018.

[11] J. Lee and J. Lee, “Prediction-based energy saving mechanism in 3gpp nb-iot networks,” *Sensors*, vol. 17, no. 9, 2017.

[12] 3gpp, “Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran),” 2013.

[13] P. Y.Sheffer and R. Holz, “Summarizing known attacks on transport layer security (tls) and datagram tls (dtls),” *Technische Universitaet Muenchen*, 2015.

[14] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, “Security of the internet of things: perspectives and challenges,” *Wireless Networks*, vol. 20, no. 8, p. 2481–2501, 2014.

[15] C. Yu, L. Yu, Y. Wu, Y. He, and Q. Lu, “Uplink scheduling and link adaptation for narrowband internet of things systems,” *IEEE Access*, vol. 5, pp. 1724–1734, 2017.

[16] F. Rusek and S. Hu, “Sequential channel estimation in the presence of random phase noise in nb-iot systems,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, 2017.