Tampere University

Justin Njimgou Zeyeum

EFFECTS OF TIME SYNCHRONIZATION ERRORS IN IOT NETWORKS

Electrical Engineering
Master of Science Thesis
June 2019

ABSTRACT

Justin Njimgou Zeyeum: Effects of Time Synchronization Errors in IoT Networks

Master of Science Thesis
Tampere University
Electrical Engineering
June 2019

Internet of Things is a term referring to the wireless connection of people and devices, briefly referred to as ‘things’. The growth of technology has become so rapid, that people are finding various ways and means to communicate to each other in a fast and reliable way. Industries and other organizations such as hospitals, military, schools and so on, are demanding better, easy and cheaper way to communicate or pass out information.

Time and frequency synchronization are basic demands for all wireless communication system to work accurately. In time synchronization, the receiver terminal determines the correct time at which to sample the incoming signal. For two or more systems to function at same time with high speed, accuracy and reliability, they must be well synchronized, and time sensitive enough so that it will not experience failure at some point in time.

This thesis focuses on the characteristics of IoT technologies, how time-sensitive an IoT network can be, and what time and frequency synchronization solutions there exist. A simulation study is also performed using Binary Phase Shift Keying (BPSK) modulation and Narrowband (NB) and Ultra-Narrowband (UNB) signals.

The simulation-based analysis is done with three error models (constant, random and clock) using MATLAB simulation, where a plot of Bit-Error-Rate (BER) versus Signal-to-Noise-Ratio (SNR) is drawn to investigate the effects of the time synchronization errors with the NB and UNB signals.

Keywords: IoT, Time-sensitivity, Clock Error, Constant Error, Random Error, Time Synchronization, NB, UNB.

The originality of this thesis has been checked using the Turnitin Originality Check service.

PREFACE

This thesis was carried out in the department of Electrical Engineering at Tampere University with the supervision of Assoc. Prof. Elena-Simona Lohan and Dr. Jukka Talvitie for the partial fulfilment of my Master of Science degree in Technology majoring in Wireless Communications.

It was a great pleasure working under my supervisors who at their best guided me through this project, by providing me with materials, study links and quality comments to enable me to complete this work. I want to say a big thank you to you both.

I will also like to thank my beloved parents Mr. Rene Njimgou and Mrs. Theresia Masah (of blessed memory) who both passed away in a space of a month during my study, for their constant support and advice given to me since I was a kid, until I became a father. They will always remain in our hearts. May their souls rest in peace.

Special thanks go to my lovely wife Mrs. Colette Nyonga Zeyeum and gorgeous daughters Chloe Masah Zeyeum and Carissa Partem Zeyeum, for being there with me all through my study.

I will also use this opportunity to thank all my lecturers and course mates for the cordial relationship we shared and their support during my studies.

Finally, I thank God Almighty for his infinite love and mercy for my life and that of my family. May his goodness, love and mercy endure forever.

Tampere, 19 June 2019

Justin Njimgou Zeyeum

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AP	Access Point
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
BS	Base Station
CSI	Channel State Information
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
DL	Downlink
DoA	Direction of Arrival
DSSS	Direct-Sequence Spread Spectrum
DBPSK	Differential BPSK
EC-GSM-IoT	Extended Coverage GSM IoT
EGPRS	Enhanced GPRS
EKF	Extended Kalman Filter
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FHSS	Frequency-Hopping Spread Spectrum
GMSK	Gaussian Minimum Shift Keying
GSM	Global System for Mobile Communication
GPS	Global Positioning System
GPRS	General Packet Radio Service
HARQ	Hybrid Automatic Repeat Request
ICI	Inter-Carrier Interference
IEEE	Institute of Electrical and Electronic Engineering
IoT	Internet of Things
ISI	Inter-Symbol Interference
IT	Information Technology
LO	Local Oscillator
LTE	Long Term Evolution
LPWAN	Low Power Wide Area Network
MCL	Maximum Coupling Loss
M2M	Machine-to-machine communication
MTC	Machine Type communication
NB-IoT	Narrow Band-Internet of Things
NCO	Numerically Controlled Oscillator
OFDM	Orthogonal Frequency Division Multiplexing

OFDMA	Orthogonal Frequency Division Multiple Access O-QPSK
PRB	Physical Resource Block
PLL	Phase-Locked Loop
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAN	Random-Access Network
RSS	Received Signal Strength
RTT	Round Trip Time
SDR	Software Defined Radios
SIG	Special Interest Group
SNR	Signal-to-Noise Ratio
TB	Terabytes
TDMA	Time Division Multiple Access
TDD	Time Division Duplex
ToA	Time of Arrival
TSN	Time-Sensitive Network
TCP	Transmission Control Protocol
UL	Uplink
UNB	Ultra-Narrowband
UTC	Universal Time Coordinated
VCO	Voltage Controlled Oscillator
Wi-Fi	Wireless Fidelity
WIoTF	Wireless Internet of Things Forum
WSN	Wireless Sensor Network
2G	2th Generation
3G	3th Generation
3GPP	3th Generation Partnership Project
4 G	4th Generation
5G	5th Generation partnership program
σ	Sigma (standard deviation)
β	Beta
μ	Mu (mean)
N	Normal distribution
$\rho(k)$	Clock offset at time k
$\alpha(k)$	Skew at time k
Δt	Time interval
$v(k)$	Driving noise of clock skew
k_i	Frequency drift
$\emptyset$	Scaling factor

C	Speed of light
LO_{Tx}	Transmitter Local oscillator frequency
LO_{Rx}	Receiver Local oscillator frequency
L_i	Data Loss Tolerance
Δf	Carrier frequency offset
Δ	Sampling clock offset
S_{min}	Minimum sample data
S_{max}	Maximum sample data
S_i	Time-sensitivity
W_i	Weight Computation

.

1. INTRODUCTION

Before the emergence of IoT, cellular technologies have adhered to an approximate 20-year cycle from launch to peak penetration, and around ten years between the launch of each new technology. Mobile communication technologies are typically categorized into generations; 1G is an analog mobile radio system of the 1980s (e.g. NMT), 2G is the first digital mobile systems (e.g. GSM), 3G is the first mobile systems handling broadband data (e.g. WCDMA), 4G “enhanced broadband”, Long-Term Evolution (LTE), 5G which is even further enhanced broadband and IoT which functions in connecting things together in a more reliable and fast way.

With these technological trend, the current demand for basic commodities (such as electricity, internet, food, water, healthcare, transport, information, to name but a few) in the world today, that is rising day by day, also the population growth, climate change, and the increase usage of science and technology in every aspect of life, is going to be solve. Some part of the world is already benefiting from 5G and IoT services and there is ongoing research to make connecting with people and things easier and faster.

IoT is very broad but addressing and investigating how it signals behave in certain conditions (such as errors and noise) is essential in the feature development. It signals such as Ultra-Narrowband (UNB) refers to technologies with less than 1kHz bandwidth, that is to achieve ultra-high range. It has high bandwidth efficiency, which uses less portion of the spectrum with continuous wave signals. Narrowband (NB) on the order hand uses a narrow set of frequency that is greater than that of UNB, that is a technology with greater than 1kHz bandwidth, it uses more bandwidth compared to UNB.

In this thesis, we investigate the behaviour of these two signals under same noise and possible error conditions, thereby plotting their bits error rate (BER) versus signal-to-noise ratio (SNR) curve.

1.1 Thesis objectives

This thesis is for the partial fulfilment of Master of Science degree in Electrical Engineering in the major of Wireless Communications. The thesis studies the effect of time synchronization errors in NB and UNB Binary Phase Shift Keying (BPSK) IoT technology. The motivation for this work is to find out which error model have minimal time synchronization effect and hence smaller effect on the performance of an IoT receiver. This is done by introducing the errors into the receiver system to see the impact it will cost when plotting the BER curve with respect to SNR.

The work is carried out by measuring the effect of time synchronization error on the NB-BPSK and UNB-BPSK using constant error model, random error model and finally clock error model. These models are compared by taking the BER versus SNR graph. The percentage error for each of the simulation is calculated and synchronization time recorded.

1.2 Author's contributions

The Author's contributions for this thesis are as follows;

- Addressing the research question about IoT synchronization errors to the best of Author's knowledge
- Searching relevant literature and information in journals, lectures and online materials
- Carrying out the simulations on a simulator initially provided by the supervisors and then developed by the Author
- Analysing the simulation results
- Writing the manuscript, and addressing the comments raised by the thesis supervisors.

1.3 Thesis structure

The thesis is structured into 7 chapters. In chapter 2, an overview of IoT Technologies is given. The author discusses their performance, latencies, spectra and whether they support synchronization or not. Chapter 3 describes what time-sensitivity is, its usual performance metrics and what time-sensitive IoT networks are. Chapter 4 describes reasons why we need synchronization and five different time and frequency synchronization solutions in IoT network. Chapter 5 describes the clock and frequency synchronization error models. Chapter 6 describes the system models, simulation methods, and results. Chapter 7 summarizes the thesis work and discusses further research work.

1.4 Research methodology

Different stages of literature review were carried during the thesis writing, as explained below,

In the first step, the following research questions were addressed

1) What are the synchronization requirements in an IoT network?
2) What is a time-sensitive IoT network? How do you define time-sensitivity? what metrics are used in time sensitivity?
3) What are some typical time and frequency synchronization solutions in IoT?

In the second step, the following research questions were addressed

1) What IoT systems to use in the simulations? (e.g. NB-IoT, Sigfox, Lora, etc.)
2) What clock models to use in the simulations?
3) How many nodes to simulate?
4) What metric to use in order to measure the receiver performance with clock errors?

The third step consisted in the implementation of a (Matlab-based) simulator for BPSK modulation.

The fourth step included the following

1) Comparison of BER with perfect synchronization versus imperfect synchronization.
2) Change of BPSK signal to a BPSK ultra narrowband (UNB) pulse similar to ones used in IoT (e.g. Telensa, Sigfox) and rechecking the performance.

The fifth methodological step consisted in introducing the errors models, namely constant error, random error and clock error. After about 4 attempts before the results were accepted.

Different media and supporting material were used were used to research on these, such as Tampere University lecture notes on wireless communication and previous theses at Tampere University, ResearchGates portal, IEEE website, ArXiv.org, Matlab exchange, GitHub, Zenodo.org, Code Ocean and Google searches.

2. IOT NETWORKS OVERVIEW

The emergence of internet of things and 5G mobile communication will and is already helping in achieving the needs of communication with low latency and reliability. It is said that all devices that benefit from an internet connection will be connected in the future, that is every person, industry will be empowered, by this IoT technology. This means that the future technology will deliver machine-to-machine(M2M) and machine-to-person communications in a massive manner [1]. Ericsson also predicts that there will be around 28 billion connected devices by 2021, where more than 15 billion will be connected by M2M and consumer-electronics devices [2].

The principle of IoT or 5G technology cannot be achieve without a specific wireless technology that will be using to deploy its connectivity, that is why many organizations have developed several wireless technologies to address the various segments of IoT connectivity. Some of the IoT technologies are short range and while some are long range. These technologies will be described here briefly.

2.1 Sigfox

Being one of the world’s leading IoT service providers with Low Power Wide Area network connectivity, Sigfox is drastically bringing down cost and energy consumption required for securely connecting IoT sensors to the cloud, where there will be no need for replacing or re-charging batteries since the devices will generate energy themselves [3]. It is compatible with Bluetooth, GPS 2G/3G/4G and Wi-Fi, offering services to about 50 countries and regions, with 4.2 Million km^2 covered and 949 million people [3].

Sigfox is an ultra-narrow band (UNB) technology that operates in the 868MHz frequency band in Europe and 915MHz in the US, it has a Physical rate of 100b/s for a 100Hz bandwidth, 1000b/s for a 1KHz channel width in Europe (and 600b/s in the US), with sensitivity of -140dBm, with a range of 40km [4]. Sigfox does not need synchronization since its technology is asynchronous. Some of its use cases are water meter, Gas meter and Electricity meter. Table 1 below shows the summary of its features.

Table 1. Sigfox features.

Specifications	**Sigfox**
Frequency Band	868MHz in Europe and 915MHz in US
Transmission Bandwidth	100Hz and 1 kHz in Europe
Data rates	100b/s and 1000b/s, 600b/s in US

Range (2-Rays)	~11km
Data size	12 bytes
Modulation Scheme	D-BPSK and GFSK
Spectrum	UNB
Applications	IoT and M2M based applications
Receiver sensitivity	-140 dBm
Bandwidth	200kHz
Time synchronization	Not supported
Latency	20s between DL and 1st UL message

2.2 LoRa/LoRaWAN

LoRa Alliance is an open standard non-profit association of more than 500-member companies, operating in 51 countries with 100 deployed LoRaWAN, who are committed to enable large scale deployment of Low Power Wide Area Networks (LPWAN) IoT via the deployment and promotion of LoRaWAN [5].

LoRa is a long-range IoT application that has a cellular topology with base stations/gateways which receive packets from devices and relay the data to a server on a TCP connection, operating in the same frequency band as Sigfox and 802.11ah [4]. Its capacity to provide wide area network service makes it to be referred as LoRaWAN, the network is composed of various elements including the endpoints, LoRa gateways, server, and a remote computer [6].

Unlike Sigfox which does not support synchronization, LoRa supports synchronization particularly in time domain and is a spread spectrum technology with wider frequency bands. Summary of the features are shown in the table 2 below. Sigfox and LoRa networks have been properly studied in [6].

Table 2. LoRa features.

Specifications	**LoRa**
Frequency Band	868MHz in Europe and 915MHz in US
Transmission Bandwidth	125kHz and 250kHz

Data rates	0.250 to 11kb/s Europe, 250b/s
Range (2-Rays)	~13km
Data size	59 and 250 bytes
Modulation Scheme	LoRa DSSS and CSS
Spectrum	Spread Spectrum
Applications	IoT and M2M based applications
Receiver sensitivity	-137dBm
Bandwidth	250kHz
Time synchronization	Supported
Latency	Insensitive to latency

2.3 Telensa

Telensa being a proprietary technology first developed by Telensa (now by Wireless Internet of Things Forum (WIoTF)) is an ultra-narrowband Low Power Wide Area network, which has proof so far to be the best in street lighting for over a decade.

Telensa provides an end-to-end solution for LPWA applications in license-free sub-GHz ISM band with low data rates, focusing on smart city applications like intelligent lighting, smart parking, to name but a few [7]. The features of Telensa are outlined in table 3 below.

Table 3. Telensa features.

Specifications	**Telensa**
Frequency Band	868MHz (Europe), 915MHz (US), 430MHz (Asia)
Data rates	62.5b/s (UL) and 500b/s (DL)
Modulation Scheme	BFSK, FHSS
Range	3km urban
Spectrum	UNB

Receiver sensitivity	-135dBm
Applications	Street lightening, smart city wireless networks
Bandwidth	625Hz
Time synchronization	Not supported
Latency	N/A

2.4 EC-GSM-IoT

The acronym for EC-GSM-IoT is Extended Coverage GSM IoT. It is a LPWA technology designed as a high capacity, low energy, long range and low complexity cellular system based on eGPRS for IoT communications, co-exiting with 2G, 3G, and 4G mobile networks [8]. It is a licensed NB spectrum that functions in the GSM band of 850-900MHz and 1800-1900MHz, intended for M2M and IoT traffic only, multiplex with GSM/EGPRS traffic channels with TDMA/FDMA access technology.

According to Ericsson, the resulting EC-GSM functionality enables coverage improvements of up to 20dB with respect on the 900MHz band, this entails that new software on existing GSM networks are enough and can give a combined capacity of about 50,000 devices per cell on a single transceiver [1]. The technology supports network synchronization. Table 4 below shows the summary of its features.

Table 4. EC-GSM-IoT features.

Specifications	**EC-GSM-IoT**
Frequency Band	850-900MHz and 1800-1900MHz
Bandwidth	200kHz (UL&DL)
Modulation Scheme	GMSK (UL&DL)
Spectrum	N/A
Data rates	20-240kb/s
Multiple Access	TDMA (UL&DL)
Receiver sensitivity	-127.7dBm

Applications	M2M and IoT traffic
Time synchronization	Supported
Latency	700ms-2s

2.5 ZigBee

ZigBee is an open standard ISM band technology developed since 1998, with a mesh IoT network ranging from 10-100m. This technology can be use in home automation, healthcare, smart lightening to name but a few. It is a low-cost low-rate network with a bit rate of 250kb/s and short latency of 30ms-1s.

Summary of its features can be seen in table 5 below. ZigBee supports Beamforming and synchronization, detailed study of ZigBee Synchronization in time domain was studied in [9].

Table 5. ZigBee features.

Specifications	ZigBee
Frequency Band	2.4GHz (global) and Sub-GHz (regional)
Bandwidth	Up to 2MHz
Modulation Scheme	BPSK+O-QPSK and DSSS
Spectrum	N/A
Multiple Access	CSMA/CA
Bit rates	250kb/s
Receiver sensitivity	-98dBm
Applications	Industrial control, Home automation
Time synchronization	Supported
Latency	30ms-1s

2.6 Weightless-SIG (P/N/W)

Weightless-SIG which stands for Weightless Special Interest Group, is an open standard technology that propagate in a range of 2-5km, which is the reason it is classified as a low power medium range technology.

This technology supports M2M communications. Its P standard is a narrow-band technology with low rate signals in the sub-GHz bands, with two-way communication. The N standard is a one-way communication technology with ultra-narrowband signals in the sub-GHz bands, while its W standard is also a two-way communication technology, which uses cognitive radio and TV white spaces for increased capacity.
The table below illustrates the features and comparison of the various weightless signals. For more explanation and understanding to Weightless-SIG standards, reader is referred to [7].

Table 6. Weightless signals features.

Specifications	**Weightless-P**	**Weightless-N**	**Weightless-W**
Directionality	2-way	1-way	2-way
Range	+2km urban	>5km	+5km(indoor) and 10km (outdoor)
Battery life	3-8 years	10 years	3-5 years
Application	Smart metering	Smart Oil and Gas	Retail/shelf updating, healthcare
Spectrum	NB	UNB	SS (TV white space)
Data rates	0.2-100kb/s	30-100kb/s	1kb/s -10Mb/s
Frequency Bands	898MHz	898MHz	490-790MHz
Modulation scheme	GMSK, QPSK	DBPSK	16-QAM, DBPSK
Receiver sensitivity	-134dBm	N/A	N/A
Time synchronization	Supported	N/A	N/A
Latency	Low	N/A	N/A

2.7 NB-IoT

NB-IoT meaning Narrow-Band IoT is a new cellular technology introduced by 3GPP in the release 13 for narrow band systems with low power consumption, that supports M2M communications, providing connectivity to low-data-rate devices with low cost. It has a bandwidth of 200kHz, supports half duplex with a maximum data rate of 250kb/s, has no roaming like other cellular devices, but supports fixed devices only with a latency of 1.6s-10s, it has a long lifespan with a battery life of 10 years and more. It functions in the LTE band and GSM band, with channel access methods OFDMA in uplink and FDMA in downlink, modulation schemes BPSK and QPSK.

This technology operates in three different modes such as; standalone, in-band and guard band. In standalone NB-IoT is deployed within one or more existing and re-farmed GSM carriers and can use all available BS transmission [10]. The resources can be operator's spectrum fragments with non-standard bandwidth [14].

In-band is within normal LTE carrier uses same PRBs as LTE, some scheduling restrictions, sharing the transmit power between legacy LTE and NB-IoT operation [10]. Guard band has an unused resource blocks within LTE carriers guard band are utilized, and its cell is served by same BS, sharing same transmit power. Less interference is expected like in In-band [10].

The technology support synchronization in both time and frequency domain, with two synchronization signals; Primary and Secondary. Table 7 below illustrates the main features of NB-IoT.

Table 7. NB-IoT features.

Specifications	**NB-IoT**
Frequency Band	LTE bands and GSM bands
Bandwidth	200kHz
Data rate	250kb/s
Modulation scheme	BPSK and QPSK
Spectrum	NB
Multiple access	OFDMA(UL) and FDMA(DL)
Applications	Pet tracking, Home security, smart waste monitoring.
Receiver sensitivity	-141dBm

Time synchronization	supported
latency	1.6s to 10s

2.8 IoT performance targets

As already highlighted in sections above, IoT targets are to supports low cost device, long battery life, extended coverage, low data rate, quality of service, scalability and system capacity, all these constitute the performance targets of the network. Below a brief discussion and comparison of the various performance targets are addressed.

2.8.1 IoT device scalability

One important factor of IoT is to support massive number of devices in its network. These technologies excel well with the increasing number of connected devices as well as their density. For the scalability to be effective, different techniques needs to be taken into consideration, for instance time and space, and efficient exploitation of diversity in a channel [11].

2.8.2 IoT latency and battery lifetime

IoT target is to support long battery life and low latency. Almost all of the devices operate using battery, so it becomes a challenge to keep their lifespan long enough, hence there is a tendency for systematic awakening of the device from sleep mode to help recover new information, which is a concern because the longer the devices are asleep the more the energy is preserve which limits the consumptions, in turn limit the exchange of information, which also affects their performance resulting to high latency.

This explains the reason why IoT technologies have different latencies and battery life span because of their different mode of operations, for example in this context NB-IoT will consumes more energy unlike Sigfox and LoRa because its OFDM/FDMA modes needs extra peak current, which will shorten its battery lifespan when comparing with Sigfox and LoRa, in return it has a better latency and makes it an ideal technology for applications that needs low latency [11].

2.8.3 IoT capacity

IoT has remarkably broaden the capacity from manually operated devices to wireless smart devices operating using the internet for remote controlling, which also save cost.

2.8.4 IoT quality of service

IoT technologies like Telensa, Sigfox and LoRa that uses an unlicensed spectra and asynchronous communications protocols, have good interference, fading and multipath propagation characteristics, but their quality of service is poor as compared with that of NB-IoT which has a licensed spectrum, with LTE synchronous protocols excellent for quality of service with high cost. Hence, for guaranteed quality of service without considering the cost, NB-IoT is a better option [11].

2.8.5 IoT network coverage and range

Sigfox technology is one of the IoT technologies that offer good coverage possibility, in that it can cover an entire city with just one base station with a range of over 40km, LoRa for instance with a range of less than 20km needs three base stations to cover an entire city [11], while NB-IoT with much lower range of less than 10km and coverage capabilities aims to improve cell coverage by trying to achieve maximum coupling loss (MCL) of 164dB through sensitivity enhancement of 20dB, it has a standalone and in-band link budget which from different studies, it gives MCL equal to 164dB, achievable for channels in Rel. 13 features [12].

For long coverage deployment, NB-IoT needs narrow bandwidth in UL with lower transmit power, higher SNR with constant transmission power, extensive use of repetition codes with coherent combining at the receiver and effective coding in UL with simple coder on transmit side. [16] gives details of link budget performance analysis for coverage.

2.8.6 IoT comparative summary

This section shows the summary table of the various IoT technologies as shown below.

Table 8. IoT technologies summary.

IoT Technologies	**Modulation Scheme**	**Spectrum**
NB-IoT	BPSK and QPSK	NB
Sigfox	DBPSK and GFSK	UNB
Telensa	BFSK and FHSS	UNB
LoRa	DSSS and CSS	Spread Spectrum
Weightless-P	GMSK and QPSK	NB
Weightless-N	DBPSK	UNB

Weightless-W	16-QAM and DBPSK	Spread Spectrum
Zigbee	BPSK+O-QPSK and DSSS	N/A
EC-GSM-IoT	GMSK	N/A

3. TIME-SENSITIVITY IN IOT NETWORKS

In this chapter, we are going to discuss what is time-sensitivity, the metrics use in time sensitivity and finally then talk about time-sensitivity in an IoT network.

3.1 Time-sensitivity

Time-sensitivity can be described as the transmission of signals or data at a precise time or the reliability of the transmitted signal in each time frame or the duration of the said signal in the channel during a given time.

In some cases, the sensitivity can also be in the device used, or the program. If is not sensitive enough it can make the program or device to be slow, that is taking more time to function than usual, which can lead to more consumption of power, or even loss of information along the way that can cause the program or device to fail. Taking for instance the IoT technologies described above like Weightless-W which is use in health care needs to be very time sensitive, even the other technologies functions in one aspect or the other need good time-sensitivity for proper functioning.

3.2 Metrics used in time-sensitivity

The definition of time-sensitivity can be further developed as follows [19];

Let's assume that data samples are denoted by S_{t_i} where t_i is the $i-th$ time instant. Taking that the limit of the data sampled is given as S_{min} for the minimum sample data, S_{max} for maximum sample data, $S_{t_{i-1}}$ and S_{t_i} they stand for information taken from the adjacent sample moment t_{i-1} and t_i also given that β, where β≥1, being time coefficient of the corresponding device, and the shortest sample time interval is Δt, which is fixed. This implies that the two adjacent sample moments are given as $t_i - t_{i-1} = \beta\Delta t$, and $|S_{t_i} - S_{t_{i-1}}|$ is the change of sample data for βΔt.

Hence the time-sensitivity metric at the moment of ti is given as;

$$k_{t_i} = \left\lfloor \frac{|S_{t_i} - S_{t_{i-1}}|}{\beta(S_{min}, S_{max})} \times A \right\rfloor \quad (1)$$

Here, A is the limit of time-sensitivity, k_{t_i} is the dimensionless integer, with values ranging as $0 \leq k_{t_i} \leq A$, which can measure the change that the sample data has at t_i. The sample frequency of a device is restricted by the device itself and the ability of its IoT node, every device has an upper limit of sample frequency, and in order to get an exact sample, Δt must meet $\Delta t \geq \frac{1}{f_{max}}$ where f_{max} is the maximum sample frequency.

From this definition we can conclude that the device could be assembled as stated by k_{t_i}, which is dynamic, and the same device can have different time-sensitivity at varying moments.

Another metric use in IoT time-sensitivity can be derived using the information on Data Loss Tolerance L_i and Weight Computation W_i as;

$$S_i = \frac{2W_i - (1-\alpha)(1-L_i}{\alpha} \tag{2}$$

Here the Time-sensitivity S_i defines how quickly the data of an IoT device need to be processed, for instance in health care system, fire sensors, create time sensitive data requires it to be served immediately [20]. The metric used for device 'i' has value at the range $0 < S_i \leq 1$, as shown in table 9 below. Data Loss Tolerance L_i on the other hand shows which percentage of the produced data of an IoT device could be forfeited by the that device, for instance IoT devices like temperature monitoring devices can forfeit up to a large amount of data while services such as emergency health care could be very data sensitive [20]. This can have value in the range $0 \leq L_i < 1$, as shown in table 9 below. Weight Computation W_i is the computation of the data loss tolerance and time sensitivity of an IoT device 'i' which indicate how much they impact the bandwidth allocation. And finally, α is a constant used to assign relative priority among time-sensitivity and data loss tolerance of device [20].

This section can be summaries by classifying the percentages of time-sensitivity and data loss tolerance service-oriented bandwidth allocation at the gateway with different IoT services as shown in table 9 below [20].

Table 9. IoT services classification.

Type of Service	**Time-Sensitivity**	**Data Loss Tolerance**	**Applications**
Low powered but High data sensitive	0.9	0	Health care system, Emergency system, Fire sensors.
High powered and High data sensitive	0.7	0.3	Video analytic, Ray tacking enabled imaging.
High powered but less data sensitive	0.5	0.5	Chrome cast, Connected camera.
Low powered and less data sensitive	0.3	0.8	Temperature sensor, Humidity sensor, blood pressure sensor.

3.3 Time-sensitive IoT networks

The emergence of Internet of Things means there will be need for high data transfer over several distributed networks, this transformation will require some standards for network distribution and transportation of critical information, for instance Police vehicle, or ambulance on an emergency service are given priority in the traffic. This can be achieved using Time-Sensitive Networks (TSN) which is an IEEE 802 Standard [17] [18].

The vision of IoT is to make things smarter than ever, connect devices and infrastructure in which case machines, electrical grid and transportation of systems will be replaced by embedded sensing, which when networked together they will form a smart system of systems. Such systems will generate great deal of data, like condition monitoring solution for the Victoria Line of the London Underground rail system, which produce 32 TB of data every day [17] [18].

Even though some of these data are not time critical and may be sent through network layers and subsystems with less consideration of synchronization or latency, time-sensitive data must be transmitted and distributed within a constrained of latency and reliability. A handful amount of network infrastructure of today is not equipped to manage such time-sensitive data. Some network systems in the industries were equipped to carry out some specific tasks, each layer has different level of latency, bandwidth and quality of service, making it more difficult for interoperability and data connections seems almost impossible [18].

Manufacturers and consumers need reliable, remote, and secure access to smart edge devices in other to be able to support the capabilities of IoT infrastructures [17]. TSN makes sure critical time-sensitive data are transmitted on time on-like the existing network infrastructure. Figure 1 below shows the standard Information Technology (IT) and time-sensitive data convergence to connect devices and the enterprise [18], Also, TSN needs clocks that are perfectly synchronized.

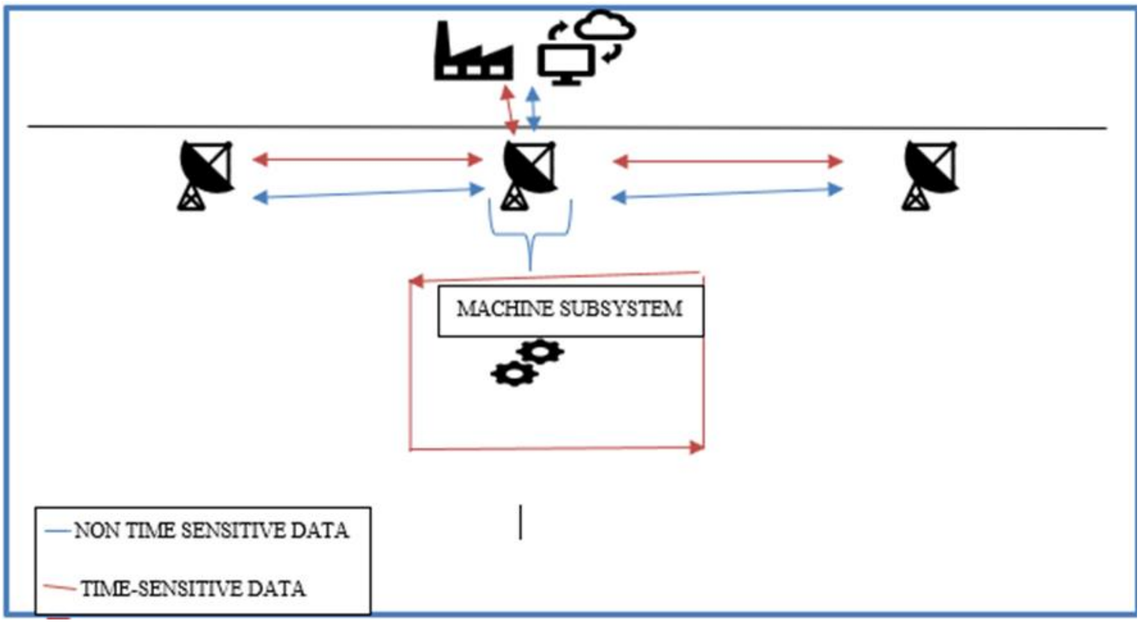

Figure 1. Standard IT and Time-Sensitive Data Converge to Connect Devices and Enterprise.

4. TIME AND FREQUENCY SYNCHRONIZATION SOLUTIONS IN IOT NETWORKS

Before going into the time and frequency synchronization solutions in IoT networks, let's look briefly into the reason why we need synchronization in the first place. We understand that for two or more systems to function at same time with high speed, accuracy and reliability, they must be well synchronized, and hence the systems will be able to communicate among themselves, also the system needs to be time sensitive enough so that it will not experience failure at some point in time, as described in chapter 3. For instance, in cellular IoT networks, synchronization is achieved by signalling, that is, there is transmitting and receiving of message signals from the network and terminal and vice versa.

Also, to meet time-sensitivity in the network, we need to avoid much signalling in IoT networks, because it consumes the battery of the terminal, hence making it less efficient. Finally, clock compatibility must be treated as utmost important because it enables the receiver to follow up the timing messages from the transmitted signal. To be further explained in Chapter 5.

Synchronization at the receiver is carried out in three (3) levels namely, frequency, time and phase. We briefly explain these below, to get the basic idea about them before using the techniques to perform the measurements that will follow, as the thesis required.

Time Synchronization

Time for instance same year, month, day or even hour, that is there is a common origin and are synchronized with frequency and phase. This can also be related to frame synchronization [12]. Like phase synchronization, requires high accuracy and stability.

Phase Synchronization

Phase synchronization is achieved by matching the time for coherent detection. It is sometimes not easy to achieve, since it needs high accuracy and stability, hence systems become complicated as a result.

Frequency Synchronization

This is done by matching the rate of the local oscillator. It is easy to realize this technique as many devices or systems has frequency calibration and can provide almost identical frequencies when needed. We can achieve this in our case study being NB-IoT by using two or more sinusoidal signals with similar frequency.

After seeing the reason why, we need synchronization in IoT networks, lets now see solutions or requirements in time and frequency synchronization in IoT networks.

4.1 Close loop methods with explicit channel feedback for each transmitter

This technique is based on a single receiver and multiple transmitters. For example, figure 2 below illustrates two transmit signals that combine coherently at the receiver [21]. This technique has also been analyzed in [22], [23].

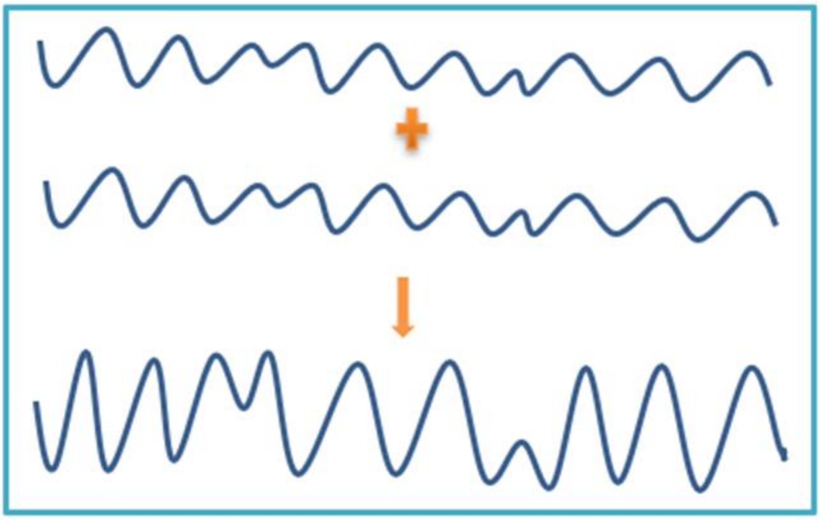

Figure 2. Two Transmit signals combine coherently at the Receiver.

The network synchronization methods to be consider for this technique are Mutual synchronization and Master-Slave synchronization.

Mutual synchronization works together with two individual clocks to find the common time scale, which is a good approach only when there is no superiority among the clocks and robustness of common time scale, with respect to the drift of any clock, which is crucial [21]. The disadvantages of this approach are that, to determine the common clock scale, there is a lot of energy consumption due to high overhead, and it needs a multiple access scheme to distinguish one clock from another [21].

Master-slave in the other hand can transmit the signals to meet coherently at the receiver with little energy lost even if the common time scale is robust. It has no problem with clock drift so long as each slave clock keeps track of the change of situation of the master clock well [21]. This method needs pre-compensation for each transmitter in order to compensate unequal propagation delays.

For this closed-loop technique we use Voltage Controlled Oscillator (VCO) as the slave clock, this approach has a problem of stability and tracking with delays from master-to-slave transmission and from slave-to-master transmission, hence for stability, the bandwidth must be small, causing the tracking ability to reduce [21]. This approach is depicted in figure 3 below, where the VCO tracks the frequency of the master clock.

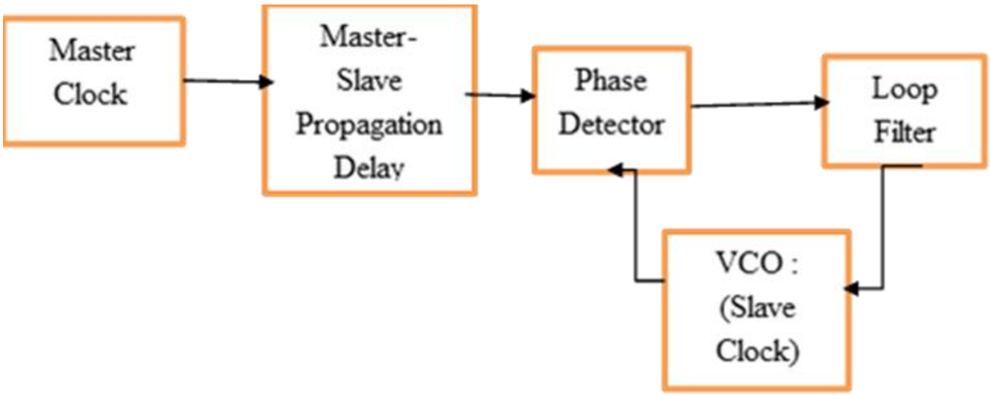

Figure 3. Master-slave Synchronization.

4.2 One-bit aggregate feedback

This feedback algorithm is used for the phase adjustment of beam-steering based on feedback packet payload [25]. This technique is the most popular because of its simplicity and scalability for practical implementation, easy to implement with low overhead [24], [25], [26].

The algorithm functions as follow; transmit node adds random phase perturbation at each time slot to its current phase, Received Signal Strength (RSS) is been estimated by the receiver, then broadcast a single bit to the various transmitters, thereby informing if the previous time slot is higher or lower than the estimated RSS [26]. For this scenario, the transmit node will keep the last perturbation, if RSS is higher, and move on to the next time slot, but if RSS is lower, the transmit nodes changes their phase to the previous time slot, then move on to the next time slot [26]. This technique has also been explained in [24].

The feedback is used simultaneously and in parallel to the Extended Kalman Filter (EKF) for frequency and phase synchronization [26]. This technique can be analysed in two synchronization sub-process; Frequency locking and beam-steering, while in frequency locking each transmitter locks its oscillator to a shared reference signal, the beam-steering adjust the phase relationships between the transmitters to add up coherently to the specified receiver as in figure 4 and 5 below [25].

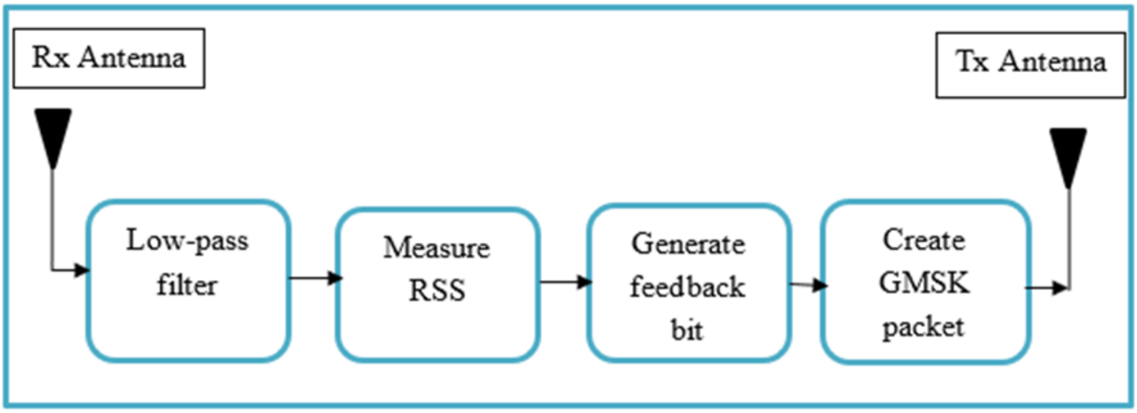

Figure 4. Block diagram of receive node.

For further details, refer to [25] which gives detailed explanation of these synchronization sub-processes.

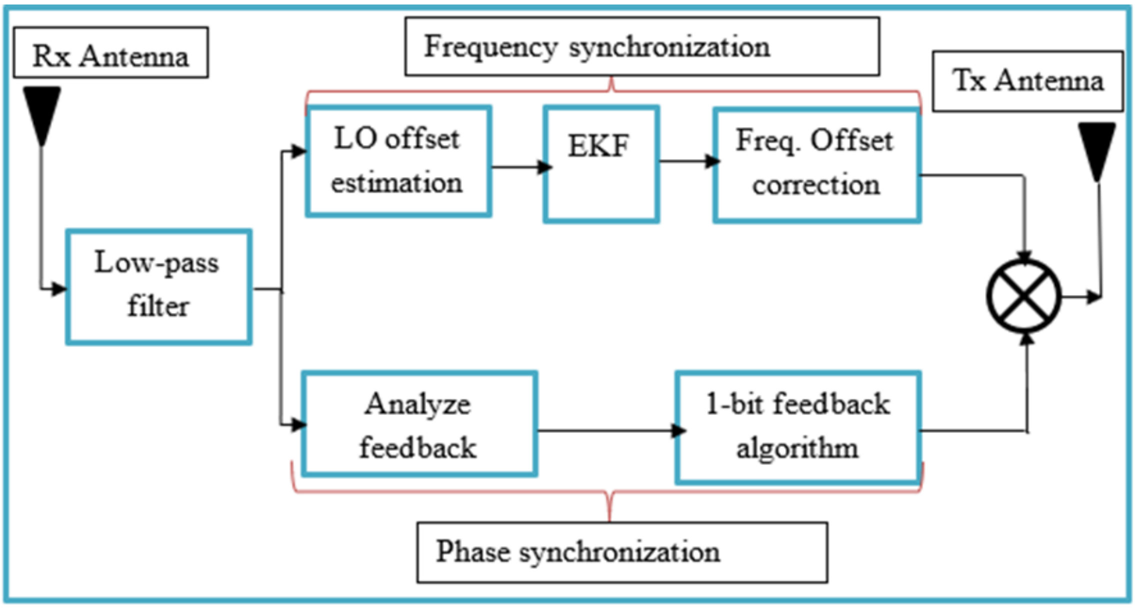

Figure 5. Block diagram of transmit node.

In the frequency synchronization process, the following drawbacks are encountered;

- The high Local Oscillator (LO) frequency offset between the transmit nodes makes it difficult for the distributed beamforming to be implemented on a Software Defined Radios (SDR) [26],
- It requires continuous pilot tone, if this is interrupted at some point, because of software lag, there will be lost of synchronization [26],
- There is small but continuous frequency offset fluctuation in the synchronization signal since the Costa loop (Phase-locked loop (PLL)) is not able to filter out LO phase noise [26].

4.3 Implicit feedback using reciprocity

Implicit feedback using reciprocity particularly at the Time Division Duplexed (TDD) system is one technique used to obtain Channel State Information (CSI) at the transmitter [24]. Reciprocity is used in a centralized multi-antenna transmitter to perform beamforming, by estimating complex channel gains of each antenna component, details about this method is studied in [27].

For channel estimation in implicit feedback, the client being the beamformee transmit sounding frame to the Access Point (AP) herein referred to the beamformer, while the AP determines the CSI based on channel reciprocity principle, on like in explicit feedback where it is in reverse order [28]. The advantage of the implicit feedback is that the AP does the CSI calculation, thereby lessen the burden on the client, also the sounding frame is sent once by the client with no CSI feedback, because the data frame follows the sounding frames immediately, therefore transmit with less time [28].

Apart from having good time interval between its transmissions, it also has some drawbacks such as;

- it needs calibration, assumptions behind calibration correction are required to be validated and finally the receive antenna which cannot transmit (receive-only antennas) [28],
- is will not able to collect sounding feedback from the client [28]. This drawback can be seen in a situation where the client has more antennas than transmitters and using switching as seen in figure 6 below, where client transmits sounding feedbacks from each of its antennas separately after switching.

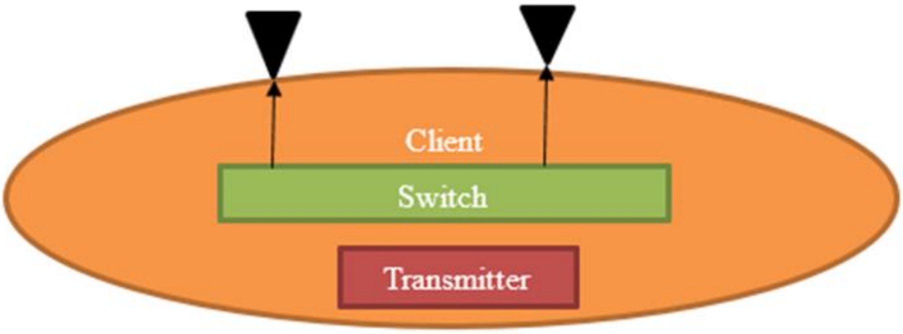

Figure 6. Client with more antennas than transmitter after switching.

Implicit feedback Beamforming has equal reciprocity characteristics between the transmitter and the receiver, but the interference seen between the two is not reciprocal, also their distortions are not reciprocal, that is the reason it needs calibration to give accurate channel characteristics between the transmit and receive stations [29].

4.4 Round-Trip synchronization

Round-trip synchronization is an open-loop carrier synchronization technique which removes the requirement of digital signalling during synchronization based on round-trip propagation delays via multi-hop chain of source nodes, where it contracted the name round-trip carrier synchronization [24].

It does not need much interaction between source nodes and destination nodes as depicted in a two-source round-trip model in figure 7 below. The idea of this technique is that, an unmodulated beacon that jumps around the circuit (figure 7) in a clockwise manner will have same phase shift as the one that jumps in an opposite direction (counter-clockwise) when the channels are reciprocal [24], [30].

In beamforming, the implementation of this technique is quite complicated, but by the restrictions that wireless transceivers might not transmit and receive on same frequency at the same time [24].

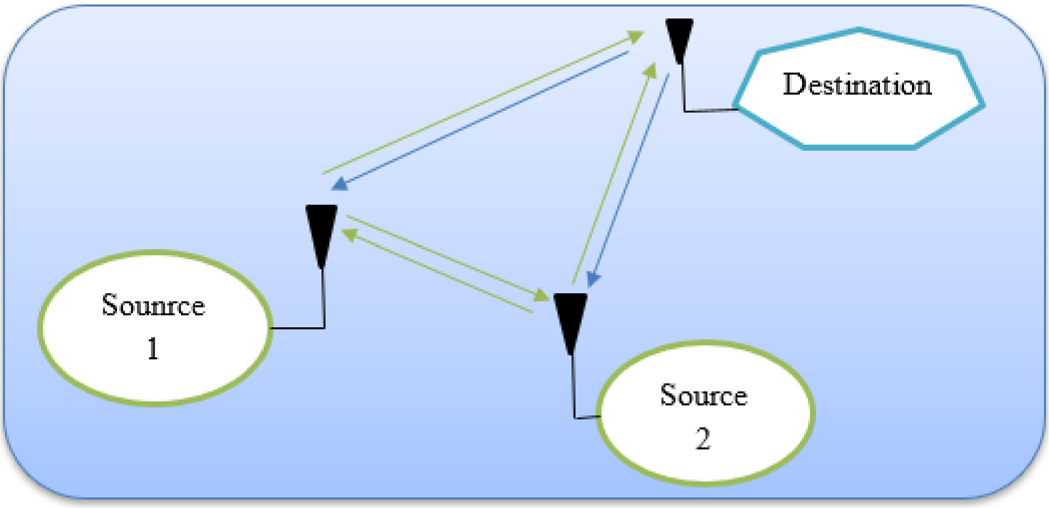

Figure 7. Round-Trip Open-Loop Carrier Synchronization.

A time-slotted round-trip carrier synchronization protocol was proposed in [30] and [31], for both two-source and M-source respectively. It is a half-duplex through time division of the channel, that functions with single frequency in all the beacons, from this experiment, it is shown that it has small overhead synchronization when it comes to beamforming gain, with total of four time-slots, three of them are used for synchronization and the fourth for beamforming, as described in [30].

4.5 Two-way synchronization

Two-way synchronization is simply the time exchange between two or more devices, which can be denoted as clients and server, with the exchange first initiated by the client.

This method works in this manner; the client sends a request at a time say t1, and the server receives the request at another time say t2, the server then send back a reply with time t3, the client receive the reply at time t4, here t1 and t4 are referred to as the timestamps which is from the client's clock, and t2 and t3 are from the server's clock, so the round trip time (RTT) is given by (t4-t1)-(t3-t2) [32], as illustrated in figure 8 below. This synchronization technique is described in [33] for accurate source synchronization and retrodirective distributed beamforming. Two-way synchronization is used in analysing the statistical properties of phase and frequency estimation errors [33]. The numerical analyses of [33] shows that the beamforming performance in near-ideal situation can be achieved with low synchronization overhead.

Two-way synchronization is from within synchronization technique which fulfils the sub-carrier-period timing accuracies needed for distributed beamforming, can be used to rectify GPS time estimate when there is outage [34]. This technique is like the techniques described in time-slotted round-trip synchronization in [31] but have some differences which are explained detailly in [34].

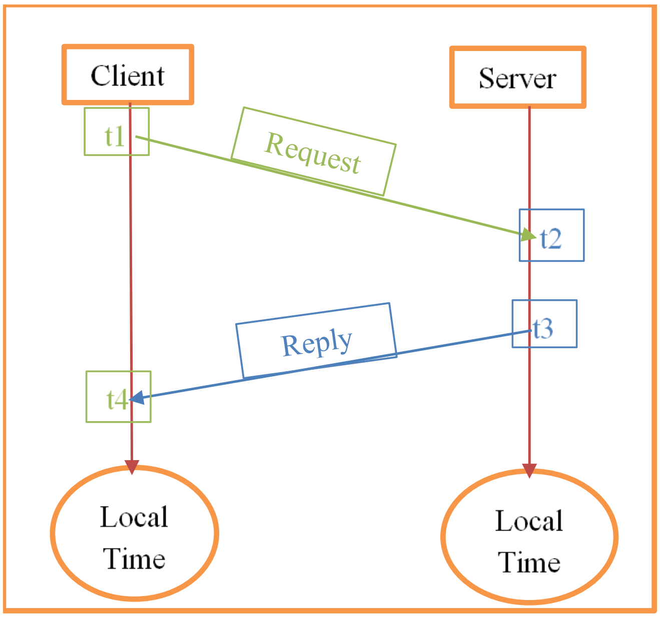

Figure 8. Two-way synchronization illustration.

Table 10 below summarises the above techniques in terms of time synchronization, frequency synchronization, their advantages and disadvantages, where they were found that is the references and their complexity.

Table 10. Summary of the synchronization solutions features.

Synchronization Solutions	**Time Synchronization (Yes/No)**	**Frequency Synchronization (Yes/No)**	**Main Advantages**	**Main Disadvantages**	**References**	**Complexity (Low/Medium/High)**

Closed-Loop with Explicit Channel Feedback for each Tx	Yes	Yes	Dissipate low energy	Low stability and reduced tracking.	[21] [22] [23] [24]	Medium
One-bit Aggregate Feedback	N/A	Yes	Simple to implement, scalable.	Requires continuous pilot tone.	[24], [25], [26]	Low
Implicit Feedback using Reciprocity	Yes	Yes	Time efficient	Requires calibration	[24], [27], [28], [29]	High
Round-Trip Synchronization	Yes	Yes	Requires minimal interaction between source and destination nodes	Implementation complicated in Beam-forming	[24], [30], [31]	Low
Two-way Synchronization	Yes	Yes	No feedback system is needed		[32], [33], [34]	Low

5. CLOCK AND FREQUENCY ERROR MODELS

5.1 Clock synchronization errors

Clock synchronization is a necessity in all devices that uses any form of network connection, be it wireless or wired, otherwise the performance of said device will not function accurately. It is also understood that devices that belongs to a network have different clock oscillator with their own drift and offsets, which must be synchronized in time for proper implementation and communication among them with the central station, so that there will be a constant update about the clock offset in a way to maintain the synchronization. Synchronization in general needs the receiver clock to achieve and trace the periodic timing information in a transmitted signal [12].

IoT is a new era of technology that has different phases of sensed data transmission and collection. These performances can be triggered at the main server at any time, so keeping the data updated is more needed than to sense them, because late arrival of data or incoherently, will hinder the transfer which will result to failure and can be costly, implying the network will not be time sensitive, so active and accurate synchronization is the ideal solution to keep the data and device parallel to each other to enhance performance [35].

This means any IoT device need to have a precise clock offset for synchronization, because clock value in real time system has much impact on a network, either it is analogous or it is divergent, as shown in figure 9 below, where dt/dc must be equal to 1 in order for the system to be on time for perfect data transmission, which does not happen like that in a real network, hence the reason why clock adjustment is required [35].

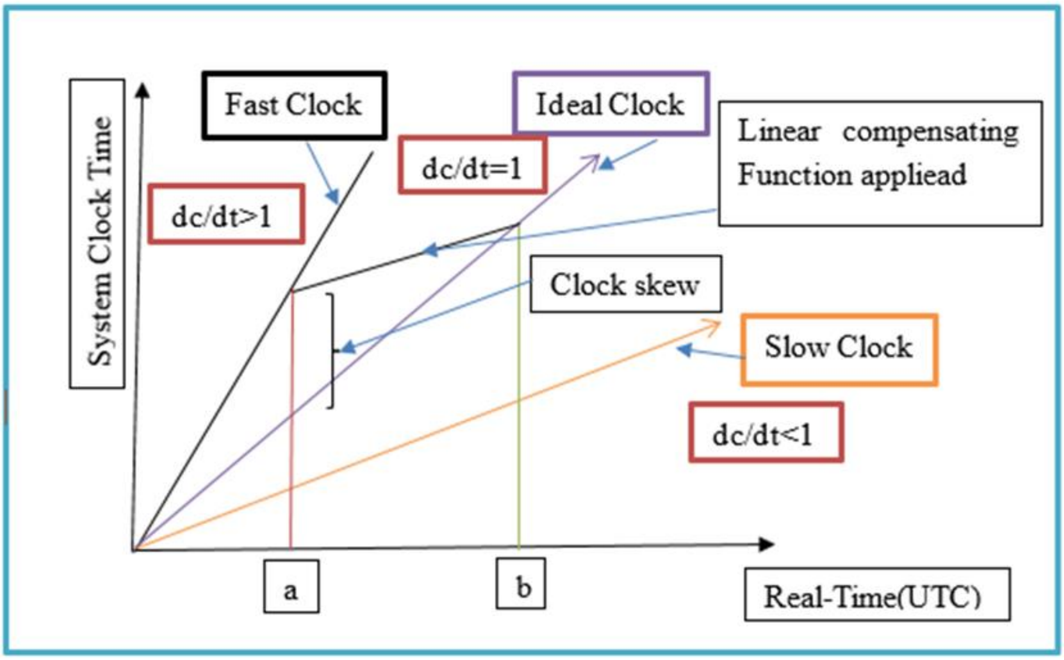

Figure 9. Clock Error and Drift Compensation.

The clock becomes faster when dt/dc is greater than 1, hence the data arrives before it sends, and the clock is slow when dt/dc is less than 1, implying the data arrives after a short delay period, in this situation, the difference amongst the ideal and fast clock is referred to as clock skew, and this can be taken care of by utilizing logical clock which adjust when system is on the network as the data is being transmitted [35]. Even though this clock skew is seen as a prevailing source of synchronization error, it is especially neglected in distributed clock synchronization algorithms [32].

Clock drift in this case is referred to as the situation where a clock device that runs on a frequency oscillator with different frequency as compared to other devices generating the clocks to tick at distinctive rate, creating a huge gap in recognize time due to different tick rate [35].

According to [32] inexpensive clock hardware is relatively unstable likened to old-styled PC clock hardware and compelling clock drift among synchronization requests aggravate error.

5.2 Synchronization errors

It is possible to have synchronization errors in any of the synchronization types be it time, frequency, phase or even all of them, but one thing that need to be taken into consideration is that receiver must be aware to start the sampling process on the wave stream with much accuracy. Failure to maintain the time can leads to waste of bandwidth and abrupt end of transmission and possibly loss of data, which implies the receiver requires good synchronization [36].

In [36] and [37] possible synchronization errors are illustrated as seen below and depicted in figure 12 which highlights the effect of sampling clock offset and symbol clock offset.

- *Carrier frequency offset (CFO)*; causes the received baseband signal to rotate at a frequency of Δf. More of this will be discuss later in this chapter.
- *Carrier phase error (CPE)*; introduces an additional phase rotation term, φ(t), into the received baseband signal.
- *Sampling clock offset (SCO)*; δ, occurs when sampling the received continuous-time waveform at an interval of $(1+ \delta)T_s$ instead of an ideal T_s .
- *Symbol timing offset (STO);* T_d, this is the error in the symbol boundary at the receiver from the actual boundary in the received waveform.

From figure 10 below, sampling at this $(1+ \delta)T_s$ interval, makes the rest sampling process to be erroneous.

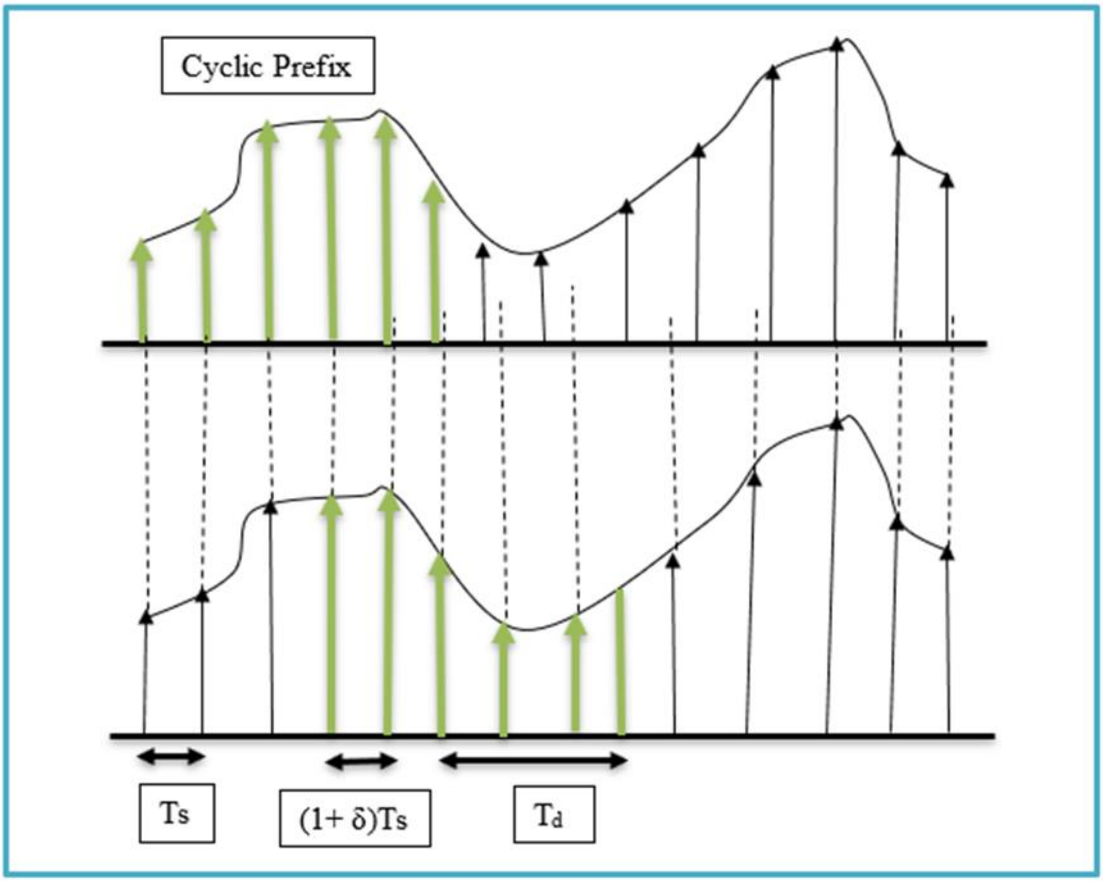

Figure 10. Synchronization Error without noise, fading, and interference for clarity.

These synchronization errors can cause block transmission, data repetition or even deletion, in some case loss of voice or data transmission, failure in traffic, mobile terminal turn to consume more battery when synchronizing with eNodeB.

5.3 OFDM synchronization

It is understood that single carrier modulations are very sensitive to time-domain errors, OFDM have the effect of frequency errors by performing Fast Fourier Transform (FFT), because of its frequency domain, even though it is more resilience [36], [37]. The FFT is built after several cycles because of heavy computations in frequency domain, which is further expected to exploit time-domain synchronization in some receivers, also the frequent occurrence of the preamble at the start of each packet having autocorrelation resources, helps synchronization of OFDM packet-based system to be carried out in time-domain [36].

One fascinating scheme of OFDM is the fading channels, but it still requires very precise synchronization. The main assignment to carry out this synchronization are symbol and frame timing, Carrier Frequency Offset (CFO), Carrier Raster Offset, and Channel estimate and equalization. Details about these assignments can be seen in [38]. For the scope of the thesis we will discuss more about CFO here.

5.4 Carrier frequency offset (CFO)

Carrier frequency offset (CFO) occurs when there is inconsistency in frequency as the transmitter aim to make use of the carrier frequency, $f_{c,}$ from the receiver, because of Doppler shift and LO fluctuation on either side [38]. The effect of the accumulated CFO is usually referred to as the scaling factor ∅.

$$\emptyset = \frac{1+LO_{Tx}}{1+LO_{Rx}} \cdot \frac{c}{c+d\prime} \qquad (3)$$

Where c = speed of light, d = distance between the units and LO_{Tx} and LO_{Rx}, being the local oscillator frequency errors expressed often in ppm for the transmitter and the receiver respectively [38]. The fact that the transmitter and the receiver have separate LO and oscillate with different carrier frequencies ($f_{c1}$and f_{c2}) might cause Inter Carrier Interference (ICI), which will affect the orthogonality of the carrier and corrupt the bit error rate (BER) [12], [38]. It is also understood that the time-drift that occurs in this process, will amount to many symbols, causing Inter-Symbol Interference (ISI) [38].

5.5 Clock models

5.5.1 Skew-based clock error models

Different systems use different clock models and finding a perfect clock model for any design will yield better clock synchronization and minimize clock skew. In order to carry out clock synchronization in IoT or for instance wireless sensor networks (WSN), all the nodes must be assembled with a local hardware clock, such as quartz crystal oscillator, which experience frequency drift k_i due to different factors, making nodes out of synchronization [39].

$$k_i = \frac{f_i}{f_0} \tag{4}$$

Where f_i is the actual frequency of the oscillator clock of $i - th$ node, and f_0 is the rating frequency. Hypothetically, quartz crystal oscillator's frequency can be exact by given an oscillator with constant frequency for a longer period, for instance minutes to hours, implying that local time of $i - th$ node can be designated as a monotonically non-decreasing function [39];

$$T_i(t) = k_i t + b_i \tag{5}$$

Here, t is the Universal Time Coordinated (UTC) or called real system time (RST), k_i being the frequency drift given in equation (4), and $b_i$time offset of the clock. Basically, each node will have different b_i and $k_i$with almost fixed time period [39].

Taking a practical situation, for instance given that Node B has clock value to be t_B, that is going to be synchronized with Node A with clock value t_A , this means that the clock value relationship among the two nodes will be given as [40];

$$t_A = t_B + \alpha(t_B - t_0) + \beta \tag{6}$$

Where β is the offset at clock B's time t_0, and α is the relative skew. This two-node model can also be implemented to a network-side synchronization, which will vary α and β among different pairs of nodes, meaning synchronization algorithm outlined among two nodes, will also work well for other pairs in the network [40].

A joint network-centric positioning and synchronization method for unsynchronized 5G ultra-dense network (UND) was studied in [40], where direction of arrival (DoA) and time of arrival (ToA) were tracked at line-of-sight remote radio heads (LoS-RRHs) utilizing extended Kalman filter (EKF). The interest for this work based on this thesis was to express how the unsynchronized clocks within a network are modelled, given that the clock model for a time-varying clock offset with a skew is assumed, as expressed in the equations below [41];

$$\rho(k) = \rho(k-1) + \Delta t \alpha(k-1) \tag{7}$$
$$\alpha(k) = \beta\alpha(k-1) + v(k) \tag{8}$$

The constant component $|\beta| < 1$, where $\rho(k)$ and $\alpha(k)$ are clock offset and skew respectively at time instant k, Δt is the time interval among $k - 1$ and k, with $v(k)$ the driving noise of the clock skew.

5.5.2 Constant error model

Constant error are errors that makes measurement to drift invariably from its expected value in a same direction, hence it has no offset, or its offset is constant. This kind of errors are not easy to identify since they are unchanged so long as the measurement conditions are still unchanged, even if the measurement is repeated several times. It is important to understand that, even though constant errors initiate a constant bias in a mean or median of a measuring data, no numerical analysis of the data can identify a constant error [43].

5.5.3 Random error model

Random error or gaussian errors are errors that are random and not easy to predict. This can also be referred to as statistical error, since it is random in nature and can be remove from a measurement by statistical means [44]. Comparing to the constant error, random error has a gaussian clock offset and are never fixed, hence they change quite often. It is also known that random errors can be taken care of by averaging, this is because they have zero expected values, hence meaning they are truly random and scattered around the mean value, implying the arithmetic mean of errors are expected to be zero [44]. The gaussian clock offset can be calculated as shown in equation (9) below:

$$G_{ClKoffset} = N(\mu, \sigma^2) \tag{9}$$

Where N is the normal distribution, sigma σ is the standard deviation, supports values from $\sigma \geq 0$ and μ is mean value, which support values in the range $-\infty < \mu < \infty$.

6. SIMULATION-BASED STUDIES

In this chapter, the simulation is carried out using three different error models as described in chapter 5 above. Based on the scope of this thesis, we study the Bit Error Rate (BER) of BPSK modulation of Narrowband and Ultra-narrowband signals of the IoT technologies, also, time synchronization with phase compensation, and symbol detection is carried out, using the above clock models.

6.1 System model

This work is carried out in such a way that the carrier modulated BPSK transmission system involving transmitter (TX), Receiver (RX), and a noise channel model in this case Additive White Gaussian Noise (AWGN), as shown in figure 11 below.

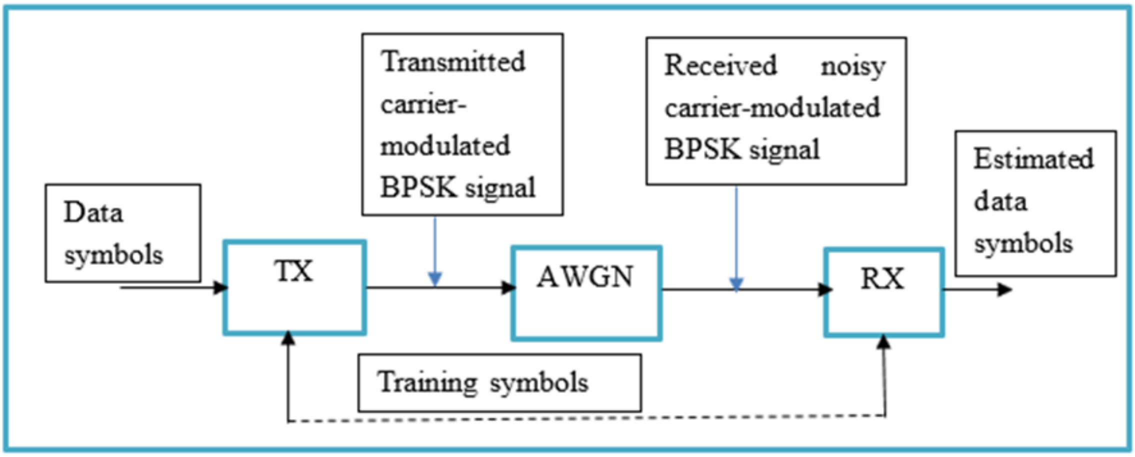

Figure 11. System model Block Diagram.

We will not go into detail study of the complex baseband signal and the I/Q modulation, but just highlighting how the simulation was done to achieve the results. Figure 12 and 13 below illustrates the transmitter and receiver structures.

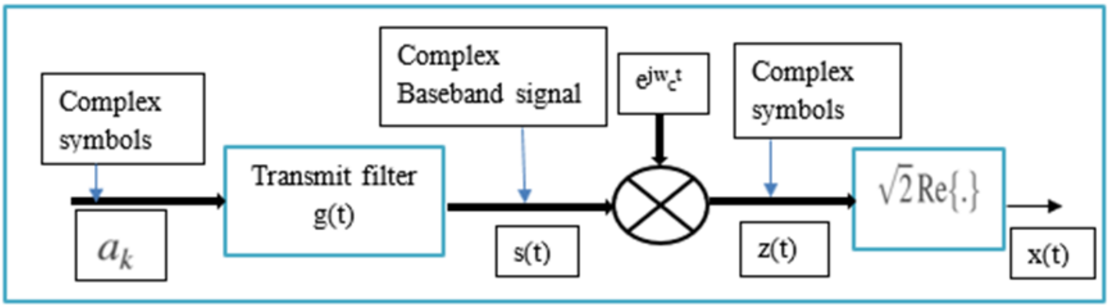

Figure 12. Transmitter (TX) Structure.

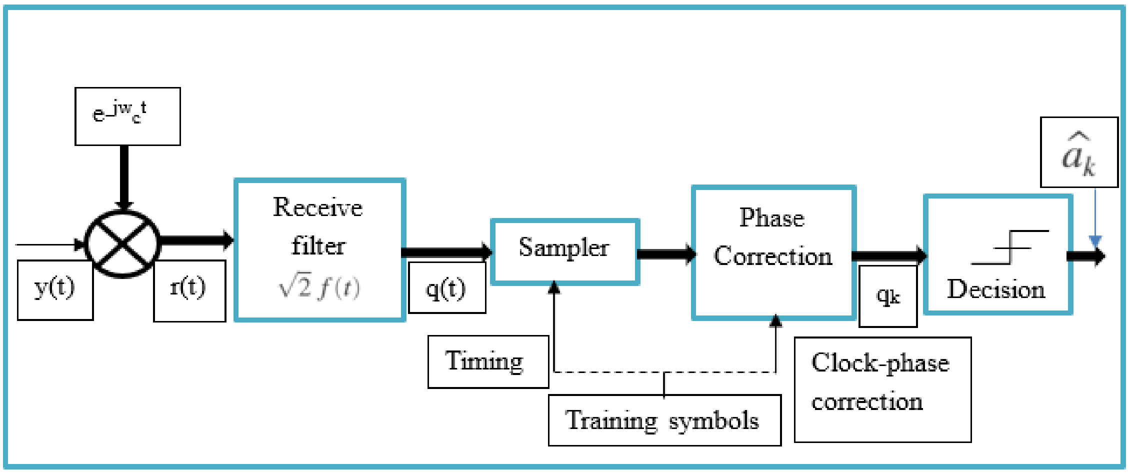

Figure 13. Receiver (RX) Structure.

A point to note here is that TX and RX structures above depend on complex calculations, and this can also be achieved using only real-valued signals such as I/Q modulation with independent I and Q components.

Root-Raise-Cosine (RRC) filters is used for the transmit and receive filters, and they both fulfils the Nyquist criterion for zero Inter-Symbol-Interference (ISI) [45]. Figure 14 below illustrates the pulse shape of this filter, generated using the general system parameters as show in the section below.

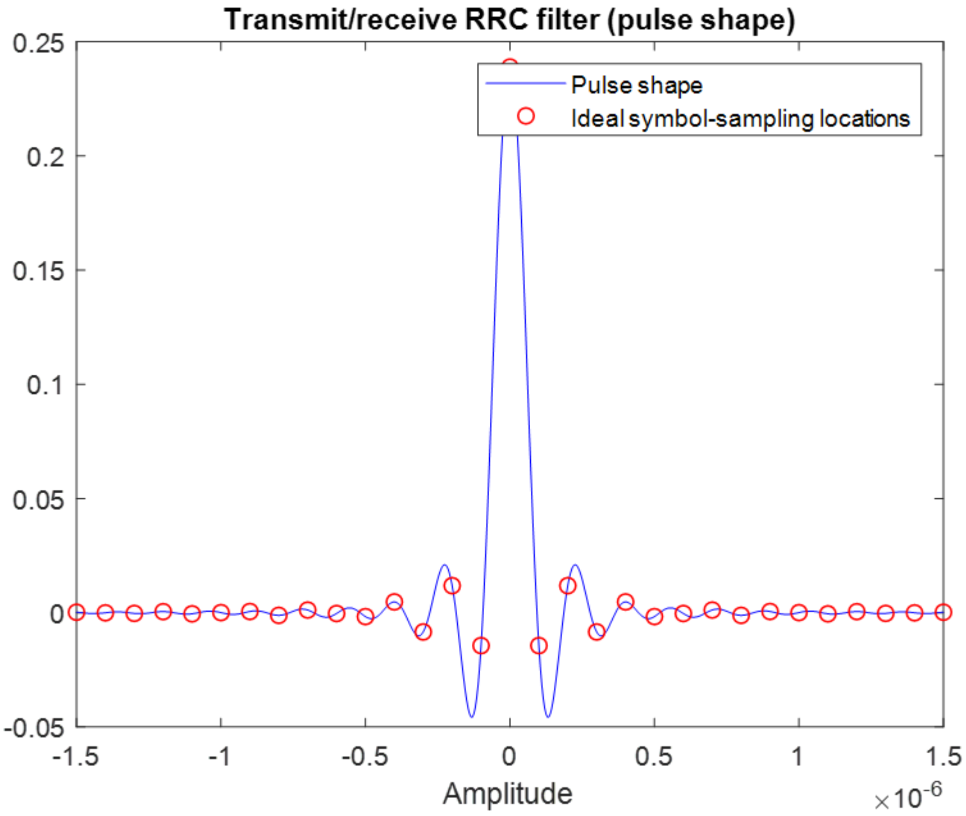

Figure 14. Transmit/Receive RRC filter Pulse shape.

6.1.1 General system parameters

Below is the table of the general system parameters that was used for both NB and UNB simulation.

Table 11. General system parameters.

Parameters	**Values/Ranges**	**Description**
SNR	-10:5:10	Signal-to-Noise power ratio in dB
N_symbols_per_pulse	30	Duration of TX/RX-filters in number of symbols
r	30	Oversampling factor (sample per pulse)
alfa	0.25	Roll-off factor (excess bandwidth)
Alphabet_size	2	Number of symbols in the BPSK alphabet
N_data_symbols	10000	Number of data symbols in samples
N_training_symbols	940	Number of training symbols in samples

The number of data symbols and training symbols given in the system parameters table above was given by defining the number of symbols to be transmitted, where for timing and phase synchronization to be achieve, the training symbols are added in the beginning of the transmitted symbol frame as illustrated in figure 15 below.

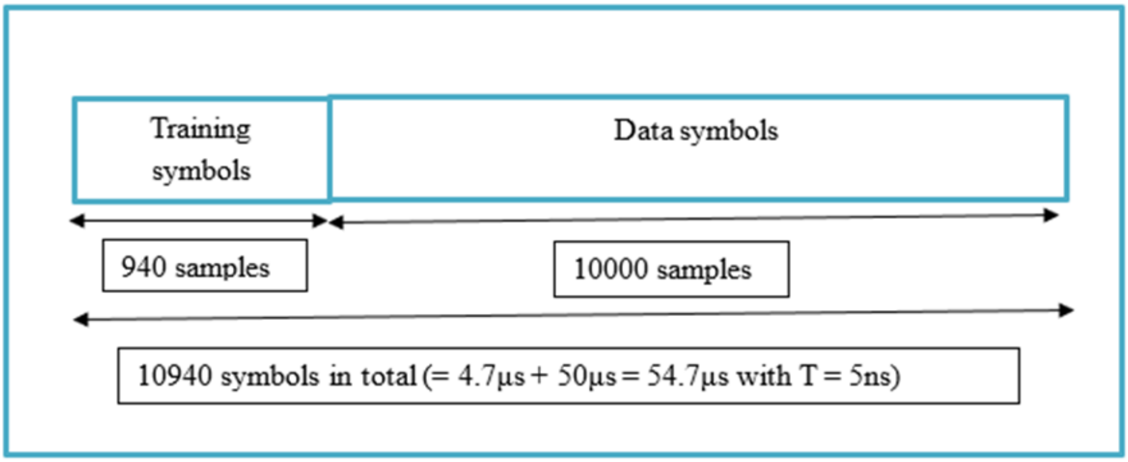

Figure 15. Symbol frame to be transmitted.

6.1.2 IoT technologies parameters

In this section, we define the IoT parameters used in the simulation. Table 12 shows the ultra-narrowband spectrum parameters, while table 13 shows that of narrowband. These spectrum parameters are used for the BPSK modulation simulation.

Table 12. Ultra-Narrowband modulation parameters.

Parameters	**Values**	**Description**
B	100	Bandwidth in Hz
T	5,000,000	Symbol time interval in Nano seconds (ns)
fc	868,000,000	Carrier Frequency in Hz
Fs	3000	Sampling frequency in Hz
Ts	333,333.33	Sampling time interval in Nano seconds (ns)
K_i	289,333.33	Frequency drift in Hz

Table 13. Narrowband modulation parameters.

Parameters	**Values**	**Description**
B	200,000	Bandwidth in Hz
T	2500	Symbol time interval in Nano seconds (ns)
fc	765,000,000	Carrier Frequency in Hz
Fs	12,000,000	Sampling frequency in Hz
Ts	83.33	Sampling time interval in Nano seconds (ns)
K_i	63.75	Frequency drift in Hz

6.2 Simulation method

After generating the carrier modulated signal with the above parameters, the transmitted symbol frame is been filtered, we do modulation / frequency translation to the carrier frequency fc. After this is done, the time vector for the oscillator signal is defined based on the reference clock time in the TX, which is later used to generate the complex-exponential carrier signal, where the real part of this signal is taken to finalise the TX process. As shown in figure 12 above.

A simple AWGN channel model is used to generate a random noise, which is scaled with a scaling factor to achieve the desired SNR, that is included on the TX signal. From figure 13 above, we estimate the transmitted symbols from the received noisy bandpass signal, a downconversion is done by demodulating the noisy bandpass signal back to baseband using same method as in upconversion in TX, with the use of the reference clock time. The received signal is later sampled to get the symbol samples.

Since the main focus for the thesis was to generate the BER with respect to the SNR, we carry out symbol-to-bit conversion, by creating input bits of the system with inverse bit mapping method, that is, by using the qamdemod function in MATLAB at the level of the TX, and used same function to get the output bit at the level of the receiver. And finally, the BER is calculated using these input and output bits generated, with respect to the errors in the system.

6.2.1 Time synchronization and phase correction

In this section we look at the common errors in a transmission system (such timing and phase errors) and how they can be corrected. The errors are introduced in the transmission system, then the symbols are synchronized, and the phase errors are compensated. This is done by adding a random propagation delay to the AWGN channel, which is estimated at the RX. Readers should note here that the phase error plot was not studied since it was not in the scope of the thesis, but it was worth mentioning. Observing that the oscillator clocks are not synchronized as mentioned in section 6.1 above, different reference clock times are defined for TX and RX [45].

For the time synchronization, an estimator for the correct symbol timing based on cross-correlation between the known training symbols and the received BPSK signal are created, which is done by upsampling the training symbols in order to match the sampling rates, where the cross-correlation is calculated as a function of time-delay between the received signal and the upsampled training symbols [45].

The phase error estimation is carried out by estimating the phase error caused by the offset between TX and RX clocks, since the training symbols' phase is known by the receiver. This can be achieved by comparing the phases between the known training symbols and received training symbols. Phase compensation or correction at this point, is obtained by dividing the signal with the channel estimate [45].

6.3 Simulation results

This section gives the simulation results of the different error models introduced into the system to plot the BER with respect to the SNR, using the Narrowband and the Ultra-narrowband spectrum as stated above.

6.3.1 Constant error case

In this section we introduce a constant error vector into the system's receiver terminal and plot the BER for perfect synchronization and imperfect synchronization results for the case of NB and UNB spectrums as shown below. The solid line represents the perfect

(Perf) synchronization curve, while the dash line represents the case when the synchronization is not perfect (Imperf).

It is obvious here that the solid line which represents the case when we have perfect or better time synchronization will have better performance in an IoT network (this applies also in the subsequent analysis of the different error models), since it has lower bits errors.

The mean error is calculated using the function mean(BER) in matlab, then converted into percentage. The total time for the transmit signal to synchronize with the receiver is estimated in matlab by measuring the start and end time of the simulation process with the function tic-toc.

The mean error percentage when we have perfect synchronization for NB-BPSK constant error (figure16) is 15.58%, while in the case of imperfect synchronization we have 17.16%, which implies more degradation of the BER is experienced because of poor time synchronization between the transmit signals and the received signals. The total simulation time it took for the transmit signal to synchronize with the receiver is 4.55 seconds.

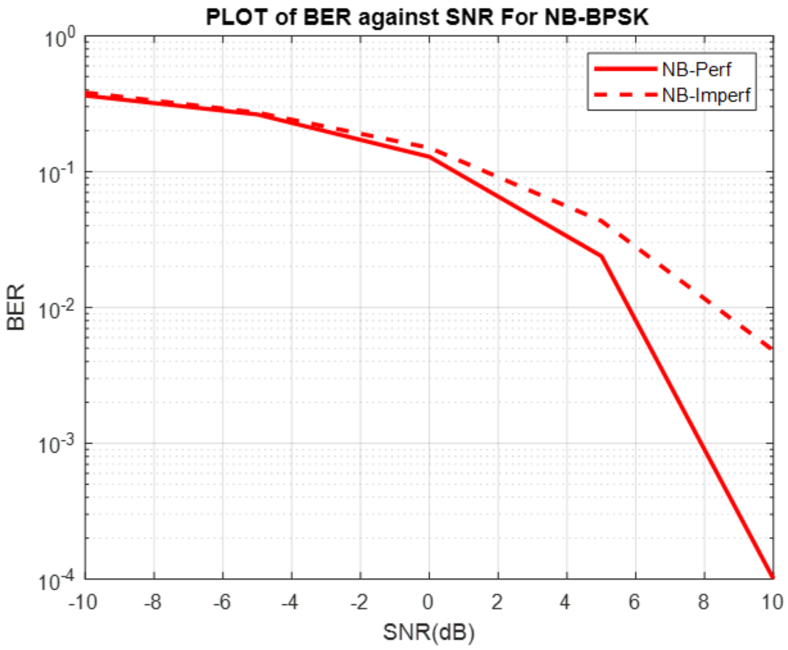

Figure 16. NB-BPSK Constant Error Simulation.

For the UNB-BPSK constant error (figure 17), the mean error percentage when we have perfect synchronization is 15.64%, while in the case of imperfect synchronization we have 16.62%, which shows the degradation is not as high as the case with NB-BPSK IoT spectrum. The total simulation time it took for the transmit signal to synchronize with the receiver is 4.52 seconds

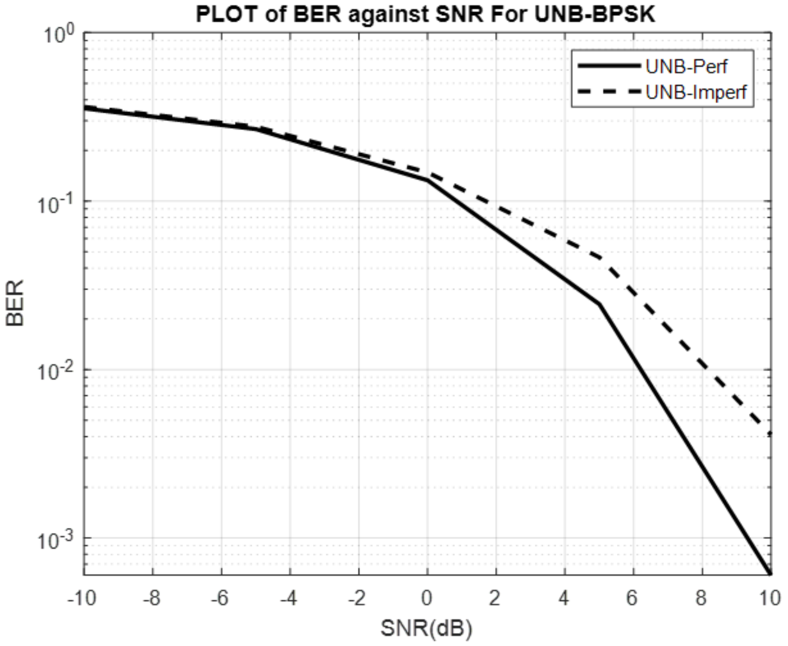

Figure 17. UNB-BPSK Constant Error Simulation.

6.3.2 Random error case

In this section we introduce a random error vector as described in chapter 5 above into the system's receiver terminal and plot the BER for perfect synchronization and imperfect synchronization results for the case of NB and UNB spectrums as shown below. The mean value here was defined as 1, while the standard deviation was defined as 10 for the simulation process, hence the simulation was performed with 10 random values.

The mean error percentage when we have perfect synchronization for the case of NB-BPSK random error (figure 18) is 17.36%, while in the case of imperfect synchronization we have 25.78%, which implies more degradation of the BER is experienced because of poor time synchronization between the transmit signals and the received signals. The total simulation time it took for the transmit signal to synchronize with the receiver 21.67 seconds.

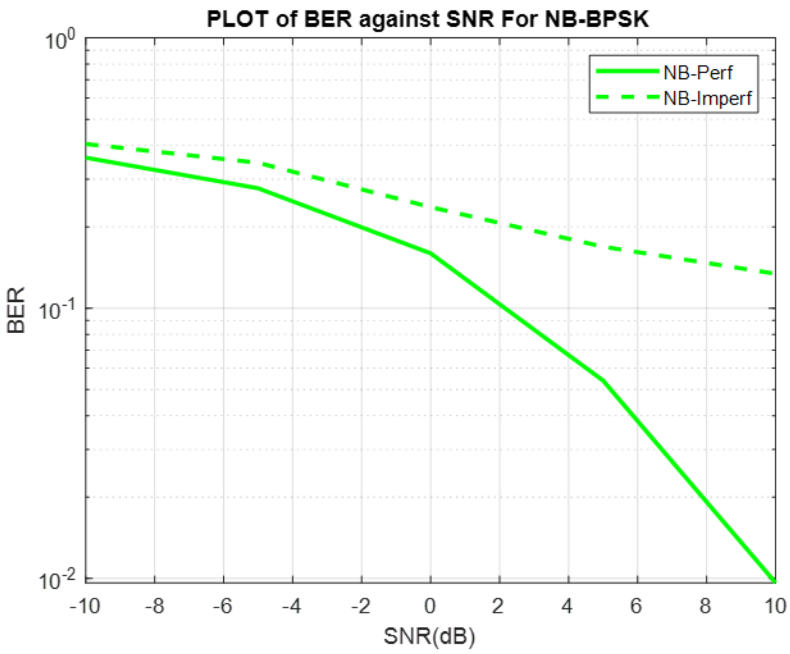

Figure 18. NB-BPSK Random Error Simulation.

For UNB-BPSK random error (figure 19), the percentage error is 17.87% for perfect synchronization and 25.63% for imperfect synchronization. The total simulation time it took for the transmit signal to synchronize with the receiver 21.40 seconds.

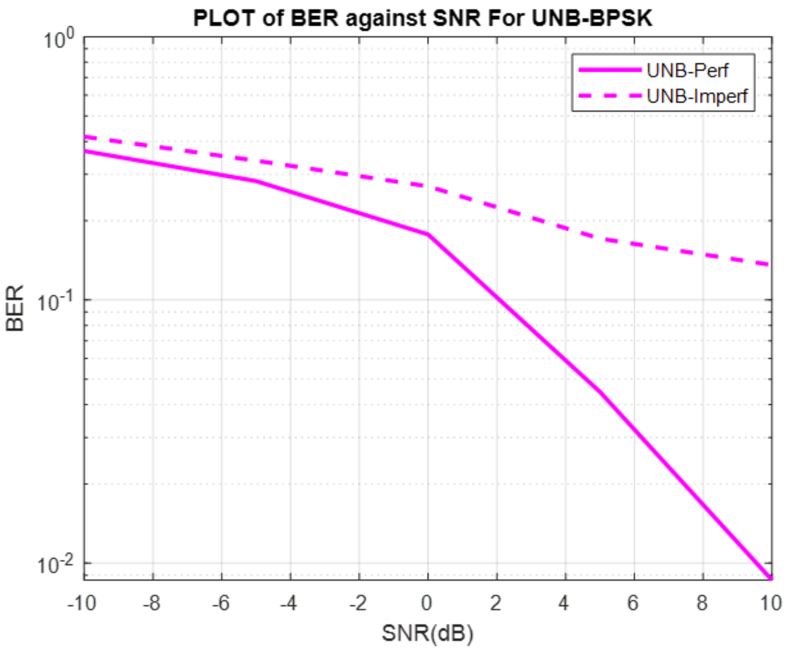

Figure 19. UNB-BPSK Random Error Simulation.

6.3.3 Skew-based clock error model case

In this section we introduce a linear clock error model vector as given in equation 5 in chapter 5 above into the system's receiver and plot the BER for perfect synchronization and imperfect synchronization results for the case of NB and UNB spectrums as shown below.

The mean error percentage when we have perfect synchronization for the case of NB-BPSK clock error (figure 20) is 15.63%, while in the case of imperfect synchronization we have 15.86%, which implies more degradation of the BER is experienced because of poor time synchronization between the transmit signals and the received signals. The total simulation time it took for the transmit signal to synchronize with the receiver 10.98 seconds.

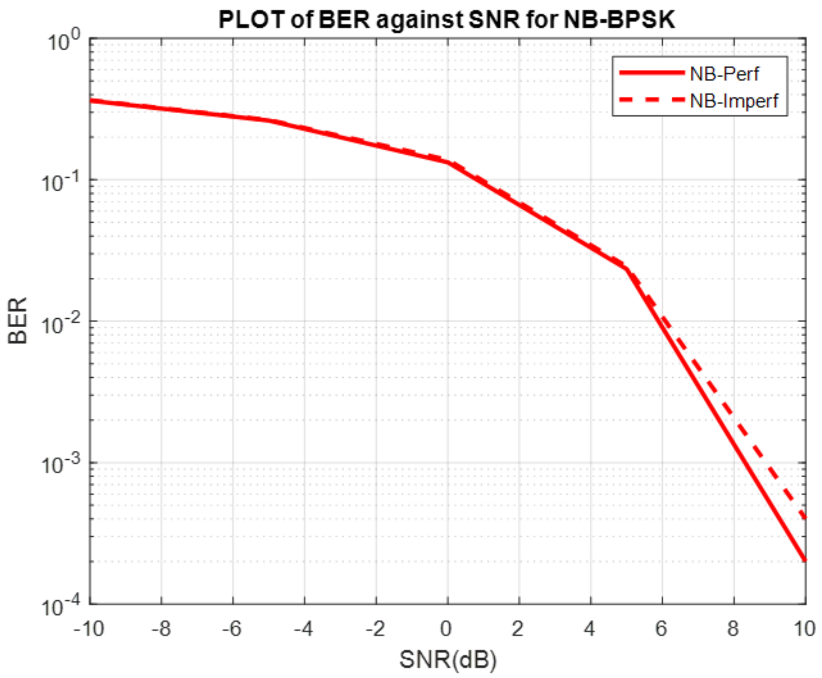

Figure 20. NB-BPSK Clock Error Simulation.

While for UNB-BPSK clock error (figure 21), the percentage error is 15.79% for perfect synchronization and 15.83% for imperfect synchronization. The total simulation time it took for the transmit signal to synchronize with the receiver 10.89 seconds.

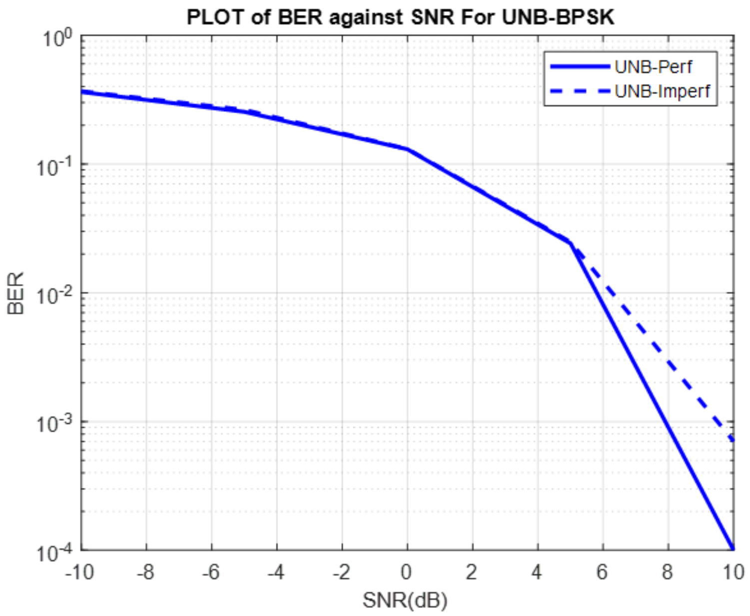

Figure 21. UNB-BPSK Clock Error Simulation.

6.3.4 Comparison between NB and UNB simulations

Here, we made a plot to compare the different spectrums when using the various error models. Figure 22 shows that of constant error, figure 23 shows that of random error and figure 24 shows that of skew-based clock error for both NB and UNB BPSK simulations.

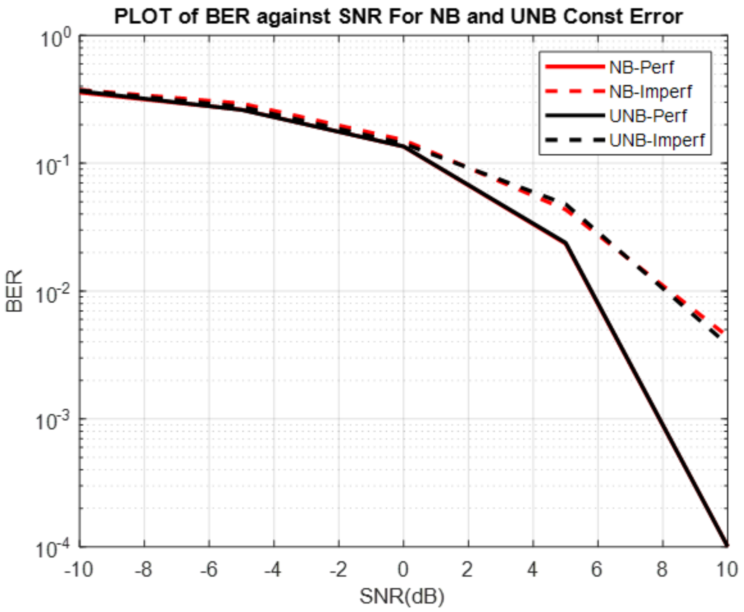

Figure 22. NB and UNB Constant Error Comparison.

In figure 22, the effect of the constant error introduced into the system is same for NB-BPSK and UNB-BPSK perfect synchronization, while in the case of imperfect synchronization a slight difference can be seen. This means NB and UNB has similar performance when the time synchronization is perfect, while UNB is a bit better than NB when the synchronization is imperfect.

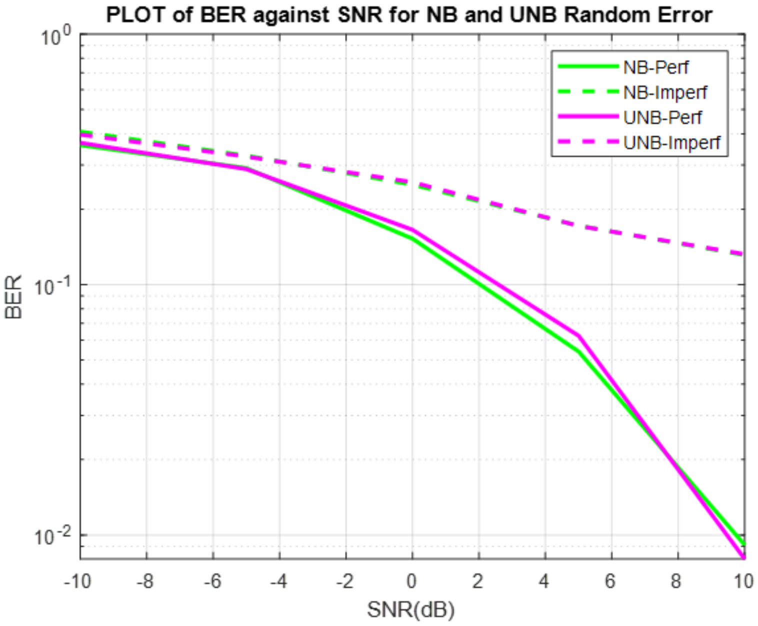

Figure 23. NB and UNB Random Error Comparison.

Figure 23 shows some variance in the curves because the error introduced is random and show more difference in the case of perfect synchronization, while for imperfect synchronization the result is almost the same. Still it can be concluded that UNB has better performance than NB at much lower bits error when the synchronization is perfect, while at higher bits errors NB has better performance. For imperfect synchronization both NB and UNB can be seen to be having same effect.

Looking at figure 24 below, we see how linear clock error behave at high bit error and at low bit error, implying there is almost no difference between the NB and UNB at high bit error, but at lower bit error we start to see some differences. Hence at lower bit errors, UNB has better performance than NB, both under perfect and imperfect time synchronization assumptions.

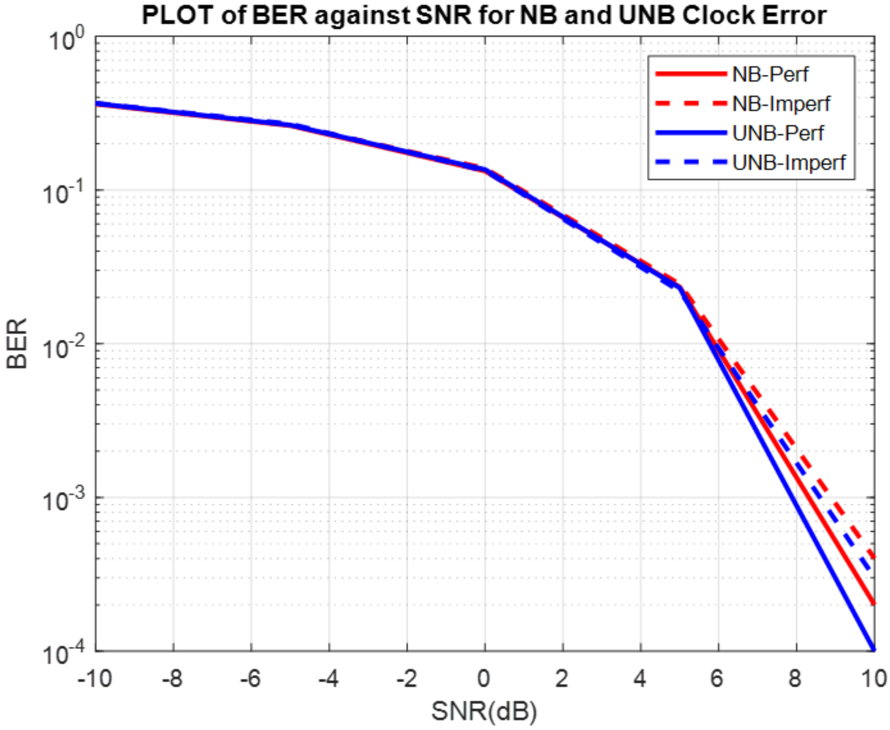

Figure 24. NB and UNB Clock Error Comparison.

6.3.5 Comparison between different error models

Finally, a combined comparison is made for the three different error models as illustrated in figure 25 below. This comparison is made to see which of the models has better performance in an IoT network in situations where there is perfect time synchronization and when the time synchronization is not perfect.

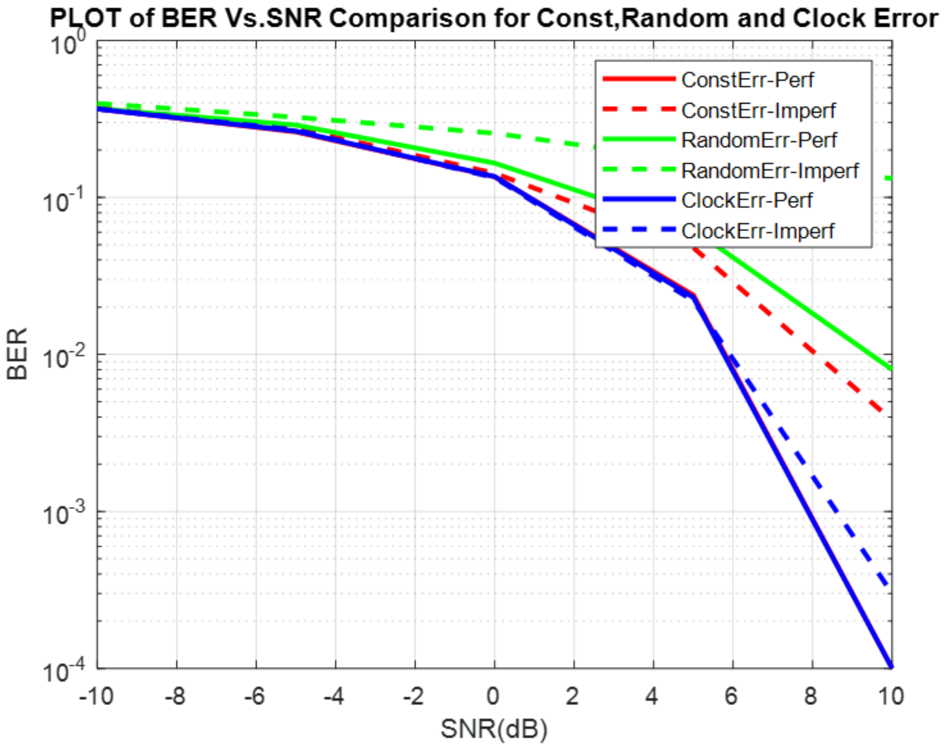

Figure 25. Comparison between Constant, Random and Clock Error Simulations.

Looking at the simulation result, one can see that the skew-based clock error has better performance than both constant error and random error in the case of perfect synchronization and imperfect synchronization, since it takes into consideration the time offset and the frequency drift. This means that it will have better time-sensitivity.

7. CONCLUSION

This thesis work was carried out exclusively on the effect of time synchronization errors on the two IoT spectrum NB and UNB technologies, using three different error models. BER simulation was done assuming BPSK modulation, in order to see which error model may have a better performance in the IoT network. From the research work done, it was clear that for any of these IoT technologies to function properly and effectively, considering different synchronization solutions, they must be time-sensitive and well synchronized.

For the constant error time synchronization, it was demonstrated that NB-BPSK with simulation time of approximately 4s, had 15.58% error when there was perfect synchronization and 17.16% when the synchronization was not perfect, while UNB-BPSK had a simulation time of also approximately 4s and 15.64% error when there was perfect synchronization and 16.62% when the synchronization was not perfect. This means that the effect of the time synchronization with constant error for the result of perfect synchronization of NB-BPSK and UNB-BPSK does not have much difference, similarly to the case of imperfect synchronization with very little difference. This also means that NB and UNB have similar performance when the time synchronization is perfect, while UNB is a bit better than NB when the time synchronization is imperfect. This model has good time-sensitivity.

Taking the random error time synchronization into effect we see that, NB-BPSK with simulation time of approximately 21s, had 17.36% error when there was perfect synchronization and 25.78% when the synchronization was not perfect, while UNB-BPSK had a simulation time of also approximately 21s and 17.87% error when there was perfect synchronization and 25.63% when the synchronization was not perfect. This means that the effect of the time synchronization with random error is so much that it is even difficult to ascertain which of the technology is better since a random value of 10 was used to carry out the simulation. It can be concluded that UNB has better performance than NB at much lower bits error when the synchronization is perfect, while at higher bits errors NB has better performance. For imperfect synchronization both NB and UNB can be seen to be having same effect. This model is less time sensitive as compared with the constant error model.

Under the skew-based clock error time synchronization simulation, it was demonstrated that NB-BPSK with simulation time of approximately 10s, had 15.63% error when there was perfect synchronization and 15.86% when the synchronization was not perfect. The UNB-BPSK had a simulation time of also approximately 10s and 15.79% error when there was perfect synchronization and 15.83% when the synchronization was not perfect. This means that the effect of the time synchronization with clock error is minimal on NB-BPSK and hence has a better performance than UNB-BPSK, while at lower bit errors UNB has better performance than NB in both perfect and imperfect time synchronization.

Finally, after investigating the NB and UNB BPSK in the three different error models, we conclude that time synchronization with clock error will have better performance in an IoT network, because it has little effect in both NB and UNB BPSK modulation, is time-sensitive, reliable and stable in both during perfect synchronization and when the synchronization is not perfect. It is a linear clock that considers the time offset of the system and the frequency drift. Networks that are time sensitive, needs clocks that are perfectly synchronized.

Further research can be carried out to investigate how these error models can be utilized to see the performance of the spread spectrum technique. Also, the effects of frequency and phase synchronization with constant error, random and skew-based clock error can be further research on. Furthermore, mitigation methods to cope with synchronization errors can also be investigated.

REFERENCES

[1] Ericsson white paper, Cellular Networks for Massive IoT, accessed 24 Oct. 2018. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf.

[2] Y.D. Beyene, R. Jantti, K. Ruttik, S. Iraji, On the Performance of Narrow-Band Internet of Things (NB-IoT), 2017 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, pp. 1-6.

[3] SIGFOX, accessed on 24 Oct.,2018. [Online]. Available : https://www.sigfox.com/en.

[4] E. Morin, M. Maman, R. Guizzetti, A. Duda, Comparison of the Device Lifetime in Wireless Networks for the Internet of Things, IEEE Access, Vol. 5, 2017, pp. 7097-7114.

[5] LoRa Alliance, accessed on 24 Oct.,2018. [Online]. Available: https://lora-alliance.org

[6] O.J. Joseph, Ultra-Narrowband Internet-of-Things Technologies, Tampere University Master’s Thesis 2018.

[7] U. Raza, P. Kulkarni, M. Sooriyabandara, Low Power Wide Area Networks: An Overview, IEEE Communications Surveys & Tutorials, Vol. 19, Iss. 2, 2017, pp. 855-873.

[8] GSMA(EC-GSM-IoT), accessed 25 Oct.,2018. [Online]. Available : https://www.gsma.com/iot/extended-coverage-gsm-internet-of-things-ec-gsm-iot

[9] D. Cox, E. Jovanov, A. Milenkovic, Time synchronization for ZigBee networks, Proceedings of the Thirty-Seventh Southeastern Symposium on System Theory, 2005. SSST '05, IEEE, pp. 135-138.

[10] Y.D. Beyene, R. Jantti, O. Tirkkonen, K. Ruttik, S. Iraji, A. Larmo, T. Tirronen, J. Torsner, NB-IoT Technology Overview and Experience from Cloud-RAN Implementation, IEEE Wireless Communications, Vol. 24, Iss. 3, 2017, pp. 26-32.

[11] A comparative study of LPWAN technologies for large-scale IoT deployment, in ICT Express, 2018.

[12] S. Gachhadar, Clock Error Impact on NB-IoT Radio Link Performance, 2018, 61 + 11 p. Available: http://urn.fi/URN:NBN:fi:aalto-201804031969 .

[13] Y.-P. Eric, J. Bergman, J. Sachs, O. Liberg, M. Sundberg Chapter 8 - NB-IoT Performance, in: Cellular Internet of Things, Academic Press, 2018, pp. 297-325.

[14] NB-IoT DEPLOYMENT GUIDE to Basic Feature set Requirements, GSMA Association, 2018.

[15] Narrowband Internet of Things Whitepaper, ROHDE & SCHWARZ, Rohde & Schwarz GmbH & Co. KG Mühldorfstraße 15 | D - 81671 München.

[16] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, A. Ghosh, NB-IoT system for M2M communication, 2016 IEEE Wireless Communications and Networking Conference, IEEE, pp. 1-5.

[17] Todd Walter, Design World 11 Jan.,2016. Accessed 30 Oct.,2018 [Online] Available: https://www.designworldonline.com/its-about-time-the-evolving-time-sensitive-networking-standard-for-the-industrial-iot/.

[18] NI Trend Watch 2016. Accessed 31 Oct.,2018 [Online] Available: http://www.ni.com/pdf/company/en/Trend_Watch_2016_Full.pdf.

[19] Xu Jiu-qiang, Zhao Hai, Gao Wei, Zhu Jian, Wang Jing-shi, Scheduling based on time-sensitivity for wireless sensor network, 2008 Third International Conference on Communications and Networking in China, IEEE, pp. 387-391.

[20] M. Al-Zihad, S.A. Akash, T. Adhikary, M.A. Razzaque, Bandwidth allocation and computation offloading for service specific IoT edge devices, 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), IEEE, pp. 516-519.

[21] Y.-S. Tu and G. Pottie, Coherent cooperative transmission from multiple adjacent antennas to a distant stationary antenna through AWGN channels, in Proc. 2002 IEEE Veh. Technol. Conf. – Spring, vol. 1, pp.130–134.

[22] Z. Ding, W. Chen, and K. Leung, Distributed beamforming and power allocation for cooperative networks, *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1817–1822, 2008.

[23] D.R. Brown III, P. Bidigare, and U. Madhow, Receiver-coordinated distributed transmit beamforming with kinematic tracking, in Proceedings of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2012, to appear.

[24] R. Mudumbai, D.R. Brown III, U. Madhow, and H.V. Poor, Distributed transmit beamforming: Challenges and recent progress, IEEE Communications Magazine, vol. 47, no. 2, pp. 102–110, February 2009.

[25] F. Quitin, M.M.U. Rahman, R. Mudumbai, U. Madhow, A Scalable Architecture for Distributed Transmit Beamforming with Commodity Radios: Design and Proof of Concept, IEEE Transactions on Wireless Communications, Vol. 12, Iss. 3, 2013, pp. 1418-1428. https://ieeexplore.ieee.org/document/6449257.

[26] F. Quitin, M. Mahboob Ur Rahman, R. Mudumbai, U. Madhow, Distributed beamforming with software-defined radios: Frequency synchronization and digital feedback, 2012 IEEE Global Communications Conference (GLOBECOM), IEEE, pp. 4787-4792.

[27] R. Mudumbai, G. Barriac, and U. Madhow, On the feasibility of distributed beamforming in wireless networks, *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1754–1763, May 2007.

[28] R.Marks, L. Yumping,Y. Boyce,Y. Feangchao, Reconsidering Implicit Feedback for Beamforming (IEEE 05.03.2018)Accessed 08 Nov.,2018 [Online] Available: https://mentor.ieee.org/802.11/dcn/18/11-18-0509-00-0wng-reconsidering-implicit-feedback-for-beamforming.pptx.

[29] Jataayu, Implicit Feedback and Explicit Feedback for Transmit Beamforming Accessed 08 Nov.,2018 [Online] Available: http://www.hitchhikersguidetolearning.com. September 17, 2017.

[30] I. Ozil and D. Brown, Time-slotted round-trip carrier synchronization,
in *Proc. 2007 Asilomar Conf. Signals, Syst. Comput.*, pp. 1781–1785.

[31] D. R. Brown III and H. Poor, Time-slotted round-trip carrier synchronization for distributed beamforming, *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5630–5643, Nov. 2008.

[32] S.K. Mani, R. Durairajan, P. Barford, J. Sommers, A System for Clock Synchronization in an Internet of Things, 2018 Accessed 14 Nov., 2018 [Online] Available: https://arxiv.org/pdf/1806.02474.pdf.

[33] R. Preuss and D. R. Brown III, Retrodirective distributed transmit beamforming with two-way source synchronization, in *Proc. 2010 Conf. Inf. Sciences Syst. (CISS).*

[34] R. Preuss and D. R. Brown III, Two-way synchronization for coordinated multicell retrodirective downlink beamforming, *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5415–5427, 2011.

[35] R.S. Jha, P. Gupta, Clock Synchronization in IoT Network Using Cloud Computing, Wireless Personal Communications, Vol. 97, Iss. 4, 2017, pp. 6469-6481.

[36] F. Shamani, Design of A Flexible Timing Synchronization Scheme for Cognitive Radio Applications, Tampere University Master’s Thesis 2013.

[37] T. Chiueh, P. Tsai, I. Lai, Baseband Receiver Design for Wireless MIMO-OFDM Communications, 2nd Edition, 2012, Wiley-IEEE Press, 346 p.

[38] E. Jörgensen, Cell Acquisition and Synchronization for Unlicensed NB-IoT, Linköpings universitet, Kommunikationssystem, 2017 Master thesis.

[39] J. Meng, K. Yang, Physical-layer time synchronization for wireless sensor networks using UWB signals, 2009 Fourth International Conference on Communications and Networking in China, IEEE, pp. 1-5.

[40] W. Sun, F. Brannstrom, E.G. Strom, On clock offset and skew estimation with exponentially distributed delays, 2013 IEEE International Conference on Communications (ICC), IEEE, pp. 1872-1877.

[41] M. Koivisto, A. Hakkarainen, M. Costa, K. Leppanen, M. Valkama, Continuous device positioning and synchronization in 5g dense networks with skewed clocks, 2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, pp. 1-5.

[42] ETSI TS 136 213 V8.8.0 Technical Specification, LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (3GPP TS 36.213 version 8.8.0 Release 8), 2009-10.

[43] D. Dunning, What is Constant Error, accessed 03 April. 2019. [Online]. Available: https://sciencing.com/constant-error-12216420.html

[44] S. Kalla, “Random Error”, accessed 03 April. 2019. [Online]. Available: https://explorable.com/random-error.

[45] M. Valkama, M. Renfors, Lecture notes on Communication Theory and Digital Communications, Tampere University of Technology, 2018.