CONSTRUCTION AND PERFORMANCE EVALUATION OF A LORAWAN TESTBED

by

Jaco Morné Marais

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

November 2018

SUMMARY

CONSTRUCTION AND PERFORMANCE EVALUATION OF A LORAWAN TESTBED

by

Jaco Morné Marais

Supervisor(s):	Prof. Reza Malekian & Dr Adnan Abu-Mahfouz
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Long Range, Long Range Wide Area Network, Low-power Wide Area Network, Internet of Things

The Internet of Things (IoT) is the creation of a large network of interconnected devices that can communicate wirelessly to enable innovation across multiple fields. These thousands of devices require wireless technologies with sufficient communication capability whilst taking their unique requirements such as low power consumption into account. A newly developed wireless technology is the Long Range Wide Area Network (LoRaWAN) protocol and the performance of LoRaWANs is the focus of this work.

Literature revealed that existing performance evaluations were mainly performed over short time periods and with only a few nodes. Additionally, communication was only evaluated over short distances in urban areas and long distance evaluations were mainly performed in rural environments. A research gap, therefore, existed for the evaluation of this technology over long and short distances and over long time periods in an urban environment.

The contributions of this research can be laid out as three objectives. The first was to determine how effective a LoRaWAN would be for nodes at larger distances (> 1.5km) from a gateway in an urban environment. The second was to determine how the Adaptive Data Rate (ADR) scheme impacts the

performance of a LoRaWAN. The final objective was to determine the impact of other LoRaWAN parameters on the performance of a LoRaWAN.

A LoRaWAN testbed was constructed consisting of 18 nodes and one gateway, and several experiments were performed. Each node was sent configuration commands remotely and each experiment was executed until every node sent 1000 packets. Packet delivery in the form of the Packet Delivery Ratio (PDR) was chosen as the performance metric to be used and this was calculated from the ratio of sent versus received packets.

The first objective was to examine LoRaWAN performance from the perspective of long range nodes. This was answered by examining the data from 5 nodes, located at distances between 1.98 km and 5.19 km from the gateway. The results show that a LoRaWAN should only be used if a PDR of between 60 % and 80 % is acceptable for long range nodes. Furthermore, there can be significant differences in the PDRs between nodes and these differences can become bigger over long distances. The data revealed that all aspects, but especially the ADR scheme, should be considered for long range nodes.

The LoRaWAN protocol's ADR scheme was the focus of the second research contribution. This scheme aims to optimise network throughput by adjusting the transmission settings of individual nodes in the network. When the ADR scheme was enabled, the PDR was consistently worse for all three groups (the 18 nodes were divided into three groups based on distance). An examination of the data revealed that, out of 5 possible data rates, the scheme dominantly assigned either the slowest or the fastest data rate. The slower data rates allow for greater range, but the scheme algorithm did not assign these rates to nodes at longer distances, and as a result their performance suffered. The data also revealed that the data rate assignments oscillated between choices, forcing nodes to switch between rates very frequently.

The last research contribution was to determine the impact of other LoRaWAN parameters on the performance of a LoRaWAN. There are numerous parameters and combinations, and thus only the impact of payload length, link checks and waiting time on the PDR were examined and these were found to be not significant. The evaluation of enabling acknowledgements did reveal a significant improvement to the PDR, but whilst this feature works well in small networks (such as the testbed), it would be detrimental in large networks. LoRaWAN gateways can either be in receiving or transmitting mode and gateways are unable to receive packets when transmitting acknowledgements, therefore resulting in an increase in missed packets for a gateway in a large and congested network.

The use of the permanent outdoor LoRaWAN testbed revealed that additional research needs to be done on the algorithms used in the specific ADR scheme tested to improve throughput efficiency. The testbed should be expanded to a multi-gateway configuration to more closely approximate the applications for IoT networks.

LIST OF ABBREVIATIONS

ADR	Adaptive Data Rate
BW	Bandwidth
CBD	Central Business District
CSIR	Council for Scientific and Industrial Research
CSS	Chirp Spread Spectrum
DER	Data Extraction Rate
DR	Data Rate
FEC	Forward Error Correction
FSK	Frequency Shift Keying
IoT	Internet of Things
IQR	interquartile range
ISM	Industrial, Scientific and Medical
MQTT	Message Queuing Telemetry Transport
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LOS	Line of Sight
LPWAN	Low-Power Wide Area Network
M2M	Machine-to-Machine
NM	Network Manager
PCB	Printed Circuit Board
PDR	Packet Delivery Ratio
PER	Packer Error Ratio
PoE	Power over Ethernet
RPMA	Random Phase Multiple Access
RSSI	Received Signal Strength Indicator
SF	Spreading Factor
SNR	Signal to Noise Ratio
Tx	Transmit Power
WSN	Wireless Sensor Network
UP	University of Pretoria

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

The Internet of Things (IoT) aims to improve several aspects of society through the wide scale deployment of low-cost and always-connected devices. By deploying these smart devices, sectors such as agriculture, manufacturing and logistics can operate at higher efficiency levels whilst saving costs [1]. The IoT consists of various types of devices ranging from consumer devices (e.g., a smart watch) to devices monitoring critical infrastructure. The large scale deployment of IoT devices requires new wireless technologies which can support them.

Out of the broader IoT domain, this study will focus on a type of wireless network referred to as a Low-power Wide Area Network (LPWAN). These networks facilitate Machine-to-Machine (M2M) communication and are typically focused on providing connectivity for sensor networks in which low power consumption is critical. These networks must be highly scalable, provide wide coverage and minimise gateway and end device cost [1].

New competitors such as Weightless, Random Phase Multiple Access (RPMA), DASH7, NB-Fi, SigFox and Long Range Wide Area Network (LoRaWAN) all strive for market share in the LPWAN domain. These new technologies have different benefits and drawbacks and a clear understanding of these differences is vital to enable selecting the appropriate technology for each use case. These networks can be studied using analytic models, simulators or testbeds [2]. As this is a new technology, analytic models and simulators are being developed, but permanent testbeds are not as prominent yet.

1.1.2 Research gap

Additional research into the limitations, capabilities and optimal use cases for LoRaWANs is required. These can be evaluated through the construction of testbeds. Temporary evaluations in which researchers drive to several points in a city and briefly record performance are common. Existing researchers have dominantly used one or no gateways and a few nodes in their research. Furthermore, all of the tests were conducted over very brief time periods (hours). More comprehensive performance evaluations using larger testbeds are required to better understand the role of LoRaWAN in the bigger LPWAN domain.

Additionally, the research found in the literature survey had most, if not all, of their nodes in a close proximity to the gateway (less than 1 km) or, alternatively, examined long ranges but in rural settings. The performance of nodes at greater distances in urban environments have not been fully examined, especially over long time periods.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The main objective of the proposed research is to create an external (outdoor) testbed consisting of multiple Long Range (LoRa) devices spread over a large distance, thereby forming a more realistic LoRaWAN. This will allow for a deeper study of the capabilities and limitations of LoRa and LoRaWANs. This testbed will be used to evaluate several metrics to better understand LoRaWANs and reveal issues or strengths that influence the protocol's suitability for IoT use cases. Furthermore, tests can be run over extended periods of time (days).

This work aims to solve the following research questions.

- How effective is a LoRaWAN for nodes at larger distances (> 1.5km) from a gateway in an urban environment?
- How does the Adaptive Data Rate (ADR) scheme impact the performance of a LoRaWAN?
- What are the impact of other LoRaWAN parameters on the performance of a LoRaWAN?

1.3 APPROACH

The first step that was taken in this study was an in-depth background review of the literature on LoRa and LoRaWANs. After this step, and in partnership with the Council for Scientific and Industrial Research (CSIR), a LoRaWAN testbed was created consisting of developed sensor nodes and LoRa gateways. The CSIR was responsible for the hardware elements in the custom LoRa sensor node and the CSIR campus served as the main deployment area for the testbed.

After the hardware had been developed, manufactured and tested, the firmware for the node and a control and management platform for the testbed were developed. Once the entire testbed was functional, multiple evaluations were performed on the network examining aspects such as the impact of varying packet size, sending interval and the ADR scheme on network performance. After the analysis of these evaluations, the insight regarding the capabilities of LoRaWANs was documented.

1.4 RESEARCH GOALS

The primary research goal was to evaluate the LoRaWAN protocol through a performance evaluation of an outdoor LoRaWAN.

1.5 RESEARCH CONTRIBUTION

This evaluation will enable a deeper understanding of the performance that can be expected from an outdoor urban LoRaWAN. A part of the work will focus on the performance of long range nodes, an area that the current literature has not yet fully addressed. Furthermore, the work will enable the identification of possible research gaps and future directions that could be addressed to improve the performance of LoRaWANs.

1.6 RESEARCH OUTPUTS

This work has resulted in a conference paper titled "LoRa and LoRaWAN testbeds: A review" which was presented and published as part of IEEE AFRICON 2017.

A journal article is currently being planned titled "Evaluating LoRaWAN using a permanent outdoor testbed". The contents will focus on the performance impact of the LoRaWAN parameters examined with an emphasis on the ADR scheme. The target journal is IEEE Sensors.

Another research output was the outdoor LoRaWAN. This testbed can be used to evaluate the current version of the LoRaWAN protocol, but can also be used in the future to evaluate new versions of the protocol once released. Researchers are working on improvements to the LoRaWAN protocol and can be evaluated once these has been released to the community. The constructed testbed will also allow other Wireless Sensor Network (WSN) related research topics to be evaluated in addition to the use of simulations.

1.7 DISSERTATION OVERVIEW

Chapter 2 contains a literature study exploring several aspects of a LoRaWAN as well as existing evaluations of this technology. Chapter 3 details the layout of the testbed and how the various parts of testbed were constructed, before describing the experimental methodology that was followed. Chapter 4 contains the results obtained from each experiment followed by initial discussions on each result. A discussion of the research questions based on the experimental work is presented in Chapter 5. Chapter 6 concludes this dissertation by summarising the work done, before addressing each research question and, finally, presenting suggestions for future work.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter contains the findings of a literature study performed on research regarding LoRaWANs. Context is provided describing why this technology was created, before the technology and the standard developed by the LoRa Alliance are explained. This is followed by a discussion on how performance can be evaluated in these networks before the literature relevant to each of the three research questions is presented.

2.2 CONTEXT

The IoT aims to connect devices through the use of M2M communication, allowing devices to exchange information and enabling the automation of tasks [3]. There has been considerable growth in the volume of devices and the revenue potential of the IoT industry, with predictions that the amount of IoT devices will soon exceed the number of consumer electronics currently connected to the Internet [2]. A key reason for this high level of growth is that there are many application areas in which the IoT can be applied. Figure 2.1 shows the various areas in which IoT application can be beneficial.

These vast numbers of devices have unique requirements such as low cost, low energy consumption and wide coverage, which have resulted in new communication technologies being developed for the IoT. As a result, several LPWAN technologies have been developed with competitors battling it out for market share.

Figure 2.1. Different IoT application scenarios. Taken from [2], © 2016 IEEE.

The big players in this sector, namely Sigfox, Weightless, LoRaWAN, RPMA and NB-IoT, have all taken different approaches to address common problems such as scalability, security, firmware updates, geo-location, licensing and power consumption [2]. A comparison between these technologies can be found in [2]. These are all complex issues and, thus, the IoT devices themselves are no longer simple devices, as they require complicated networking stacks with some using embedded operating systems [4].

2.3 THE LORAWAN PROTOCOL

This work set out to conduct a performance evaluation of the LoRaWAN protocol and will thus focus on the literature discussing this technology. A quick note on terminology: a LoRaWAN is a wireless network using the LoRaWAN standard which was developed by the LoRa Alliance to communicate. LoRa is the transmission technology serving as the physical layer for these networks and was developed by Semtech and remains proprietary. Figure 2.2, showcases the LoRaWAN stack and its three layers: Application, LoRaWAN MAC and the LoRa physical layer.

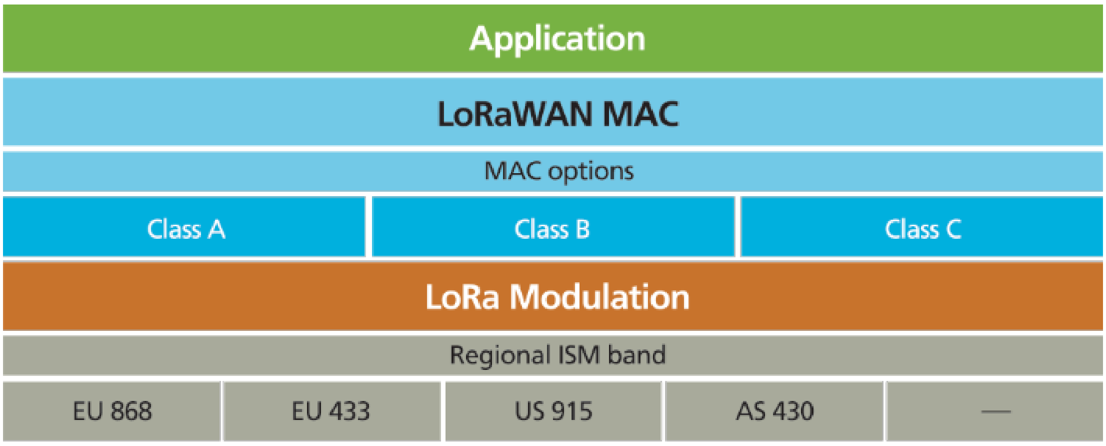


Figure 2.2. The LoRaWAN stack. Taken from [5], © 2017 IEEE.

Whilst the physical layer is closed, the LoRaWAN standard is open and is under active development by commercial and industrial partners [2]. The LoRa Alliance released the first version of the LoRaWAN specification in July 2015. Version 1.1 of the specification, reference [6] of this document, was released in October 2017 and can be downloaded from `https://lora-alliance.org/`.

2.3.1 LoRa

LoRa is a physical layer specification and uses Chirp Spread Spectrum (CSS) modulation and Forward Error Correction (FEC) to mitigate interference. This combination enables long transmission ranges as transmissions down to 19.5 dB below the noise floor can be decoded successfully [7]. LoRa operates in either the 433, 868 or the 915 MHz Industrial, Scientific and Medical (ISM) bands [7]. As the use of these bands differs across the world, the LoRaWAN specification has different regional parameters (defined in [8]) and thus the performance is region-specific. South Africa follows the European regional parameters and allows for operation in the 868 MHz band.

LoRa and Sigfox both occupy the 863-870 MHz ISM band and also share this band with several other devices such as alarm systems and RFID tags [9]. The ambient interference present in 5 typical urban locations on the frequencies of interest for LoRa and Sigfox was examined in [10]. These measurements showed significant activity in the 868 MHz ISM band and a 22-33 % probability of interfering signals above -105 dBm in two of the areas.

Whilst LoRa is a propriety modulation, a LoRa radio can be sourced from different manufacturers [11]. There are four configuration parameters that can be adjusted: Spreading Factor (SF), Bandwidth (BW), carrier frequency and coding rate. The LoRa modulation's chip rate is equal to the bandwidth, i.e. 125 000 chips per second if a bandwidth of 125 kHZ is used [12]. The spreading factor is the ratio between the symbol rate and chip rate and the choice of spreading factor between SF6 up to SF12 provides a trade-off between range and throughput. A higher spreading factor increases the number of chips per symbol, and this decreases the SNR required for demodulation but increases a transmission's duration [13]. Lower spreading factors increase the transmission rate, allowing for more data to be transmitted before the ISM band's duty cycle limits are reached [7, 14]. A key feature of LoRa is that communication on different SFs are orthogonal to each other allowing a gateway to receive them simultaneously. This orthogonality is widely considered to be perfect, but analysis has shown that the symbols are not perfectly orthogonal and that packet loss can occur if the interference power is strong enough [15, 16].

There are three BW options but only two, namely 125 kHz and 250 kHz, are allowed in South Africa. The higher option, 250 kHz, enables higher data rates and is exclusively used alongside SF7 whilst 125 kHz is utilized alongside SF7 to SF12 (SF7 can thus be used together with two options) [12]. The carrier frequency specifies which centre frequency must be used. In the LoRaWAN standard, another option, namely Data Rate (DR) is specified, which is merely a way to specify a combination of SF and BW as one option. In South Africa, there are seven options, namely DR0 to DR7, with DR0 referring to the slowest SF and BW combination (SF12 and 125 kHz) and DR6 referring to the fastest option (SF7 and 250 kHz). The seventh option (DR7) refers to Frequency Shift Keying (FSK) modulation and is rarely used.

The coding rate specifies which FEC rate should be used to counteract bursts of interference, with higher coding rates allowing additional error correction, which increases a transmission's duration (air time) but offers more protection [12]. LoRa has four coding rates namely $\frac{4}{5}$, $\frac{4}{6}$, $\frac{4}{7}$ and $\frac{4}{8}$. LoRa packets have two possible formats namely explicit and implicit. The explicit packet format includes a header, whilst the implicit mode removes the header in order to reduce the transmission time. The header specifies the length of the payload in bytes, the FEC code rate used and if the optional 16 bit Cyclic Redundancy Check (CRC) is present. In the implicit format the header is excluded and these three options must be manually configured at both ends [13]. The header of a LoRa packet will always be transmitted with the maximum code ($\frac{4}{8}$) and the data payload with the coding rate specified in the header.

All of these settings have an impact on performance, with an additional discovered influence being temperature and humidity, which has a potential impact on LoRa's reliability and therefore should be taken into account [17].

Whilst a LoRaWAN will always use LoRa as its PHY layer, the reverse is not the case. The LoRa PHY layer can be used on its own, with a different MAC layer entirely or simply a modified version of the LoRaWAN protocol. This freedom allows for the creation of device-to-device connections [18, 19, 20, 21] or multi-hop networks [7, 22, 23, 24, 25] as well as other modifications. With these modifications, care should be taken to ensure that security threats are not created in the process [26]. There is also a popular industrial LPWAN offering, namely Symphony Link by Linked Labs, that uses LoRa for the physical layer and a custom MAC layer.

LoRa's effectiveness at low power long range communication has attracted the interest of drone researchers which are experimenting with adding LoRaWAN gateways to drones. This allows for ground sensors, with limited energy storage capabilities, to efficiently transmit their data via regular flybys of LoRaWAN equipped drones [27, 28].

2.4 LORAWAN

The LoRaWAN standard uses LoRa as a base and adds additional features required by an LPWAN technology. The standard specifies a star of stars topology in which devices directly communicate with one or more gateways as shown in Figure 2.3. These gateways in turn communicate, possibly over the Internet, with the network server. The network server is responsible for administrative tasks, such as dropping duplicates if a device's message was received by multiple gateways, and will forward messages to the correct application server [29].

The standard defines three device classes, namely A, B or C. All LoRa vendors must implement class A, the most power efficient of the three as devices are unreachable for most of the time. After a Class A device has sent a transmission to the gateway, it opens two short downlink receive windows during

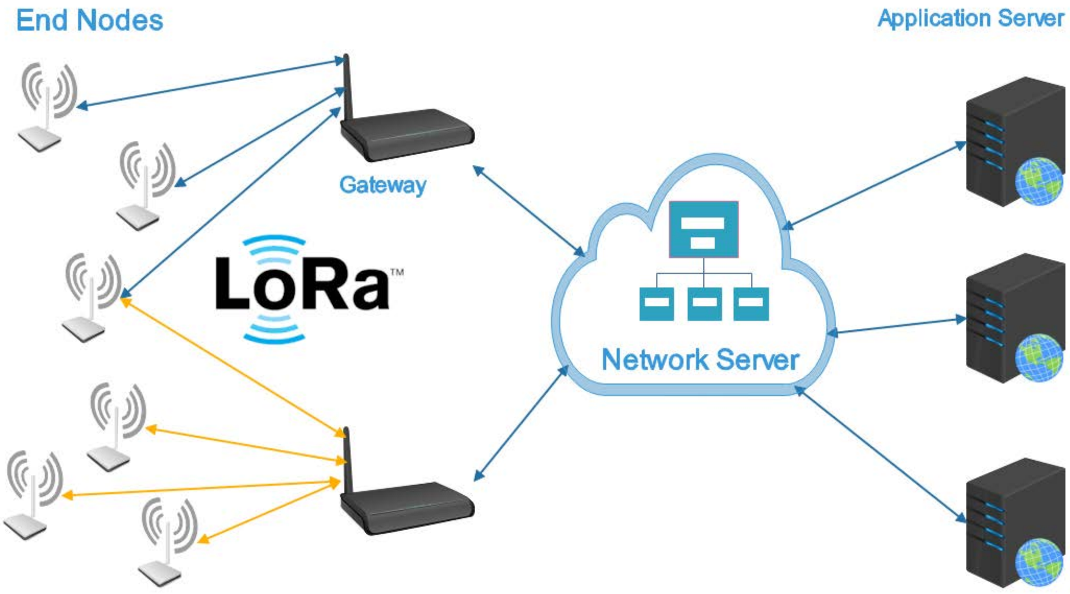


Figure 2.3. LoRaWAN architecture.

which the gateway may transmit data to the device. This is demonstrated in Figure 2.4 and any downlink communication outside of these windows will only be sent after the device's next transmission.

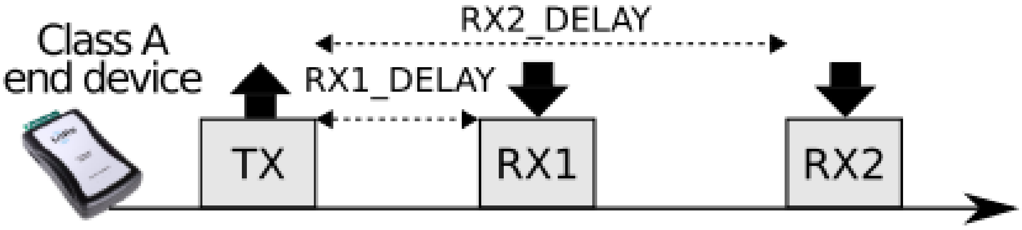


Figure 2.4. Receive slot timing for class A devices. Taken from [14], © 2017 IEEE.

Class B is a modification of class A and adds periodic receive windows, whilst class C keeps a receive window open at all times. Class A devices are very energy efficient but are also the hardest to communicate with as their transceivers would be in sleep mode until a packet must be sent [14]. Other than ensuring duty cycle limits are adhered to, devices may transmit a packet at any time as LoRaWAN follows the ALOHA scheme.

LPWAN requirements such as acknowledgements, firmware updates, localisation, roaming and security are all addressed in the LoRaWAN standard [2, 30]. Acknowledgements are possible but discouraged as a gateway cannot be in the transmission and receiving modes simultaneously. The standard ensures security with separate AES-128 device, network and application encryption keys [31]. Enhanced security protocols have been proposed to improve the security of LoRaWANs and to ensure end-to-end security [32]. The standard is still under active development to improve the solutions to these requirements. A key LoRaWAN feature is the ADR scheme. This scheme aims to maximise battery life and throughput by adjusting the data rate and RF output for every device in a LoRaWAN [6]. This has an additional benefit of increasing network capacity, as messages sent with different SFs are orthogonal and can thus be received simultaneously.

2.5 CONSIDERATIONS WHEN EXAMINING PERFORMANCE

A vital consideration when choosing between LPWAN technologies is performance. Any advantage that low deployment costs offer will quickly be eroded if the technology's coverage and throughput capabilities forces you to deploy more devices than a competing technology. Additionally, any business attempting an IoT deployment will prefer a single technology capable of handling both their outdoor and indoor connectivity needs [33]. Finally, some use cases such as asset tracking or equipment monitoring, require that nodes maintain connectivity even when mobile. It is important that all of these items are considered in addition to the traditional performance metrics of packet delivery, throughput, battery life and signal strength.

Performance evaluations can take place is several forms; they can be simulation based, emulator based or experiments performed either in a lab or in the field. Furthermore, experiments can be temporary and only investigate one performance metric, or permanent installations examining several metrics. Different evaluations are required to examine the performance of indoor versus outdoor and moving versus stationary nodes.

Simulators, emulators and experimental work all have a role to play when conducting performance evaluations. Simulators avoid the delay of first building a testbed and are highly configurable, but may provide overly promising results due to not fully capturing the limitations of firmware running on actual components [34, 35]. Emulators fill the gap between simulators and experiments, but they also cannot fully capture the radio frequency environment of a deployed testbed [34]. Testbeds are costly, take time to build and are not as configurable as simulations, yet they remain a popular and accurate method for evaluating wireless networks [34, 35].

Simulators for LoRaWANs have been developed, with popular simulator being LoRaSim [29, 36], with extensions added in [37, 38, 39]. Ns-3 is another popular WSN simulator and ns-3 modules have also been developed for this platform [14]. The simulators have been used to examine the ADR scheme [37, 5, 40], network capacity [14] or interference [36]. Mathematical evaluations are also used to provide insights into energy consumption [41], network capacity [42, 43, 44], scalability [45, 46, 47, 48] and throughput [49, 50, 51].

Packet delivery remains a popular evaluation metric, either in the form of examining an individual node's Packet Delivery Ratio (PDR), its inverse Packet Error Ratio (PER), or by examining the network as a whole with Data Extraction Rate (DER) to capture overall network behaviour. When computing the packet loss, care must be taken to ensure sufficient data points are being used. For example, in [52], the packet error is calculated for several zones. Zone A's calculation has a PER of 63.26 % (756/1195) whilst zone D has a PER of 20.34 % (24/118). However, there is an order of magnitude difference in the number of packets used to calculate zone A's PER versus zone D. If 1 more packet had been dropped, this would influence the PER with 0.087 % in zone A versus a change of 0.85 % in zone D.

2.6 PERFORMANCE EVALUATIONS

LoRaWAN is a popular research area with new work on various topics being produced. An earlier version of this chapter was published as a conference paper (see [53]) in which Table 2.1 appeared. Since this table was published, LoRaWAN research has increased in popularity, adding many additional data, so much so that producing a new version of this table would have been a lengthy exercise. From the initial survey shown in this table the wide variation in system parameters used in the various performance evaluations was already evident. Additionally, not all required information is supplied in

Table 2.1. Testbeds found in literature. Modified from [53], © 2017 IEEE.

Ref.	Gate-ways	Nodes	Tx power (dBm)	SF	BW (kHz)	ISM band (MHz)	RSSI/ SNR meas-ured?	Distances covered (km)	Payload (bytes)	How was reliability measured?
[54]	1	1	20	12	125	-	Yes	7 floors	large	RSSI values
[55]	3	2	14	varied	-	868	Yes	2.2	seq. number	Record ACKs
[7]	0	6	17	varied	varied	-	-	0.342	varied	Packet reception rate
[56]	1	1	-	varied	-	-	-	2	-	SF needed for coverage
[33]	1	1	14	varied	125	868	Yes	0.5 m to 60 m	varied	Number of lost packets and packet error
[57]	1	1	-	10	250	868	Yes	0.276 to 8.52	10, 50 and 100	PER
[58]	0	7	-	-	-	915	Yes	0.5 to 2.7	26	% valid packets received
[59]	1	1	14	12	125	868	Yes	15	seq. number, status	% packets lost.
[60]	1	1	14	12	125	868	-	65 m to 195 m	-	% received packets
[61]	0	2	3	varied	-	2450	-	9.75 outdoor, 30 m indoor	21	% valid packets
[62]	1	1	2/14	several	125	-	Yes	3.4 outdoor, 100 m indoor	1, 25, 51	% packets received and avg. throughput (bytes/s)

the published research which complicates comparisons. Finally, even the metric with which reliability was measured differed between researchers.

The abundance of literature led to the choice to discuss only the work that is directly relevant to the research questions. It is common for a paper to discuss several focus areas with varying depth, thus some papers will be discussed under multiple research questions.

2.6.1 Performance of nodes at larger distances

By design, a LoRaWAN can support communication over considerable distances and does not require line of sight between the gateway and a node. In rural or line of sight conditions, a properly mounted LoRaWAN gateway can cover much larger distances than a gateway deployed in an urban environment.

The impact of line of sight was presented in [59], and a PDR of 62 % was achieved via a boat for line of sight distances of between 15-30 km. This was much higher than the PDR of 26 % achieved for car tests of distances between 10-15 km in Oulu, Finland. This dramatic decrease was in part caused by the fact that the packets were sent whilst mobile. Follow-up research found that, when speeds of 40 km/h are exceeded the communication performance worsens due to the Doppler effect [63, 64].

A LoRaWAN is certainly a strong candidate for rural or flat areas as shown in [57]. A gateway was placed on a building close to a river and performance measurements all along the river showed a 100 % PDR for distances up to 7.5 km when a payload of 10 bytes was used.

When performance evaluations were conducted in urban, suburban and rural environments, coverage was achieved for around 6 km in urban and suburban areas with over 18 km achieved in the rural scenario [64]. Their urban evaluation, which enabled acknowledgements, showed a PDR of 100 % for DR0 to DR5 for distances below 3 km, although over how many packets this was calculated was not specified. Even at distances between 5 km and 6 km, a 100 % PDR achieved when DR0 was used, with the other data rates resulting in PDRs of between 30 % and 50 %.

A LoRaWAN's coverage was examined in Paris in which nodes were placed in locations ranging from 650 m to 3.4 km away from a gateway situated in a house [62]. One hundred packets were sent in each test, and when SF12 was examined, a PDR of 100 % was achieved for 650 m and 1.4 km, nearly 100% at 2.3 km, around 90 % at 2.8 km and just below 40 % at 3.4 km. Alongside SF12, SF7 and SF9 were also tested at these distances and especially SF7 showed a significantly worse PDR, quickly dropping from around 80 % at 650 m to 50 % at 2.3 km before remaining 0 % for further distances.

Antenna choice is especially important for long range nodes as demonstrated in [65], where an internal antenna limited connectivity to only 600 m in Melbourne's Central Business District (CBD). It should be noted that, as this was conducted in Australia, other regional parameters were used which also impacted transmissions.

2.6.2 Performance impact of the ADR scheme

The ADR scheme is defined in the LoRaWAN standard and specifies how a gateway can use LinkADRReq messages to specify new transmission settings to devices. The scheme will adjust a device's DR, Transmit (Tx) power, repetition rate (number of times to repeat a frame) and list of allowed channels. The default repetition rate is to only send a frame once and a device will reply to a LinkADRReq request with a LinkADRAns command to confirm each requested change [6, 66]. The scheme has a mechanism to compensate for too high data rate assignments, and a node will adjust its data rate lower if its periodic connectivity checks fail to receive a response from a gateway [66]. The specification does not stipulate how the network server should calculate the optimum data rate and transmit power for each device; this is left to the device manufacturer or developer [12]. With this freedom, the standard's goal of optimising the network to use the fastest data rates available can be changed to instead optimise for congestion or PDR and not only for throughput.

To ensure scalability, one suggested approach is that congestion be estimated by examining the network's throughput, RSSI and number of connections at a gateway before nodes are sent LinkADRReq messages [67]. Fair Adaptive Data Rate (FADR) uses RSSI values in its calculations when determining SF and Tx power assignments. Additionally, it assigns SFs to groups of 50 nodes, which were created by ordering the nodes based on RSSI [37]. Another approach increases the ADR algorithm's complexity by suggesting additions such as adjusting data rates before incrementing Tx power, the averaging of SNR history and accounting for hysteresis [66]. A contention-aware ADR approach is proposed in [40], which tracks the number of nodes per SF and aims to increase the number of devices using low SFs in order to maximise the network's throughput.

Two algorithms are proposed in [5], which aim to maximise network throughput and the data extraction rate, a measure similar to PDR but which focuses on the network as a whole instead of on individual node performances. An expansion on this work is presented in [68], in which two new schemes are proposed to improve performance. The first scheme, EXPLoRa-KM, targets the SF allocation of overcrowded areas whilst the second, EXPLoRA-TS, aims to equalise traffic loads between SF channels.

The ADR scheme, as implemented in the Things Network, was evaluated using OMNeT++ in [69]. Simulations found that whilst the ADR scheme is effective in stable channel conditions, performance suffers when used with highly variable channels. A solution, named ADR+ was developed and outperformed the standard scheme in highly variables channel conditions. A second proposed ADR scheme, which takes device locations into account are also presented. This scheme is especially helpful in densely populated areas as the current ADR scheme assigns SFs based on SNR values, which could be very similar for dense nodes. As a result, all nodes in the area would be assigned the same SF and thus the orthogonality features of LoRa are not taken advantage of.

Additionally, experiments found that the ADR scheme needs to configure transmission parameters by talking the whole network into account. As an example, in dense networks with a lot of nodes, assigning the fastest data rate to all nodes is not beneficial as it will increase the chance of collisions.

A scheme optimised for the PER experienced by long range nodes is proposed in [70]. An improvement of nearly 50 % was achieved for these nodes with the overall network seeing an improvement of 42 %. The improvement is achieved by taking into account the insights gathered from deriving the optimum spreading factor distribution for LoRaWAN's 5 discrete power levels.

2.6.3 Performance impact of other LoRaWAN parameters

A LoRaWAN's performance is not only influenced by aspects such as communication range or the ADR scheme, but also by factors stemming from the use case: packet size, sending interval and the use of acknowledgements. Additionally, as the testbed uses Multitech mDots, an additional parameter, namely link checks, can also have an influence. Link checks are a mechanism provided by Multitech's mDot library that allows a node to periodically test if it is still connected to a gateway. This mechanism is similar to the one used by the ADR scheme but merely informs the user application that the link connectivity tests failed. This link check mechanism appears to be unique to Multitech's offering as no performance evaluations of this feature were found in the existing literature.

When the impact of 2 different packet sizes on the PDR was examined in [64], it was found that a definitive PDR improvement was attained for shorter packet lengths. The improvement is not consistent with results varying from less than 2 % to more than 20 % depending on the data rates used and the scenario (urban, suburban or rural). When different packet sizes were examined in a more open setting (next to a river) in [57], similar results were achieved for a payload of 10 or 100 bytes but payloads of 50 bytes performed significantly worse. As part of an experiment on average throughput per spreading factor, payload sizes of 1, 25 and 51 bytes were tested in [62]. The distance from the gateway is unknown but the results, created using 100 packets per test, showed that lower spreading factors resulted in higher throughput. Long distance nodes would use the highest spreading factor, and their results for SF12 showed that decreasing the payload from 51 bytes to 25 bytes resulted in a throughput decrease from 0.38 bytes/s to 0.26 bytes/s. The researchers note that the LoRaWAN standard was not designed for high throughput, but rather to support a large number of devices.

The default behaviour in a LoRaWAN is to send all packets as unconfirmed frames, but support is provided for confirmed frames for which the receiving party will transmit an acknowledgement. An end device may transmit an acknowledgement immediately following a message from the gateway or defer the acknowledgement to be sent alongside its next data message [6]. When acknowledgements were requested by nodes in an evaluation of a three gateway LoRaWAN, it was noted that in 2.5 % of cases the data arrived but the device did not receive an acknowledgement and this could result in unnecessary retries [55].

To investigate the impact of downlink traffic in which acknowledgements materially influence performance, the popular LoRaSim simulator was extended into LoRaWANSim [39]. When several scenarios were evaluated, it was found that, as network size increases, a gateway will reach its duty cycle limits when attempting to transmit all of the required acknowledgements. Simulations also revealed that the protocol assumes that failure to receive ACKs was as a result of using a too high data rate and did not consider the possibility that the message was received but the gateway could not transmit an ACK due to duty cycle restrictions. Additionally, as a gateway cannot transmit and receive simultaneously, it will not receive a message when transmitting an acknowledgement, causing those messages to be retransmitted [39]. The use of acknowledgements has a major impact on performance in large networks and greatly reduce their capacity [39].

When acknowledgements were evaluated in a single gateway network, it was found that acknowledgements only improved the PDR for a low number of devices (100, 500 and 1000) and only when data was sent every 60 thousand seconds [14]. For a 100 device network, an improvement was also observed for a 6 thousand second data sending interval, but in general, if the number of devices increased or the sending interval decreased, the PDR was lower. In networks with two or four gateways, the PDR is improved compared to a one gateway network as more acknowledgements can be sent as duty cycle restrictions apply per gateway and not per network [14].

The sending interval defines the time period between transmission, and this was evaluated using ns-3 using three time periods, namely 600 s, 6 000 s and 60 000 s [14]. These were compared against each other for multiple numbers of end devices. For small networks of less than 500 nodes, the difference in the PDR is small, but in a network of 5000 devices, the differences between 600 s and 6 000 s was nearly 8 percentage points with a drop of nearly 23 percentage points between 6 000 s and 60 000 s [14]. As the sending frequency and the number of devices increases, more and more packets arrive at the gateway during an existing transmission, and if their SFs are the same, the packet is dropped.

2.7 CHAPTER SUMMARY

This chapter detailed the results of a literature review performed on the LoRaWAN literature. The review starts by providing context on why LoRaWANs are needed by examining the IoT domain and the various use cases which require a suitable M2M communication solution.

The LoRaWAN standard is still under active development by the LoRa Alliance and uses LoRa, developed by Semtech, as the physical layer. Unlike other LPWAN solutions, the LoRaWAN standard is open and is thus popular in the research community which contributes. LoRa uses a combination of CSS modulation and FEC to provide the interference mitigation required for long range communication. The LoRaWAN standard is built on top of LoRa and specifies a star of stars network topology and three devices classes. A key LoRaWAN feature is the ADR scheme, which controls transmission settings such as the data rate and transmit power of individual nodes in the network in order to optimise throughput.

Performance evaluations of LoRaWAN can take the form of mathematical models, simulations or experiments in either laboratory conditions or field deployments. The literature contained various performance metrics such as PDR, PER, DER, throughput or energy. An earlier literature review resulted in Table 2.1 and demonstrated that performance evaluations had a wide variety of settings, thus complicating performance comparisons.

The findings of several research projects on the performance of long range LoRaWAN nodes were examined and the substantial impact of rural versus urban areas was identified. Rural evaluations demonstrated coverage at distances longer than 15 km versus urban areas in which coverage varied widely from less than 1 km to nearly 8 km. At longer ranges, the coverage differences between spreading factors become more evident, with SF12 as the only suitable choice for long range nodes.

The ADR scheme is a popular topic in LoRaWAN research as this scheme's operation has a large impact on a network's throughput, scalability and PDR. The current ADR scheme has the sole objective of optimising network throughput, and therefore researchers have suggested other algorithms aimed at optimising other performance aspects. The performance impact of other factors such as packet size, sending interval, acknowledgements and link checks were also investigated. Whilst some researchers found that smaller packet sizes improved the PDR, other experiments did not come to the same conclusion. Packet sizes are an important part of raising a network's throughput, but choosing the correct spreading factor can also have a big impact on throughput.

In small networks of less than 500 devices, the impact of the sending interval on PDR is minimal, but in larger networks, the increased frequency of packet collisions at the gateway results in a reduced PDR. The Multitech mDot library allows link checks to be performed by nodes. The literature searched did not yield examples of performance evaluations of this feature.

Communications in a LoRaWAN are normally performed using unconfirmed frames but the standard does provide acknowledgement (confirmed frames) functionality. When nodes require acknowledgements for their uplinks, a gateway must enter transmission mode to send an ACK, causing it to miss incoming transmissions during this time. Through simulation, the impact of acknowledgements on networks of various sizes was investigated. Enabling ACKs can improve the PDR for small networks of less than 1000 nodes, but as the number of devices increases and the sending interval decreases, the PDR drops dramatically. A LoRaWAN gateway must adhere to duty cycle limits and may therefore not be able to transmit all ACKs that are requested. Adding more gateways helps to alleviate this problem but the use of acknowledgements should be carefully managed in order to avoid congestion.

The insights gathered from the literature review were used during the development of an outdoor LoRaWAN testbed in an urban environment. The development process is presented in the next chapter.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter details the design and implementation of a LoRaWAN testbed. Design considerations and a description of the testbed's location will be followed by details on the various hardware and software elements of the testbed. Finally, the experimental methodology is discussed.

3.2 DESIGN CONSIDERATIONS

The literature review in Chapter 2 revealed that there is a need to perform evaluations of the LoRaWAN protocol using a permanently deployed network. By deploying nodes at fixed distances, one can provide insights on the impact of range on performance, without requiring that node(s) be constantly relocated. Additionally, permanently deploying multiple nodes allows for continuous simultaneous data collection.

In order to deploy an outdoor LoRaWAN testbed successfully, several hardware and software requirements must be met. Nodes will require a rugged rain proof enclosure and a convenient way to keep the batteries charged. A robust but configurable mounting system will allow nodes to be easily placed in different environments. Finally, a visual indication (e.g., LEDs) that the node is working will make maintenance easier.

On the software side, the testbed's software should be robust and configurable. The software should allow for experiments to be configured remotely as visiting each node to modify its configuration will be time consuming. Another requirement is a monitoring system providing a visual indication of the testbed's status and individual node performance. Finally, a form of data logging is required to capture incoming packets and their meta-data during experiments.

Due to cost considerations, an initial testbed size of 20 nodes was set as a goal. This number will keep costs low, whilst allowing for various distances from the gateway to be evaluated. As some nodes were kept for testing and servicing purposes, the final testbed consists of 18 nodes.

3.3 TESTBED LOCATION

The testbed is geographically spread over 3 locations. The majority (13) of the nodes are dispersed over the Pretoria campus of the CSIR. Two nodes are located nearly 2 km away at a residence and three nodes were placed on rooftops approximately 5 km away at the University of Pretoria (UP). Two nodes are located inside whilst the rest are outside. Figure 3.1 shows the testbed's locations.

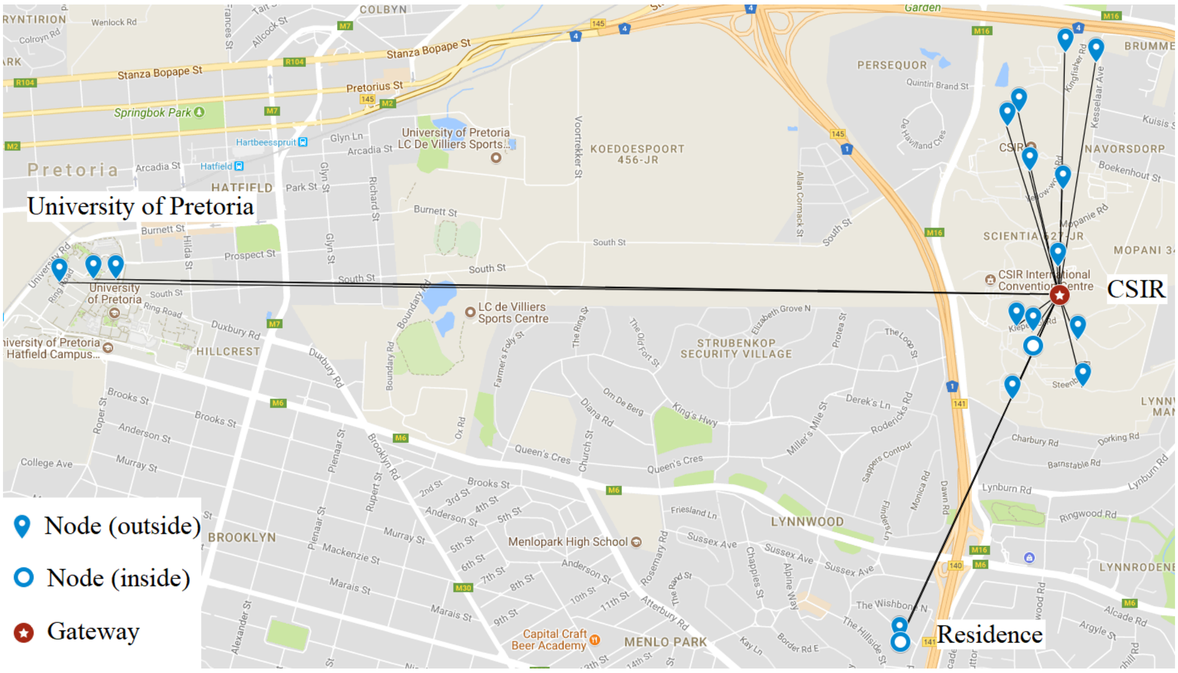


Figure 3.1. Map displaying location of nodes.

On the CSIR's campus there are two LoRaWAN gateways, one owned by the CSIR and another by Comsol [1]. The CSIR's gateway is dedicated for LoRaWAN based research and the Comsol gateway services all of Comsol's IoT clients in the area.

---

[1] A last-mile access provider (`https://comsol.co.za/`).

The testbed is one of two LoRaWAN projects on the CSIR's Pretoria campus. These two projects connect to the same LoRaWAN network as they share the CSIR's gateway. The second project, monitoring water usage, consists of 34 nodes attached to water meters throughout the campus. Each node sends measurements every 10 minutes and the ADR scheme is enabled. Out of the 34 nodes, one does not connect to the CSIR's gateway but instead connects to the Comsol gateway. Figure 3.2 shows the Multitech LoRaWAN gateway used by the testbed. The gateway is located on the highest hill on the campus but is not in Line of Sight (LOS) for most nodes due to buildings, terrain changes and plant growth.

Figure 3.2. Gateway at the CSIR.

The distance between the gateway and each node is shown in Table 3.1. The distances are approximates as they were calculated by Google's My Maps service during the creation of Figure 3.1. The nodes can be grouped into three groups based on distance. The "Near" group consists of the 6 closest nodes, all within 0.5 km of the gateway. The "Far" group consists of the next 7 nodes with distances between 0.54 km and 1.25 km. Finally, the "Furthest" group consists of 5 nodes with distances from 1.98 km to 5.19 km. The groups differ in size as the groups were chosen to better investigate the impact of distance on performance.

Table 3.1. Straight line distances between the 18 nodes and the gateway.

Node	Distance (km)	Location	Group
e5-43	0.14	CSIR	Near
eb-94	0.24	CSIR	Near
eb-96	0.25	CSIR	Near
e5-3f	0.28	CSIR	Near
a6-59	0.30	CSIR (indoors)	Near
e5-47	0.50	CSIR	Near
eb-9a	0.54	CSIR	Far
eb-98	0.60	CSIR	Far
e5-45	0.66	CSIR	Far
e5-48	0.91	CSIR	Far
e5-46	0.97	CSIR	Far
e5-44	1.22	CSIR	Far
e5-4b	1.25	CSIR	Far
e5-40	1.98	Residence (indoors)	Furthest
eb-97	1.98	Residence	Furthest
e5-4a	4.90	UP	Furthest
eb-99	5.02	UP	Furthest
e5-42	5.19	UP	Furthest

3.4 HARDWARE

In this section, the hardware components of the nodes are described. In the design, assembly and testing of the nodes the author of this dissertation was only consulted with, as these were the responsibility of a CSIR employee.

The LoRaWAN offerings of several manufacturers were considered before Multitech's mDot module was chosen as the basis for the testbed's nodes. The main decision criteria was ease of development, on both a hardware and software basis. The mDot is in a higher price class than some of its competitors, but Multitech offers excellent technical documentation, firmware examples and support for their products. Not only is the mDot LoRaWAN 1.0.2 compliant but firmware for the STM32F411RET microprocessor can be written using the Arm Mbed platform. Using a certified LoRaWAN device was key to ensuring that the performance of the LoRaWAN protocol can be accurately measured.

The mDots were ordered with already soldered headers and an SMA connector, allowing for easy interfacing with the node's Printed Circuit Board (PCB) and Multitech's recommended 1/4 wavelength whip antenna providing 3.0 dBi gain. Each mDot has a unique 64 bit end-device identifier (DevEUI) assigned by the chip manufacturer.

A 3.7 V 1000 mAh Lithium Polymer rechargeable battery serves as the power source for each node. This battery ensures the mDot's low-dropout 3.3 V regulator is catered for. The battery is recharged by a 60 x 140 mm 1 W 6V solar panel via a Linear Technologies LTC4070 battery charging IC.

A custom PCB was designed for the testbed and, in addition to the mDot and the LTC4070, a DS1621 temperature sensor and two LEDs were added. A female header provides a convenient interface for serial communication with the mDot. The mDot slides into a header interface, allowing it to be removed if needed. The assembled PCB can be seen in Figure 3.3.

The PCB and battery were placed inside an IP 67 rated enclosure and attached to an adjustable pole mounting system. The solar panel was glued to the mounting system and its orientation can be adjusted for maximum sun exposure. A deployed node is shown in Figure 3.4. Hardware schematics and a block diagram are presented in Appendix A.

Figure 3.3. The PCB developed for the testbed.

3.4.1 Gateway considerations

Several options were considered when a gateway was selected for the testbed. Gateways can be purchased off the shelf from vendors such as Multitech, Kerlink or Cisco, or created by combining a Raspberry Pi with, for example, the WiMOD iC880A LoRaWAN concentrator. Based on cost, customer support and configurability, a Multitech MultiConnect Conduit running mLinux was purchased. Multitech provides indoor and outdoor gateways; the outdoor gateways are the same product as the indoor gateway, but are placed in an appropriate enclosure and can be powered using Power over Ethernet (PoE). Indoor gateways use a Pulse Electronics W1063 3dBi antenna whilst the outdoor gateway uses a L-com Global Connectivity HGV-906U 6 dBi antenna. It should be noted that this testbed uses only one gateway due to their high costs. As only one gateway is used, the network server runs locally on the gateway. In multiple gateway setups, a packet would be considered received by the centralised network server if received by at least one gateway.

Figure 3.4. Node deployed at the CSIR.

3.5 SOFTWARE

3.5.1 Overview

An overview of the software elements of the testbed is shown in Figure 3.5. Firmware executing on the mDots communicates with the gateway which in turn relays packets to the web server. On the web server, the packets are first processed by a Node-RED application before being stored in an InfluxDB database. The Node-RED application also provides a dashboard to control the testbed. Collected data can be viewed and monitored with the Grafana platform.

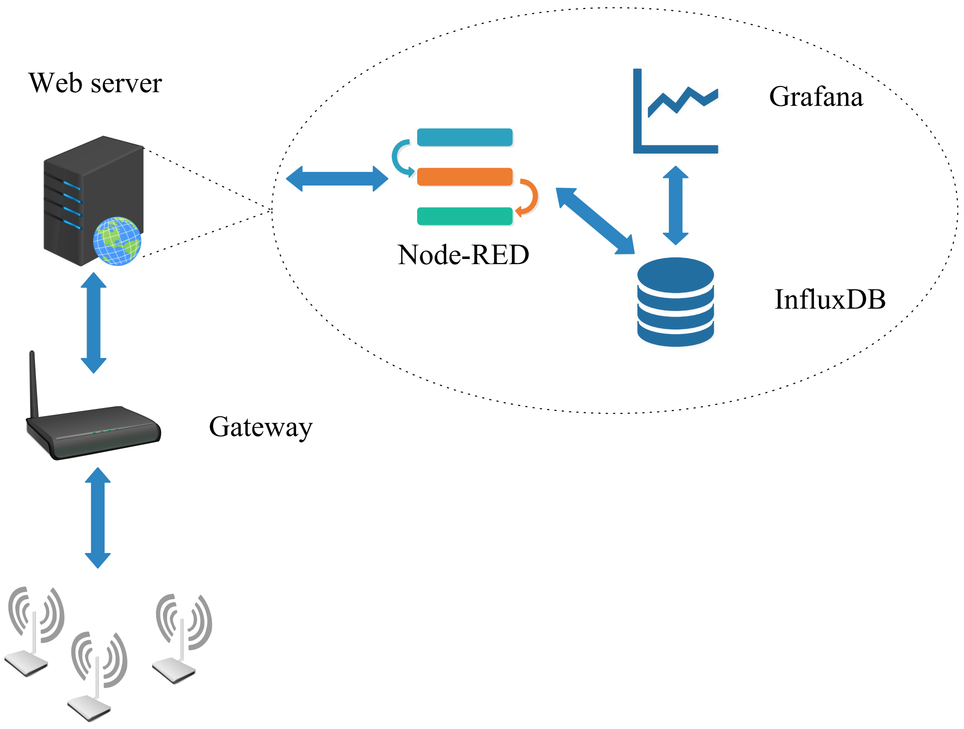


Figure 3.5. Testbed's software components.

3.5.2 Firmware

The firmware was written in C++ using the ARM Mbed platform. This platform provides an online development platform with version control and private repositories for project sharing. Multitech's mDot library provides a convenient method to configure settings or to receive and send packets. Any settings, such as the coding rate, not modified through remote configuration remained at the defaults specified by Multitech in their mDot library version 3.0.0 (revision 62:255e2ddc294e). In the case of the coding rate, the coding rate is thus $\frac{4}{5}$.

Remote configuration can be performed by sending each node a set of instructions. These instructions allow nodes to be reconfigured remotely and were based on the popular AT command style. A subset of the command set is shown in Table 3.2, and the complete set can be found in Appendix B.1.

Table 3.2 shows some of the available commands. Commands can query values (AT+WT?), set values (AT+WT=5) or be instructions (AT+RS). Commands can be sent individually or can be chained by

Table 3.2. Node command set.

Command	Purpose	Options	Valid values
AT+AD	Set/Disable adaptive data rate	? or =	0 or 1
AT+CG	Query node config	?	
AT+DR	Set/Query data rate	? or =	0-5
AT+PL	Query max payload size	?	num of bytes, depends on SF
AT+PS	Set/Query payload size	? or =	size in bytes
AT+RS	Restart mDot		
AT+WT	Set/Query inter packet waiting time	? or =	in seconds

separating with semicolons, for example "AT+PS=20;AT+DR=2;AT+WT?". Instead of querying each modified setting individually, AT+CG can be used to query all settings.

For example, an AT+CG might return "AC0;LC45;TX11;AN3;AD1;WT60;PS45;DR0;". This indicates that requesting packet acknowledgements are disabled (AC), link connectivity checks (LC) are set to Multitech's defaults, transmit power (TX) is at the default of 11 dBm, antenna gain (AN) is set to the default of 3 dBm, adaptive data rate (AD) is enabled, the waiting time between packets (WT) is set to 60 s, application payload length (PS) is 45 bytes and the data rate should the ADR scheme later be disabled would be DR0.

The firmware contains a watchdog timer which will reset the mDot if not "fed" within its 25 second countdown window. As the nodes are in remote locations, this feature allows them to reset themselves should any part of the firmware become unresponsive. The mDot is also equipped with a bootloader, allowing new firmware to be flashed over the serial connection.

To simulate sensor readings, each node will send a payload of the size specified by AT+PS roughly every x seconds, where x is the amount of time specified by the AT+WT command. The mDots are not equipped with real time clocks and their timing is also not synchronised. This is to mimic common IoT use cases in which regular updates are required, but their timing does not have to be extremely precise. The payload itself consists of randomly generated bytes. Whilst the payload is normally random, for specific events, a node will send status messages as its first three messages instead of random bytes. These messages are: STARTED when powered on, REJOIN if successfully rejoined a network and STARTEDWD if the watchdog triggered a reset. These strings have a number appending to them, either 1, 2 or 3 to indicate this is the first, second or third REJOIN/STARTED/STARTEDWD. Additionally, PAC2B is used to warn the user that the node attempted to send packets longer than the legal limit for the current SF.

Appendix B.1 contains a flowchart demonstrating how the command string is processed and how payloads are generated.

3.5.3 Gateway

The Multitech MultiConnect Conduit Gateway runs version 3.3.9 of mLinux, an open source embedded Linux distribution, and runs Multitech's network server version 1.0.36-r1.0. The gateway is equipped with a single Multitech LoRa Accessory mCard (MTAC-LORA-H-868), which can support thousands of mDots [71]. This gateway is connected to the CSIR's internal network and runs the LoRa Network Server and a Message Queuing Telemetry Transport (MQTT) broker. The gateway is only accessible on the CSIR's internal network and thus a Node.js script acts as a bridge for any MQTT communication between the gateway and the public-facing Node-RED application.

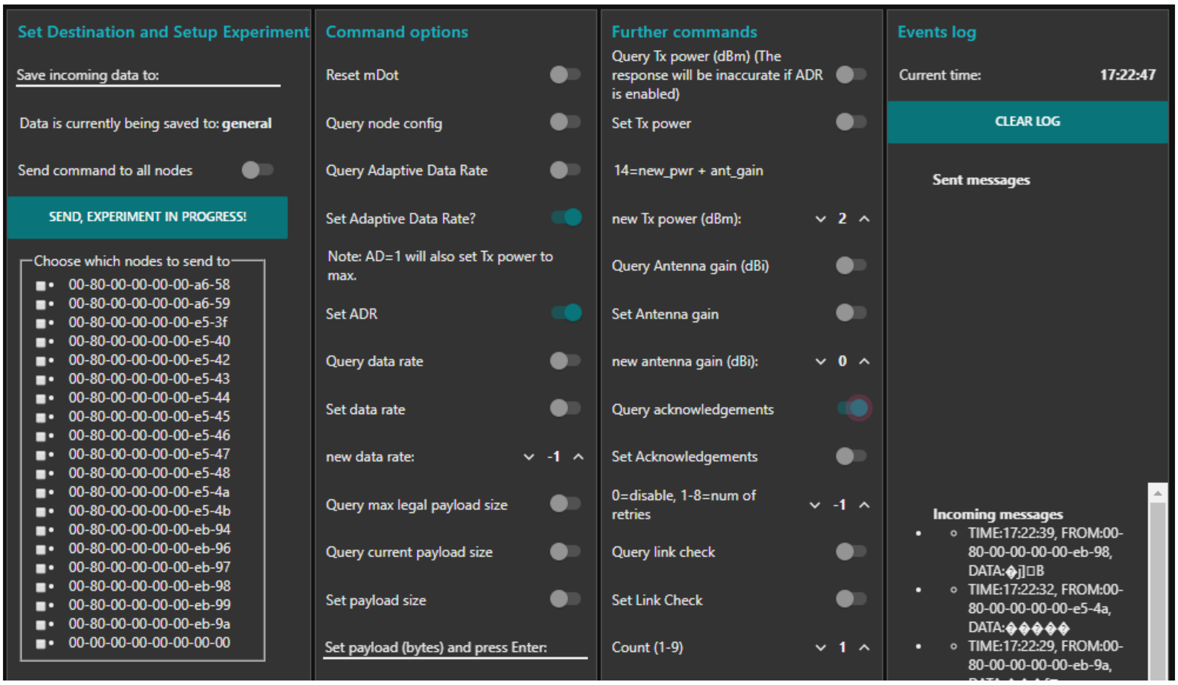
Packets are exchanged with the gateway by subscribing or publishing to several MQTT topics. Packets sent from a node can be received by subscribing to "lora/<devid>/up", where <devid> is replaced by the node's device id (e.g., 00:80:00:00:00:00:a6:59) . Sending a packet to a node is performed by publishing to "lora/<devid>/down". Packets received or destined to be sent are in JSON format and the "data" field is encoded as Base64.

Packets received by subscribing to a node's /up topic contain several JSON fields which include: data (base64 encoded), timestamp, Signal to Noise Ratio (SNR), Received Signal Strength Indicator (RSSI), SF used, packet sequence number and several others. Downstream packets also have several possible JSON fields: data (base64 encoded), ack (request an acknowledgement) and an application port number.

3.5.4 Node-RED

A Node-RED application running on the web server is responsible for communicating with the gateway, saving incoming data packets, generating alerts and providing an interface to configure nodes.

Figure 3.6 shows parts of the configuration interface for the testbed; this was built using the npm module node-red-dashboard. The user can specify which commands must be sent to either all nodes or specific nodes. The user can also specify where incoming packets must be saved to in the database.



Save incoming data to:

Data is currently being saved to: **general**

Send command to all nodes ○

Choose which nodes to send to

- ☐ 00-80-00-00-00-00-a6-58
- ☐ 00-80-00-00-00-00-a6-59
- ☐ 00-80-00-00-00-00-e5-3f
- ☐ 00-80-00-00-00-00-e5-40
- ☐ 00-80-00-00-00-00-e5-42
- ☐ 00-80-00-00-00-00-e5-43
- ☐ 00-80-00-00-00-00-e5-44
- ☐ 00-80-00-00-00-00-e5-45
- ☐ 00-80-00-00-00-00-e5-46
- ☐ 00-80-00-00-00-00-e5-47
- ☐ 00-80-00-00-00-00-e5-48
- ☐ 00-80-00-00-00-00-e5-4a
- ☐ 00-80-00-00-00-00-e5-4b
- ☐ 00-80-00-00-00-00-eb-94
- ☐ 00-80-00-00-00-00-eb-96
- ☐ 00-80-00-00-00-00-eb-97
- ☐ 00-80-00-00-00-00-eb-98
- ☐ 00-80-00-00-00-00-eb-99
- ☐ 00-80-00-00-00-00-eb-9a
- ☐ 00-00-00-00-00-00-00-00

Reset mDot ○

Query node config ○

Query Adaptive Data Rate ○

Set Adaptive Data Rate? ●

Note: AD=1 will also set Tx power to max.

Set ADR ●

Query data rate ○

Set data rate ○

new data rate: -1

Query max legal payload size ○

Query current payload size ○

Set payload size ○

Set payload (bytes) and press Enter:

Query Tx power (dBm) (The response will be inaccurate if ADR is enabled) ○

Set Tx power ○

14=new_pwr + ant_gain

new Tx power (dBm): 2

Query Antenna gain (dBi) ○

Set Antenna gain ○

new antenna gain (dBi): 0

Query acknowledgements ●

Set Acknowledgements ○

0=disable, 1-8=num of retries -1

Query link check ○

Set Link Check ○

Count (1-9) 1

Current time: **17:22:47**

```
TIME:17:22:39, FROM:00-80-00-00-00-00-eb-98, DATA:�j]□B
TIME:17:22:32, FROM:00-80-00-00-00-00-e5-4a, DATA:�����
TIME:17:22:29, FROM:00-80-00-00-00-00-eb-9a,
```

Figure 3.6. Dashboard to control testbed's nodes.

Messages are forwarded via MQTT from the gateway to the Node-RED application and the base64 payload is decoded. The decoded payload is examined to determine if a STARTED or STARTEDWD payload was sent. When these messages are detected, a tweet is sent via a npm package providing access to the Twitter API. This alerts the user that a node has reverted back to its default configuration, and reconfiguration to match the current experiment is required.

When decoding a node's base64 encoded payload care, must be taken to ensure it is decoded to the right character set. Base64 encoding is used to ensure binary data can be reliably transferred between systems as the encoded text will only contain ASCII characters. Table 3.3 shows how the same Base64 string will differ in length when decoded into ASCII versus UTF-8.

Table 3.3. Decoded length differences when decoding a Base64 string into ACII vs. UTF-8.

Base64 string	**ASCII length**	**UTF-8 length**
BUAzR/kvkufW6npvy3nerqfSugc=	20	18
HAL64eaqQTOc00K7BCKXlgLauIU=	20	19
vMR7Il9LxG+ge5NBEaz6XOAlmNw=	20	20

Initially, the Node-RED application was set to decode into UTF-8 and, as a result, the captured payloads of the first few experiments differ from the expected lengths. However, the payload itself was the correct length, and the experiment's meta-data is still valid. An incorrectly decoded payload was saved, but as the payload was randomly generated this is not a problem.

3.5.5 InfluxDB

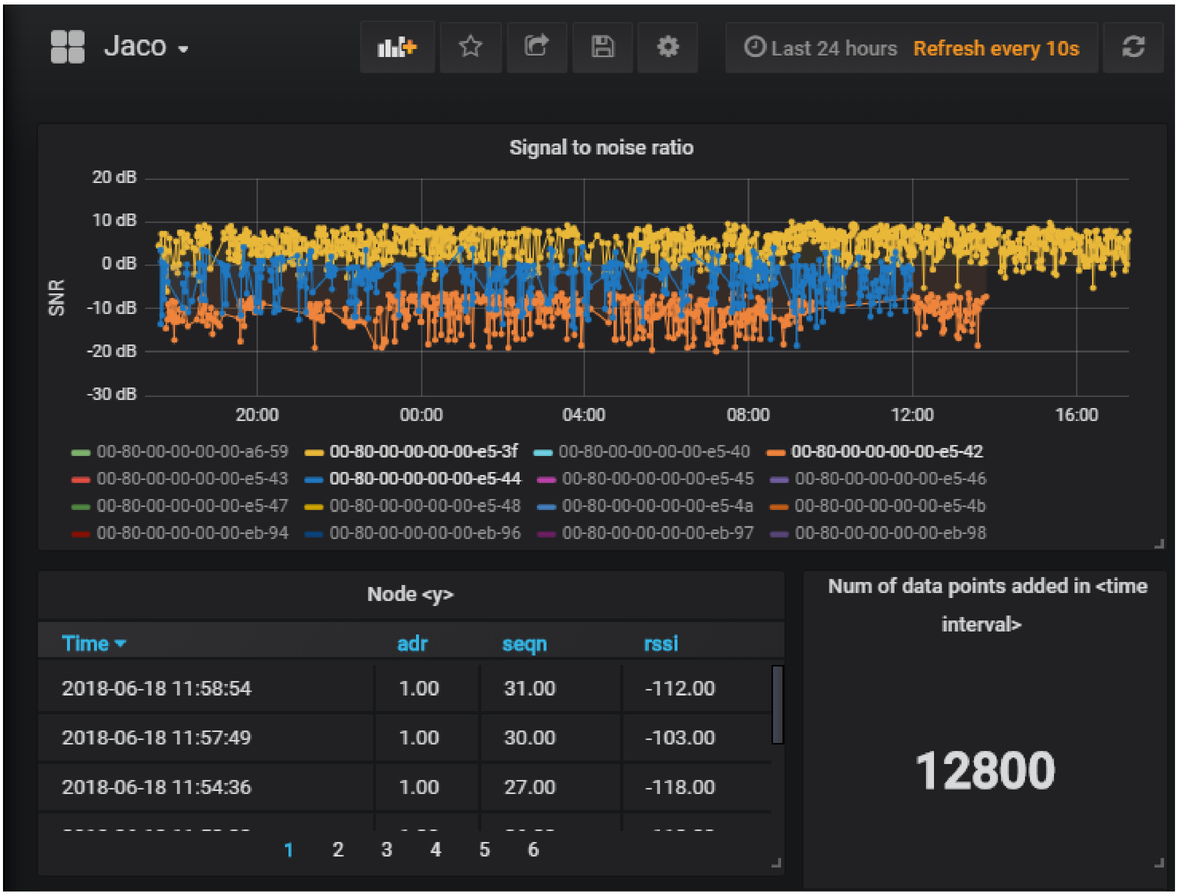
The decoded data field and meta-data of the incoming data packets are stored in an InfluxDB database. InfluxDB is an open-source time series database developed by InfluxData, optimised for IoT and real-time applications.

Interfacing with the database is done using either InfluxQL (a SQL-like language available through a command line tool), a Node-RED node, Grafana or with a Node.js Client. The command line tool is used for configuration and creating backups. The Node.js client is used to extract data for analysis. The testbed currently uses version 1.4.2 of InfluxDB.

Each experiment performed with the testbed consists of several sub-experiments, with each data set stored separately. Each sub-experiment has a “measurement", a container similar to a “table" in a SQL database. Several fields from the JSON meta-data about a packet is stored alongside the time and decoded data string. These fields include: RSSI, SNR, deveui, sequence number, frequency, channel, data rate etc.

3.5.6 Grafana

Grafana, an open source visualisation tool, is used as a handy visual way to monitor the current state of the testbed. Grafana provides a convenient method to view the stored meta-data of each node for a specified time interval. Figure 3.7 shows the Grafana dashboard. The dashboard consists of multiple panels which are configured to run database queries and display the results either as a graph, table or single value.



Node <y>

Time	adr	seqn	rssi
2018-06-18 11:58:54	1.00	31.00	-112.00
2018-06-18 11:57:49	1.00	30.00	-103.00
2018-06-18 11:54:36	1.00	27.00	-118.00

Num of data points added in <time interval>

12800

Figure 3.7. Grafana's dynamic dashboard.

3.5.7 Data analysis scripts

The data gathered for each experiment is queried and processed using Node.js before being graphed using a Jupyter Notebook. The data captured in each measurement is queried using the InfluxDB Node.js client and the PDR is calculated using the packet sequence numbers. The PDR is an indication of what percentage of data packets were successfully received over a period of time. Every packet is given a sequence number by the node, and when plotted in Grafana appears as shown in Figure 3.8.

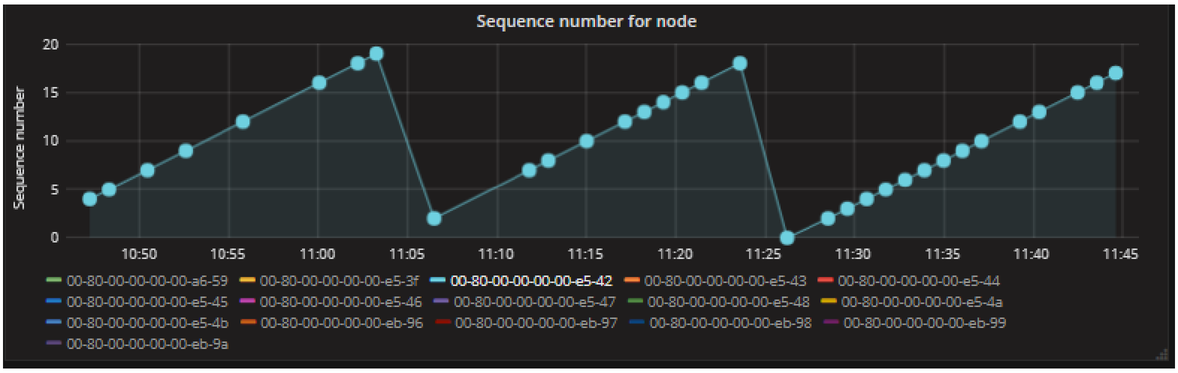


Figure 3.8. Packet sequence numbers in Grafana.

In Figure 3.8 it can be seen that the sequence numbers for packets sent from this node increase linearly, with occasional resets back to zero. The gaps between dots indicate that a packet(s) was sent but not received. The resets back to zero are either due to the node resetting or because the node has reconnected to the network. When a node reconnects it resets the sequence number but still has the same configuration required by any experiment in progress.

The Node.js processing script breaks the data collected from a measurement into arrays for each node and calculates the number of packets sent and the number received. A sequence number series of "5,6,8" indicates that four packets were sent and only three were received. The script also handles reconnection events and can detect if, for example, a sequence is "2,3,4" that the node sent five packets ("0" and "1" wasn't received).

The script processes the array until a specified total number of packets had been sent and records how many hours each node took to reach the threshold. The Node.js scripts then save a summary for each node in a text file, which is read by a Jupyter Notebook document. Jupyter Notebook is an open source web application which allows you to combine explanatory text, code and its results in one document. Notebooks are used for further data formatting of the text files and drawing various graphs.

Algorithm 1 shows how the sequence numbers are processed in order to calculate each node's PDR. The first step is to split each node's array of sequence numbers into separate smaller arrays whenever a reconnection or disconnect event is detected. These events are identified by examining the sequence numbers for continuity, that is whenever the number sequence restarts at 0, a disconnect has occurred. The first packet after the reset (number 0) can be lost and thus the algorithm should not only look for zeros. As an example, a sequence array of "5, 6, 8, 0, 1, 2, 3, 1, 2, 3, 4" would be split into "5, 6, 8", "0, 1, 2, 3" and "1, 2, 3, 4". These subsets are then processed individually and a minus operation between successive numbers is used to detect missing packets. For example when "5, 6, 8" is processed $8-6=2$, and thus the missing packet variables will record that 1 packet was not received.

This process is repeated for all subset arrays with the total number of missing packets being updated after each array. The device's PDR can then be calculated by dividing the total number of received packets (the length of the seqn array) by the total number of sent packets.

Algorithm 1 Pseudocode for PDR calculations.

```
for each device in the testbed do
    Loop through an array of this device's recorded packet sequence numbers and identify reconnection and disconnection events by doing the following:
    for number in this device's seqn array do
        if number > the next number then
            Record that the array must be split at the index for the next number
        end if
    end for
    Split seqn array into multiple smaller arrays according to the recorded index values

    Process each subset array as follows:
    for number in the current subset array do
        if next number - current number is not 1 then
            missing packets = next number - current number -1
        end if
    end for
    update variable tracking total missing packets for device

    After all subsets have been processed
    total sent = length of seqn + total missing packets
    total received = length of seqn array
    device's PDR = total received / total sent packets
end for
```

3.6 EXPERIMENTAL METHODOLOGY

This testbed was built with the aim of obtaining a quantitative understanding of how a LoRaWAN would perform. An exhaustive performance evaluation is impractical as wireless networks are known for highly variable performance in normal environments versus laboratory conditions [72]. Additionally, acceptable performance will depend on the particular use case. Nevertheless, given the important role these networks will play in the future, any insights gathered regarding performance in an operational context will be valuable.

The testbed, and by extension the protocol, can be evaluated using several performance metrics. While scalability and energy consumption are important metrics, the main performance metric was chosen as packet delivery. The PDR was chosen as the main performance metric, as the main function of a sensor network would be the relay of sensor data.

3.6.1 PDR

Equation 3.1 shows how each node's PDR (%) can be calculated. This calculation is performed over either a block of time or a target for the number of sent data packets as the answer depends on the number of packets recorded for each node.

$$\frac{received\ data\ packets}{sent\ data\ packets} * 100 \tag{3.1}$$

When a node has disconnected from the network and is attempting to reconnect it will send join-request packets. Some of these packets might not be successfully received by the gateway and thus a device might send multiple join-request messages. Equation 3.1 defines the PDR as being focused on data packets and thus does not reflect in any way the success ratio of join-requests.

An alternative definition of PDR would be to include all packets and not just data packets. When Equation 3.1 is performed on the testbed's data, there will be a small error when compared to the alternative definition. The reason that join requests are not included in Equation 3.1 is that they would have to be tracked manually on the node, as the gateway would not be aware of any unsuccessful join requests and only starts logging a node once it has successfully joined.

Each node in the testbed can be in one of three states, shown below in Figure 3.9. The number of sent data packets can only be calculated during the normal state, as it is unknown for the misconfigured state and zero for the reconnecting state. During the reconnecting state, the node attempts to reconnect to the network by following the over-the-air activation procedure. During this procedure, a node will make multiple attempts to send join-request messages. As these messages are not data packets they are not stored in the database.

In the misconfigured state, the node has disconnected from the network and during its reconnection attempts misidentified itself with a DevEUI composed of zeros. The gateway accepts the node, but as the DevEUI does not match a known DevEUI, these packets are not forwarded to the web server. A node will remain in this state until it disconnects and reconnects with its correct DevEUI. This misconfiguration problem was only identified after several months and was brought to Multitech's attention and is still under investigation [2]. This state is very rare, and since its identification and subsequent tracking from April 2018 has not occurred again. Its influence on data collected prior to April 2018 is unknown but due to its nonoccurrence since April it is unlikely to have occurred often in the previous data.

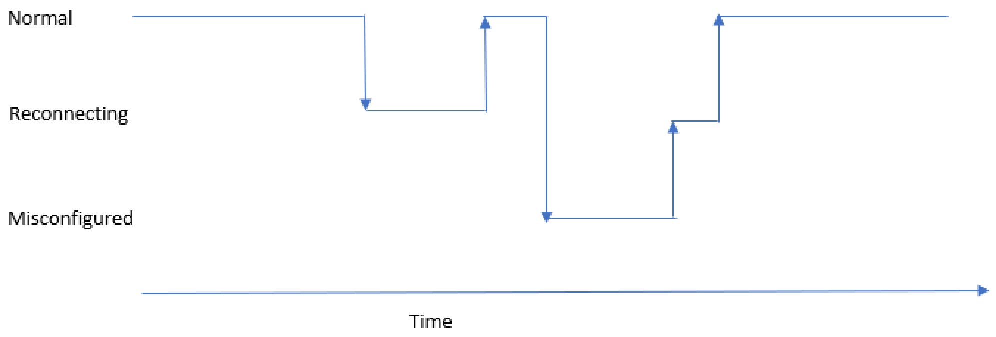


Figure 3.9. State diagram showing a node's states.

In the normal state, each data packet successfully received by the gateway is stored in a database and each packet has a sequence number that is incremented by the node with each sent packet. Equation 3.1 requires the number of sent packets to be known; however, this is only possible in the normal state. Therefore, the calculated PDRs can only provide an estimate of performance. This estimate should be fairly accurate as nodes spend the majority of their time in the normal state.

A test was conducted to confirm that nodes spend the majority of their time in the normal state. In the test, all nodes were configured to each send 1 packet every 60 seconds and all packets were saved in a measurement. The number of hours it took each node to reach 1000 was calculated and it would have taken a minimum of 16.666 hours to send all 1000 packets (implying 100 % normal state). The nodes spent an average of 81.33 % of their time in the normal state, with a worst case of 43.99 % and a best case of 92 % of their time. This means that on averages nodes took 19.77 hours to send 1000 data

---

[2] Opened support ticket with Multitech on 26 April 2018 (ticket ID: #5088879)

packets, and the node that had the least amount of disconnections took 18 hours. The measurement's data was recorded in March 2018 (before any tracking of the misconfiguration state), thus the split between the reconnecting and misconfigured states are unknown.

Each experiment consists of a set of configurations, for example, examining the packet size consists of logging data with packet sizes of 5 to 50 in increments of 5. Each configuration will be stored in its own InfluxDB measurement. It was decided that configuration will be maintained until each node has sent 1000 packets, with one packet sent every 60 seconds. An alternative approach was considered in which data is captured for a set amount of hours. The PDR could then be calculated over the total period or per hour and then averaged to obtain the final PDR.

The per hour averaged approach was rejected due to the impact of periodical disconnects and reconnects as shown in Table 3.4. The node was disconnected from the network for the majority of hour four but managed to reconnect and send two packets. An average of averages error would occur when the average PDR for hours one to six would be calculated by simply averaging the PDR column. Additionally, when the data is split into hourly blocks and processed independently, this type of situation frequently occurs with the packet sequence numbers: "50, 51 ; 60, 61" (; indicates a new hour). If this is processed as hour one (50, 51) and hour two (60, 61) the missing packets 52-59 wouldn't be detected, causing the calculated PDR to be inaccurate.

Table 3.4. A reconnection's impact on the calculated PDR.

Hour	**Received packets**	**Total packets**	**PDR (%)**
1	56	94	59.57
2	57	88	64.77
3	36	81	44.44
4	2	2	100.00
5	74	102	72.55
6	73	94	77.66

Calculating the PDR over the total period eliminates these problems, but small variations in the amount of sent packets per node occur, which results in different totals used for the denominator in the PDR calculations. All nodes are programmed to transmit with the same sending interval but reconnections and clock drift due to temperature or manufacturing differences result in different sent packet totals. Therefore, calculating the PDR over the total period was also rejected.

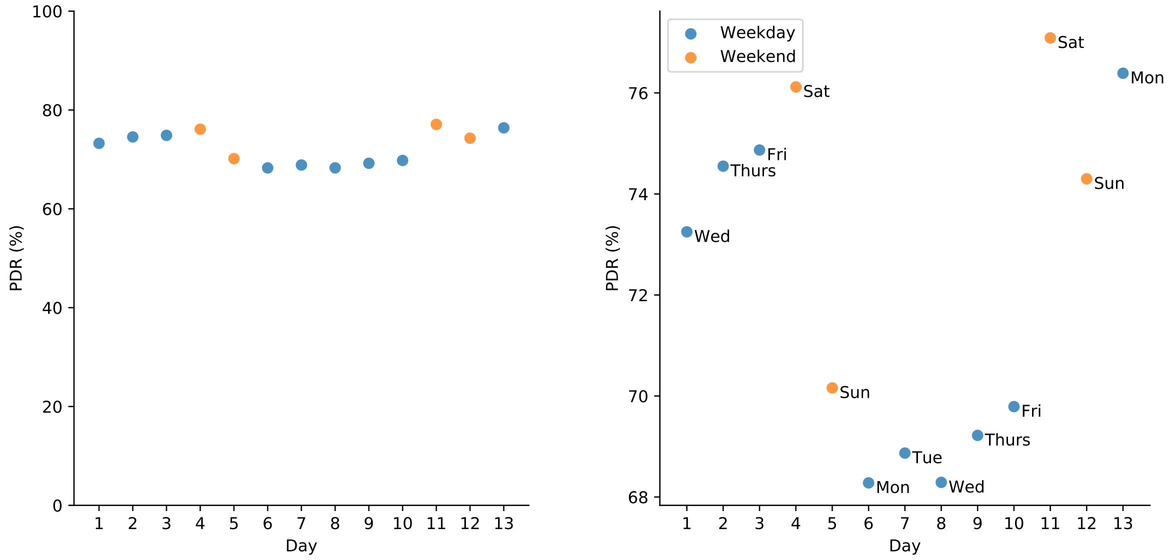
3.6.2 Exploratory data analysis

Exploratory data analysis was performed to better understand the data generated by the testbed. The aim of this data analysis is to gain insight into the environmental conditions, and to see if any time periods should be avoided when conducting experiments. Additionally, the PDR was calculated using different targets for the number of packets sent by a node to see if 1000 packets per node is an appropriate target.

The daily PDR, when averaged over 18 nodes for a consecutive 13 day period, is shown in Figure 3.10. The graphs show that the percentage point difference in PDR per day remains small over the period. The slight decrease seen for week two is likely due to external factors such as weather and interference from other users in the frequency bands. More data is required to see any long term trends, but the available data shows that the entire week can be used. A measurement in which nodes sent a packet every 30 seconds was used for this analysis as this was the only measurement with several days worth of data.

After the examination of daily PDRs, an investigation into hourly PDRs was performed. This was to determine if any time periods should be excluded as external factors might have a significant impact on the testbed in certain hours. If a node was disconnected for an entire hour, its PDR was considered as 0 % as no data packets were captured. Figure 3.11 shows the hourly PDR for the first day (Wednesday). The average PDR for this day was 72.83 % with a standard deviation of 2.56 % and no clear patterns can be seen.

The hourly PER for all 13 days is presented in Figure 3.12. When examining each day it can be seen that the PDR fluctuates up and down throughout the day. The hourly PDR for each day in week 2 was consistently lower when compared to week 1 and 3. No hour(s) during this week was significantly higher than previous hours, although if some were found this could have explained the decrease seen in Figure 3.10's week 2.



(a) Daily PDR **(b)** Zoomed in version of (a)

Figure 3.10. Daily Packet delivery rate, averaged over 18 nodes each sending 1000 packets.

The average and standard deviation for each day shown in Figure 3.12 is provided in Table 3.5. The variation in the standard deviation is low indicating that any patterns in the PDR and the influences of the external environment remain consistent over time. Therefore, all hours in a day will be used for experiments. Figure 3.13 shows the histogram and the associated kernel density estimate for the hourly PDR over the 13 day period for each group. By plotting each group separately, the impact of the range differences become evident, and as expected, the decrease in PDR corresponds with the increase in range. For all three groups the distribution of the data corresponds to a normal distribution.

Finally, the PDR was calculated using different target totals to see if 1000 packets per node is an appropriate target. Figure 3.14 shows the results of target totals 100 to 10 000. The changes in PDR appears to be minimal in Figure 3.14(a), and a closer examination (3.14(b)), confirms that 1000 packets per nodes is a suitable choice. The average PDR of a 1000 packets is 73.25% which is 0.41 percentage points lower than the average for the 10 000 packets measurements which have a standard deviation of 0.74 percentage points. Sending 1000 packets will take every node at least 16.67 hours when sending one packet every 60 seconds. Using a value such as 100 will only capture packets for 1.67 hours which is a very short duration and results could be influenced by an external event. Using more than 1000 packets will result in tests taking several days to complete, reducing the number of tests that can be

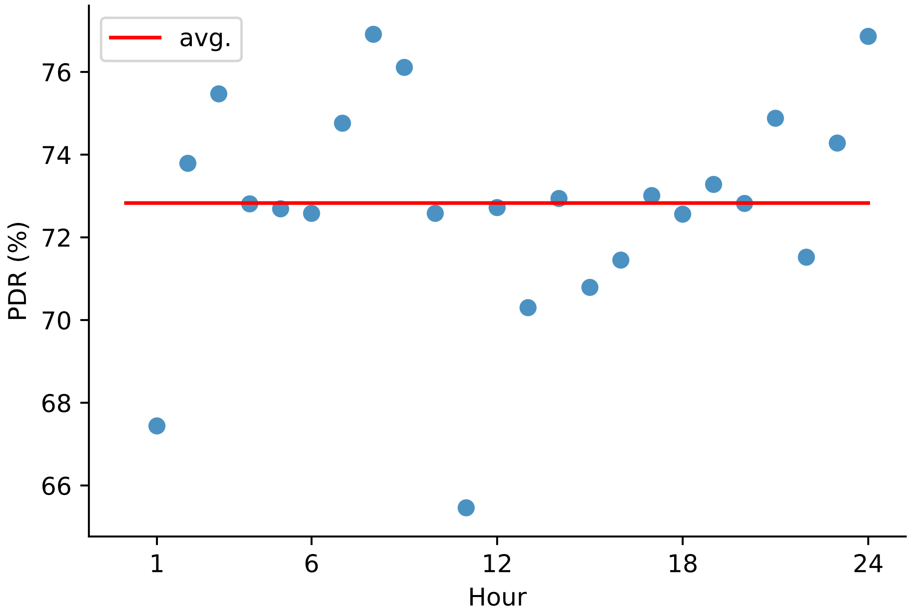


Figure 3.11. Hourly PDR over a 24 hour period (averaged over 18 nodes).

Table 3.5. Average PDR and standard deviation for 13 days.

Day	**Average PDR (%)**	**Standard deviation**
Wed	72.83	2.58
Thurs	74.57	2.32
Fri	75.5	2.17
Sat	73.22	3.54
Son	70.32	1.53
Mon	70.15	2.05
Tue	68.94	1.83
Wed	69.58	2.08
Thurs	70.27	2.45
Fri	68.97	2.79
Sat	76.53	1.67
Son	75.4	2.35
Mon	76.33	2.52

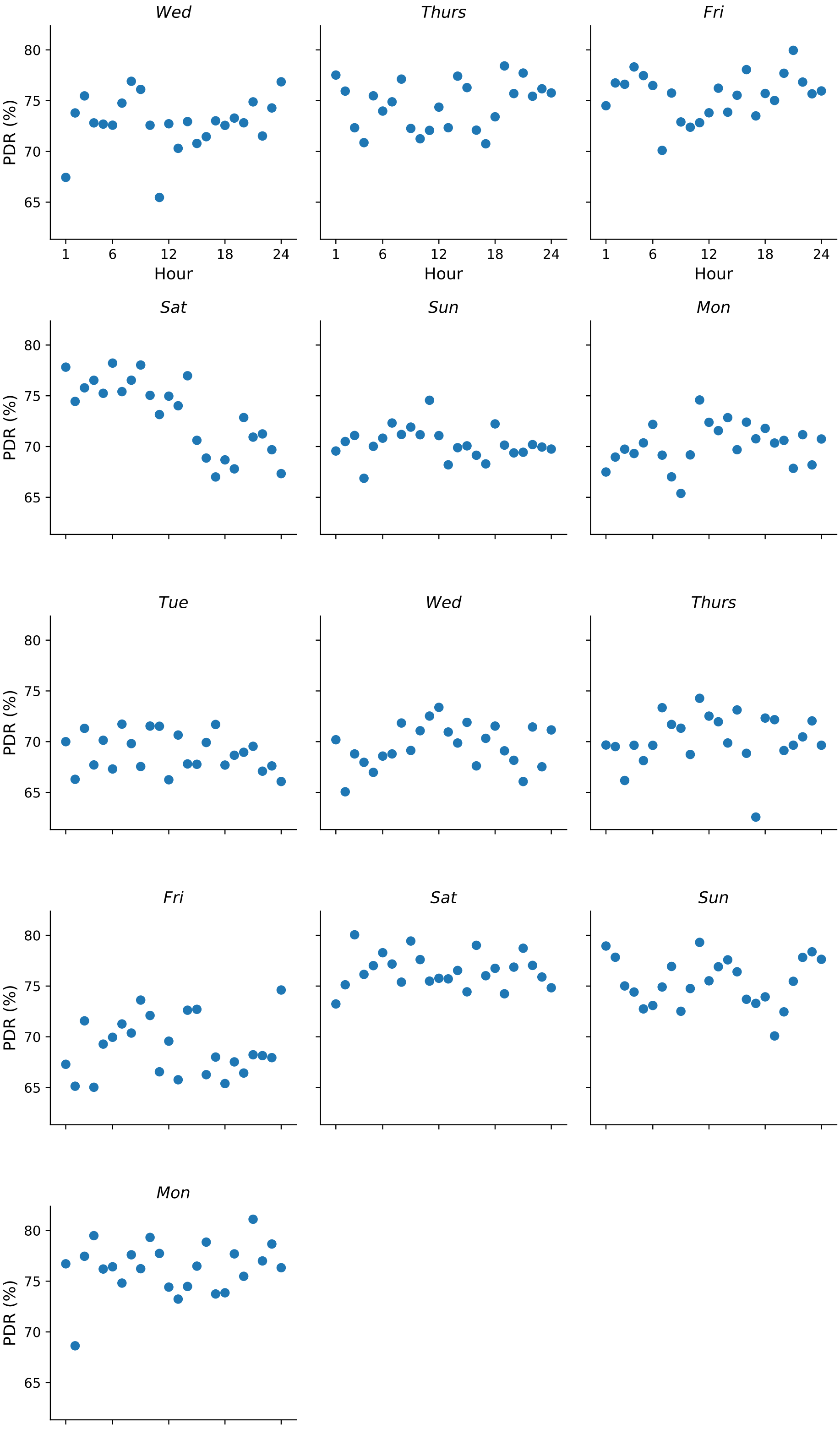


Figure 3.12. Hourly PDR for 13 days (averaged over 18 nodes).

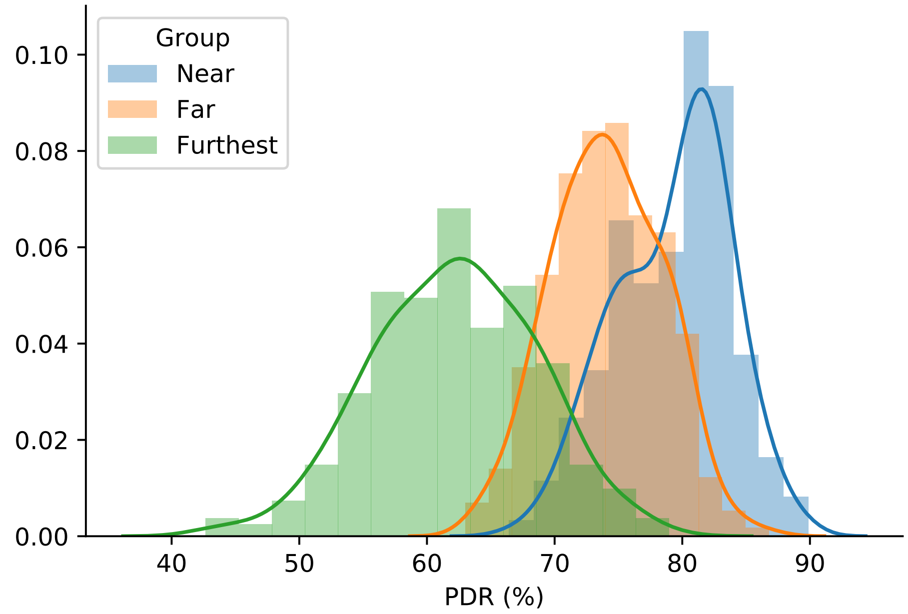
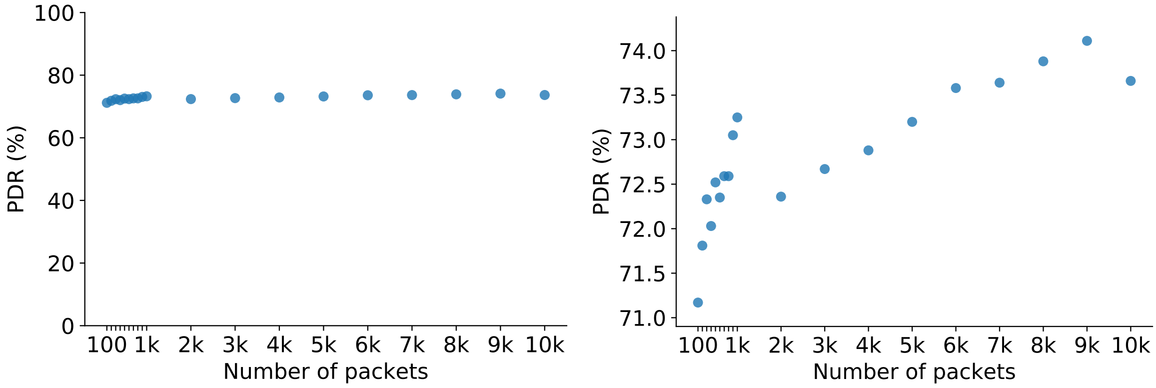


Figure 3.13. Kernel density estimate of the hourly PDR for each group over the 13 day period.

performed in a reasonable time.



(a) Averaged PDR

(b) Zoomed in version of (a)

Figure 3.14. Packet delivery rate, averaged over 18 nodes each sending the same target amount of packets.

3.6.3 Experimental procedure

All days and all hours of the week will be used with the exception of any time slots in which a node had reset due to software or low battery problems. When this occurred, a tweet would have been sent to the user and the node reconfigured. The time between the tweet and the reconfiguration will be excluded when the data is analysed.

These configurations can be started at any time but, when querying data from the database, a server time synchronisation problem must be considered. The web server system time is not synchronised with a time server and thus its clock is slightly ahead and drifts further ahead. When the current time is 09:46:00 AM the database will record an incoming packet's time as approximately 10:01:00 AM. Figure 3.12 showed that no hour is significantly different from another so this drift is not calculated and compensated for. Instead, for convenience and consistency, the first recorded packet's timestamp is examined and for that measurement data is taken from the next hour. For example, if the first packet was recorded as 10:30:45 AM, data is queried from 11:00 AM. This makes comparing the amount of time each node requires to reach 1000 packets easier.

3.7 CHAPTER SUMMARY

In this chapter, the design and implementation aspects of the LoRaWAN testbed were presented. An overview of node locations, design considerations and the various hardware and software parts was provided. This was followed by a discussion on experimental methodology and procedure.

In Section 3.2 the key design considerations of a permanent outdoor testbed was examined. These include a mounting system, battery recharging, the remote configuration of experiments and data visualisation. Section 3.3 contains location information on all 18 nodes; the nodes were split into three groups based on distance from the gateway. Nodes were split into a Near group (less than 0.5 km), a Far group (between 0.54 km and 1.25 km) and a Furthest group (between 1.98 km and 5.19 km). Data analysis by group helps to better investigate the impact of distance on performance.

The hardware implementation was provided in Section 3.4. Nodes consist of a Multitech mDot and a rechargeable battery in an enclosure, with a solar panel and mounting bracket allowing nodes to be placed freely. A Multitech MultiConnect Conduit running mLinux will serve as the testbed's gateway.

Section 3.5 specifies the various software elements of the testbed. C++ firmware allows nodes to be sent AT command style commands via the gateway. Additionally, a web server running Node-RED, InfluxDB and Grafana with the Node-RED application provides the user with a convenient control interface. Incoming data packets are forwarded by the gateway using MQTT to the Node-RED application, which processes and stores them in an InfluxDB database. A Grafana dashboard, reading from this database, serves as a visual tool to monitor the testbed. The data gathered for each experiment was processed using Node.js before being graphed using a Jupyter Notebook.

The experimental methodology followed is presented in Section 3.6, with Packet Delivery Rate (PDR) chosen as the primary performance metric. PDR calculations will have a small error as a node can be in three possible states (Normal, Reconnecting and Misconfigured). The calculations can only be performed when nodes are in the Normal state, but a test determined that nodes will spend an average of 81.33 % in this state, making the error small.

It was determined that the best method of calculating the PDR was by using a target for the number of sent data packets per node rather than a time period method. Analysis revealed that the time period method results in unequal amounts of sent packets between nodes, impacting the standardisation of PDR calculations. An investigation into a suitable target number showed that a target of 1000 packets per node is suitable since this average is only 0.41 percentage points lower than the average using 10 000 packets. This allows tests to be completed in a short time span (approximately 17 hours). Exploratory data analysis also revealed that all days of the week and all hours of each day can be used when logging data.

CHAPTER 4   RESULTS

4.1 CHAPTER OVERVIEW

This chapter details the results gathered from experiments performed using the constructed LoRaWAN testbed. An overview of the measurements discussed is presented in Section 4.2, followed by the presentation of the results. PDR was used as the main performance metric, but other metrics were used where needed. The experiments, in the order in which they are discussed are: an investigation into RSSI and SNR values, impact of ADR, impact of payload length, impact of waiting time, impact of link checks and, finally, the impact of acknowledgements. All experiments investigated performance by examining each group of nodes (Near, Far and Furthest) independently. In some experiments individual node(s) were measured.

4.2 MEASUREMENTS

4.2.1 Overview

Several experiments were performed using the methodology described in Section 3.6, with the collected data stored in an InfluxDB database. A baseline node configuration was developed, and a set of experiments was executed in which each experiment changed one configuration element from this baseline. This approach allows each experiment to investigate a specific aspect, for example, how is PDR influenced by packet length? Each experiment had its own dataset which contained the data of several tests, each stored in a measurement, which is an InfluxDB data structure. For example, the dataset for the experiment examining packet length contains 14 measurements as 14 tests were conducted to examine 14 different packet lengths. In these tests, the baseline node configuration serves as a starting point, with only the payload length element being modified.

The baseline node configuration, described in the format defined in Section 3.5.2, is the following: "AC0;LC45;TX11;AN3;AD1;WT60;PS5;DR0". Each configuration element and its purpose is described in Table 4.1. Note that, if the ADR scheme is enabled, the data rate set with the "DR" element will be ignored as the ADR scheme is in control. A measurement in which each node sent 1000 packets, using the baseline configuration, was created to serve as a performance baseline for experiments to compare with.

Table 4.1. Baseline node configuration defined.

Item	**Purpose**
AC0	Requesting packet acknowledgements are disabled
LC45	Link connectivity is checked every 4 packets and 5 checks must fail
TX11	Transmission power is set to 11 dBm
AN3	Antenna gain is set to 3 dBm
AD1	ADR is enabled
WT60	waiting time between packets is 60 seconds
PS5	application payload length is 5 bytes
DR0	data rate is set to DR0 (a SF of 12)

Table 4.2 shows each measurement in the InfluxDB database and its purpose. Each measurement has a name by which it will henceforth be called, and a separate name in the InfluxDB database. These two names are normally the same, but if a test had to be repeated due to, for example, node failures or server maintenance, they will differ. As described in Section 3.6, tests were started at any time and were stopped once 1000 packets per node were sent. Note that, due to packet loss, 1000 packets transmitted will result in less than 1000 packets received and subsequently stored in a measurement. The number of sent packets is calculated by examining the packet sequence numbers, for example, the sequence numbers of three stored packets could indicate that the node sent 5 packets. Section 4.3.1 discusses the SNR and RSSI behaviour of the testbed. Unlike PDR calculations which can calculate missing packets, a SNR and RSSI value is not imputed for lost packets. This section only examines values for received packets. Therefore, this section uses 574 packets (the biggest number of received packets in all measurements).

Table 4.2. Measurements in the InfluxDB database and their purpose.

Name	InfluxDB name	Purpose
Baseline	Baseline	Baseline configuration.
Basic	Basic	ADR is disabled.
ACK1	ACK1	Requests packet acknowledgements but does not retry.
ACK2	ACK2	Requests packet acknowledgements and retries once.
ACK3	ACK3_r	Requests packet acknowledgements and retries twice.
PS10	PS10	Application payload is set to 10 bytes.
PS15	PS15	Application payload is set to 15 bytes.
PS20	PS20_r	Application payload is set to 20 bytes.
PS25	PS25_r	Application payload is set to 25 bytes.
PS30	PS30	Application payload is set to 30 bytes.
PS35	PS35_r	Application payload is set to 35 bytes.
PS40	PS40_r	Application payload is set to 40 bytes.
PS45	PS45	Application payload is set to 45 bytes.
PS50	P510	Application payload is set to 50 bytes.
PS100	PS100	Application payload is set to 100 bytes.
PS150	PS150	Application payload is set to 150 bytes.
PS200	PS200	Application payload is set to 200 bytes.
PS241	PS241	Application payload is set to 241 bytes.
WT30	wt30	Waiting time between packets set to 30 s.
WT45	wt45	Waiting time between packets set to 45 s.
LC15	LC15_rr	Link checks set to every packet and 5 checks must fail.
LC35	LC35	Link checks set to every 3 packets and 5 checks must fail.
LC55	LC55	Link checks set to every 5 packets and 5 checks must fail.
LC53	LC55	Link checks set to every 5 packets and 3 checks must fail.
LC59	LC59	Link checks set to every 5 packets and 9 checks must fail.

4.2.2 A protocol issue experienced during payload length experiments

In PS100, PS150, PS200 and PS241, only 9 nodes were the focus of the test. The LoRaWAN protocol limits the maximum payload sizes for LoRa packets based on the SF and BW used to send them. Nodes that are far away from the gateway or exhibit significant connectivity losses will end up being set to SF/BW combinations that do not allow longer payload lengths. These nodes were excluded from calculations when longer payload sizes were examined. Table 4.3 shows the nodes that were used in payload size tests that exceeded 50 bytes. All nodes were used for tests up to 50 bytes.

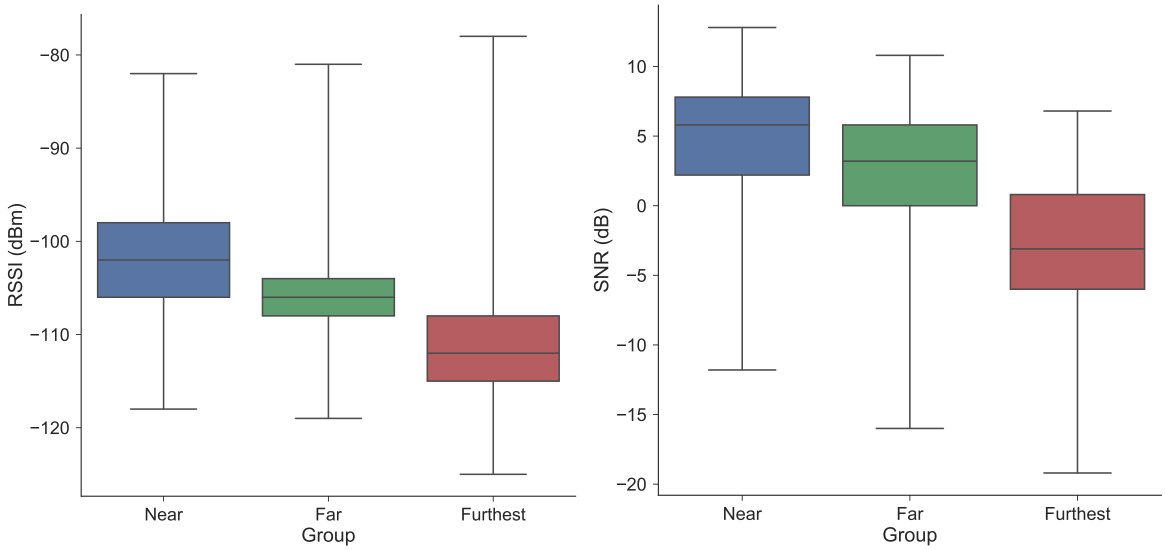
Table 4.3. Nodes included in tests with payload lengths of more than 50 bytes.

Node	Distance (km)	Location	Group
e5-43	0.14	CSIR	Near
eb-94	0.24	CSIR	Near
eb-96	0.25	CSIR	Near
e5-3f	0.28	CSIR	Near
e5-47	0.50	CSIR	Near
eb-9a	0.54	CSIR	Far
e5-45	0.66	CSIR	Far
e5-48	0.91	CSIR	Far
e5-46	0.97	CSIR	Far

4.3 RSSI AND SNR

4.3.1 Performance per group

Due to the CSS modulation that LoRa nodes use, the nodes are capable of successfully transmitting packets over long distances to a gateway capable of demodulating the packets with their low RSSI values and negative SNRs. Figure 4.1 shows a box-and-whisker plot for captured RSSI and SNR values from the Baseline measurement, which had RSSI and SNR data on 1000 received packets. Each group's coloured box is drawn with the lower quartile forming the bottom of the box, the middle quartile (median) forming the middle line and the upper quartile creating the top of the box. The lower quartile line is defined as the value below which 25 % of the recorded values lie, and the upper quartile line is the value below which 75 % of the recorded values lie. The interquartile range (IQR) can be calculated as the upper quartile minus the lower quartile and this range indicates where 50 % of the recorded values lie. From the box extends the lower and upper whiskers showing the values outside the IQR.



(a) RSSI values by group.

(b) SNR values by group.

Figure 4.1. Captured RSSI and SNR values for 1000 received packets per node, ordered by group.

The Baseline measurement's data was used as is, with the exception of 1 packet from e5-40 and 1 packet from eb-97 which were removed, as their recorded RSSI values were both 199 dBm. This value was exceptionally high compared to other values for node e5-40, with a mean of -113 dBm and the values for eb-97 which had a mean of -114 dBm. These outliers were over 25 times both nodes' standard deviation of approximately 9 dBm.

The lowest recorded RSSI and SNR value were respectively -125 dBm and -19.2 dB, with the best values being -78 dBm and 12.8 dB. The big difference between the best and worst conditions in which packets were successfully received showcases LoRa's high sensitivity and suitability for long range communication.

Figure 4.1(a) shows the impact of distance on the received signal; the upper, middle and lower quartile values all decrease from one group to the next. The long whiskers of all the box plots show the wide spread of recorded values outside the IQR. The Far group's boxplot shows a much smaller IQR then the others, but the length of its whiskers show that, while 50 % of recorded values were densely grouped, there were still values recorded over a much wider RSSI range. There is a large overlap between the whisker ranges of the groups, and nearly any recorded RSSI value could belong to any of the groups, making determining a node's group based on RSSI highly inaccurate.

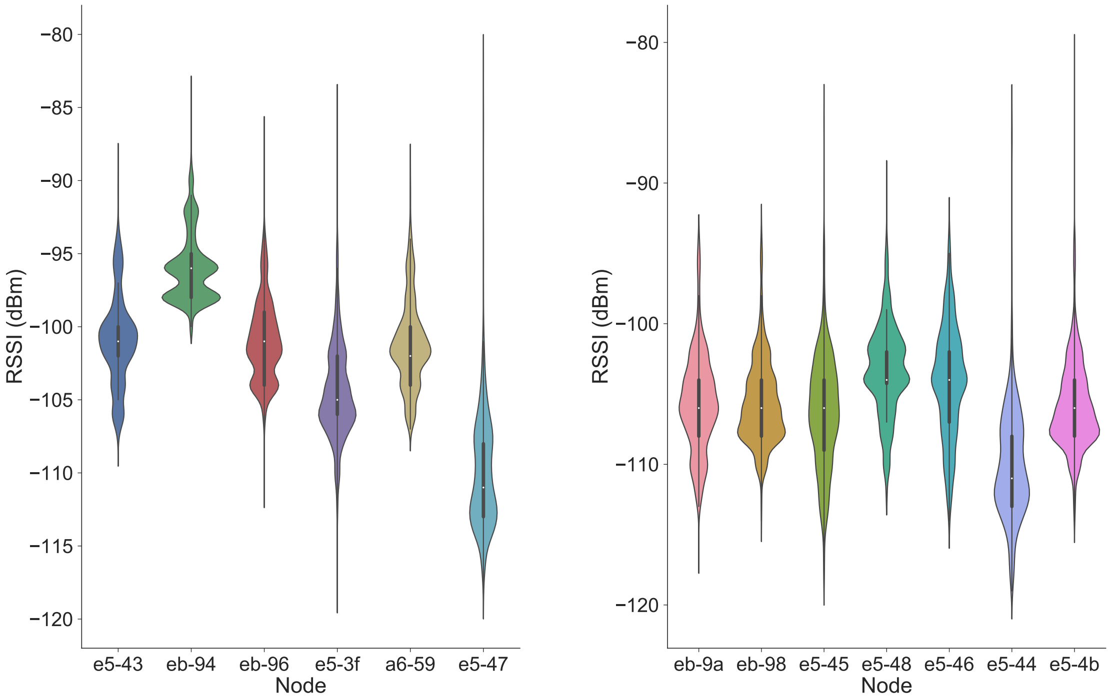
Similar to 4.1(a), the SNR boxplots shown in Figure 4.1(b) have a constant decrease in all quartiles values between one group to the next. The Near group IQR indicates that at least 50 % of the packets were recorded with a positive SNR with this group's whisker showing that some values were recorded far below 0 dB. The Furthest group's box plot shows the wide range of values captured for this group and that more than half of packets received had a negative SNR. As with the RSSI values, determining a node's group based on SNR values would be highly inaccurate.

4.3.1.1 Individual node performance

Figures 4.2 and 4.3 shows RSSI and SNR violin plots for individual nodes to reveal each node's individual performance. Nodes are plotted according to increasing distance from the gateway. In Figure 4.2(a), the shape of the distribution for each node in the Near group forms several small clusters either around, above or below the median. In the other two groups most nodes are more spread out, indicating a greater range in the captured RSSI values. All nodes had long upper whiskers, showing that every node, regardless of distance, experienced occasional packets with very good RSSI values.

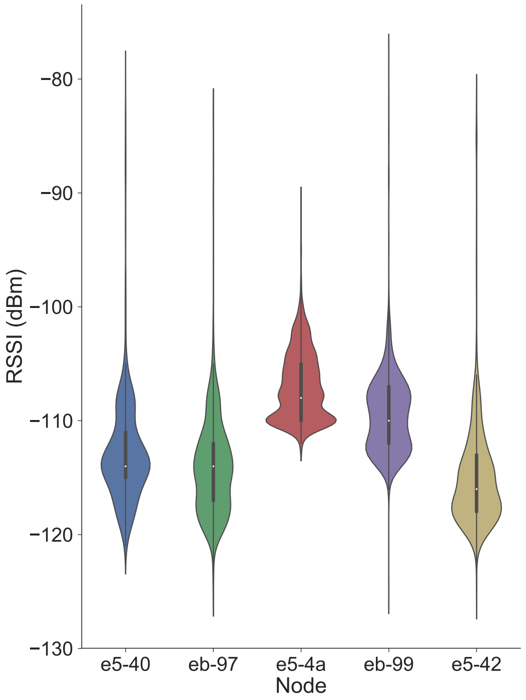
The whiskers also show that, if RSSI was to be used as a basic indicator of distance from the gateway, the number of measurements should be large enough to ensure that an occasional good or bad measurement does not impact any distance calculations. When the Near, Figure 4.2(a), and Furthest, Figure 4.2(c), groups are compared, it is clear that RSSI values do not always decrease based on distance, for example, e5-40 is 2.92 km closer to the gateway than e5-4a, but due to other factors such as position, height, and the immediate surroundings, node e5-40 has lower RSSI values in general.

The violin plots of the SNR values for most nodes in Figure 4.3 show a trend of being fairly symmetrical with respect to the median. Unlike the RSSI violin plots, the SNR violin plots show a more gradual tapering off for the lower 25 % of values. Similar to the RSSI plots, it will be hard to determine a



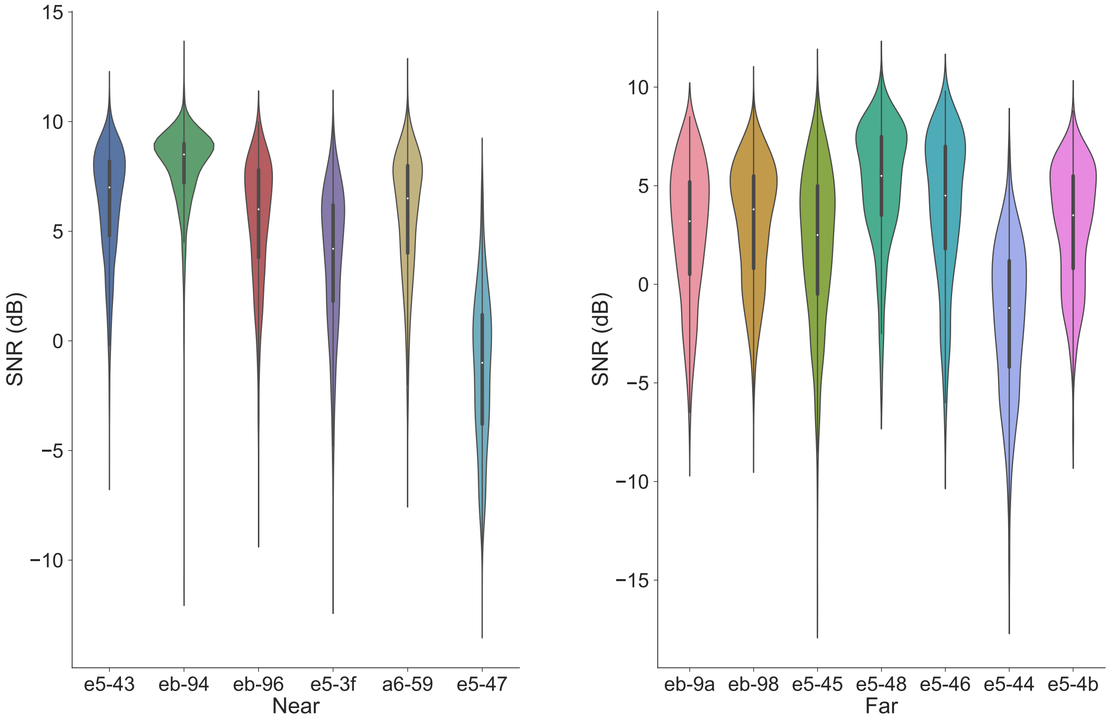
(a) Near group.

(b) Far group.



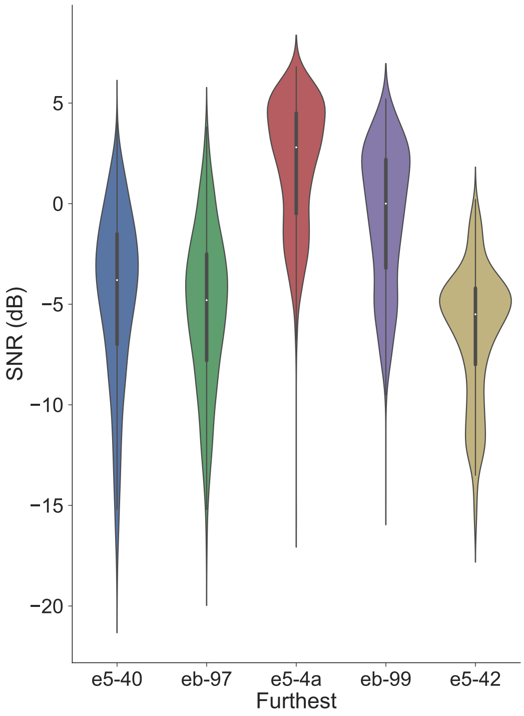
(c) Furthest group.

Figure 4.2. RSSI values for every node in each group.



(a) Near group.

(b) Far group.



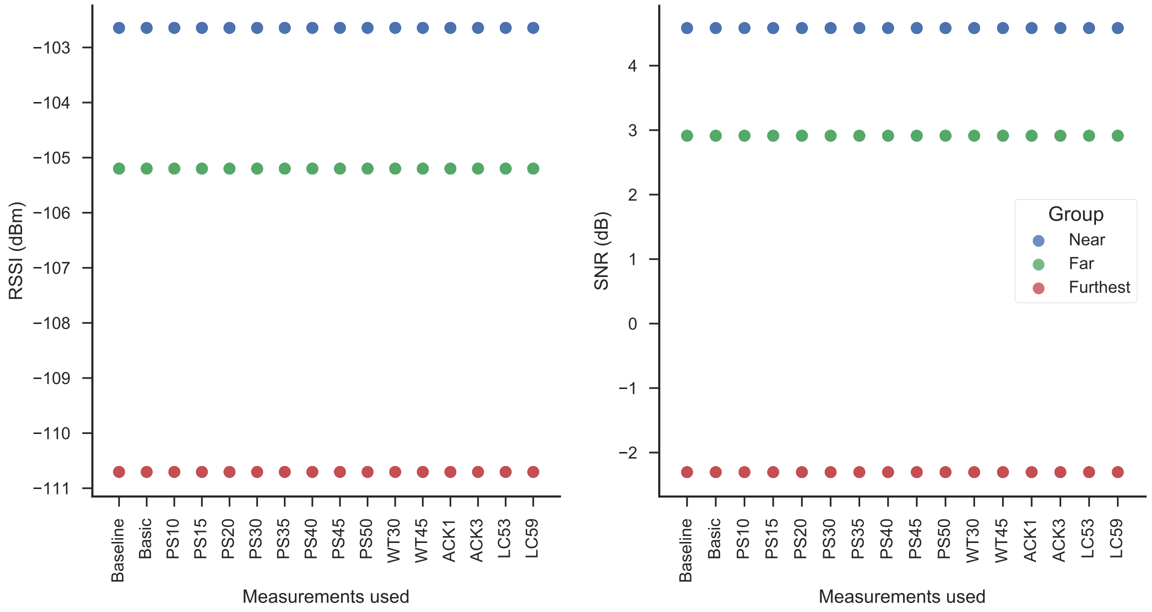
(c) Furthest group.

Figure 4.3. SNR values for every node in each group.

node's distance based on its SNR. The Near and Far groups, Figures 4.3(a) and 4.3(b), have very similar SNR levels whilst the Furthest group's levels were lower (Figure 4.3(c)).

4.3.2 Performance over several measurements

After this initial investigation into the Baseline measurement, the consistency of RSSI and SNR values over several measurements in the InfluxDB database was investigated. Table 4.2 details the purpose of each measurement and the average RSSI, and SNR values per group for these measurements are shown in Figures 4.4(a) and 4.4(b). It should be noted that the averages were calculated over 574 packets per node in every measurement as this was the maximum amount of received packets available in all the measurements. The graphs show no significant changes between the overall RSSI or SNR values in each measurement. The graphs do show a clear and constant difference in the RSSI and SNR values between the distance groups.



(a) RSSI levels averaged per group.

(b) SNR levels averaged per group.

Figure 4.4. Captured RSSI and SNR values for 574 received packets per node for multiple measurements.

4.4 IMPACT OF THE ADAPTIVE DATA RATE SCHEME ON PACKET DELIVERY

4.4.1 How the ADR scheme works

One of the main elements that allow LoRaWAN networks to handle thousands of nodes is the ADR scheme. This scheme adjusts the data rate and the RF output, also known as the Tx power, of individual nodes [6]. An increase in the spreading factor effectively increases the receiver's sensitivity thereby enabling longer range communication, but also increases the packet's transmission time resulting in lower data rates [62]. LoRa has two BW options allowed in this region, namely 125 kHz or 250 kHz. The 250 kHz option reduces the receiver's sensitivity (reducing range), but increases the transmission rate, and thus the data rate, when compared to 125 kHz [62]. In South Africa, which uses the EU868 standard, the ADR scheme therefore can select from 6 spreading factors. The SF7 spreading factor can be used with a BW of either 125 kHz or 250 kHz, while other SFs must use 125 kHz. Therefore, the fastest transmission option is SF7 in conjunction with a BW of 250 kHz, and the slowest is SF12 with a BW of 125 kHz. For the rest of this section, the notation "SF7BW250" and "SF7BW125" will refer to a node being assigned a SF of 7 with a bandwidth of either 250 kHz or 125 kHz.

CSS modulation enables node communications at different data rates to not interfere with each other, and thus allows a gateway to receive transmissions from multiple nodes simultaneously. The ADR scheme was developed for static nodes (fixed locations) and for environments in which the radio channel attenuation remains stable [6]. The ADR scheme will instruct nodes to use the fastest data rate possible to increase battery life and maximise network capacity [6].

The ADR scheme uses the LinkADRReq MAC command to request a node to perform an adjustment. This command provides a node with a data rate, Tx power, a channel mask (indicating which channels to use for uplinks) and a value stating how many times an uplink frame should be transmitted [66]. By default, a frame is only transmitted once, but through this command, this can be adjusted to retry transmission for up to 15 times. The maximum Tx power and allowed channels are region-specific and a LinkADRReq will be answered by the node with a LinkADRAns command to indicate if it accepted or rejected the new settings [6].

The ADR scheme contains an acknowledgement system designed to allow nodes to periodically confirm that the network received their uplink frames. A counter (ADR_ACK_CNT) is incremented with each uplink, and if after ADR_ACK_LIMIT number of uplinks was sent without any downlink frames being received from the gateway, the ADR acknowledgement request bit (ADRACKReq) is now set in future uplinks [6]. Once this bit is set, the network must respond with a downlink frame within the next ADR_ACK_DELAY number of uplink frames. If no downlink is received after this limit, the node will proceed with attempting to regain connectivity by switching to lower and lower data rates, each time switching to a lower rate once ADR_ACK_DELAY uplinks have been sent with no response. The ADRACKReq is only set if the device is using a data rate above the minimum [6].

Up to now, there has been no mention of how the ADR scheme determines the contents of the LinkADRReq commands it sends to each node. The LoRaWAN specification does not specify how ADR should be implemented, instead it is left to the Network Manager (NM). A superior ADR algorithm can be a competitive advantage and, as a result, vendors keep their implementations private [66]. The Things Network, a global collaborative network, uses an open source NM whose ADR algorithm is publicly available [73]. This implementation uses SNR values from the most recent 20 uplinks and will send an adjustment request if a fixed threshold is exceeded [66]. Multitech's implementation differs and is described as follows: "The network server samples the SNR from each packet and computes a possible datarate based on each sample. Six packets must be received by the network server before it will adjust the datarate of a device. Samples for the last 11 packets are maintained and when LinkADRAns is sent, the max datarate that has met a threshold of packets will be sent for the device to change to" [74]. Additional information and the threshold value is not supplied.

The scheme bases its choice of data rate on the past performance of each node. The popular opinion is that it is in the best interest of the network as a whole if nodes in close proximity to the gateway use lower SFs, freeing up the others SFs for use by long distance nodes. This argument, however, remains in dispute (see [15]). Packets modulated using different SFs enable multiple concurrent transmissions to be successfully decoded by the gateway, however, there are some caveats - as explained in [15].

4.4.2 Performance impact of enabling ADR

In Figure 4.5, 1000 received packets from the Baseline and Basic measurements are used to show the impact of enabling ADR for all distance node groups. In the Basic measurement, ADR is disabled and all nodes use the maximum spreading factor (12). The figure shows that the averaged PDR for each group is consistently worse when ADR is enabled, with the Furthest group showing the biggest difference. The PDR drops by 3.6, 8.6 and 12.5 percentage points respectively for the Near, Far and Furthest groups.

This decline is unfortunate as the LoRaWAN specification recommends that the ADR scheme should be enabled whenever possible. The results indicate that the scheme is not choosing rates with reliable packet delivery as a key goal. An increase in the PDR could be expected if an ADR disabled network was suffering from lots of collisions caused by all of the nodes operating with the same SF. Enabling ADR can improve the PDR in such a network if the scheme assigns different SFs to some of the nodes, thus reducing congestion.

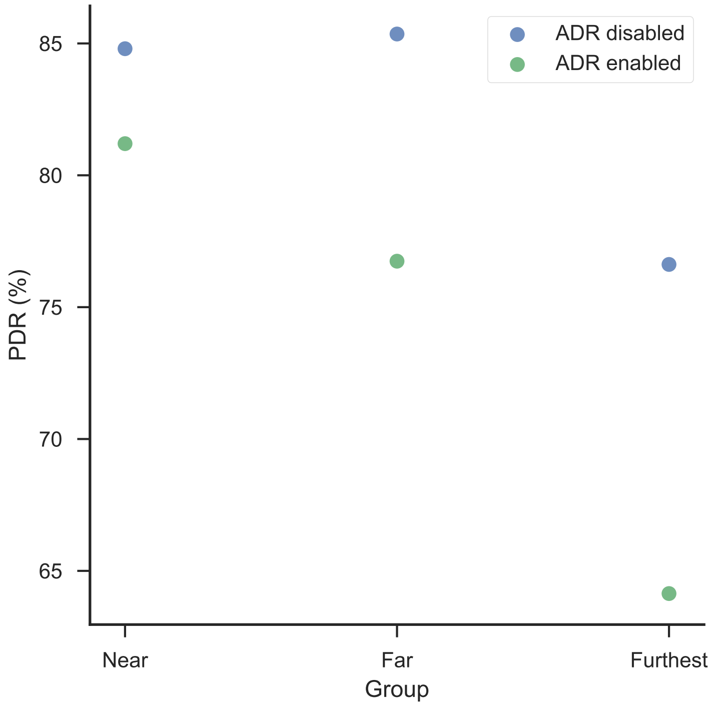


Figure 4.5. The PDR per group when ADR is enabled versus all nodes using SF12.

This decrease in the PDR can also be seen in Figure 4.6, in which the PDR is plotted for each node. The nodes are ordered by distance from the gateway, with the first 6 nodes representing the Near group, the next 7 the Far group and the remaining 5 the Furthest group. Nearly all nodes had a lower PDR, with only a few nodes performing better when ADR was enabled.

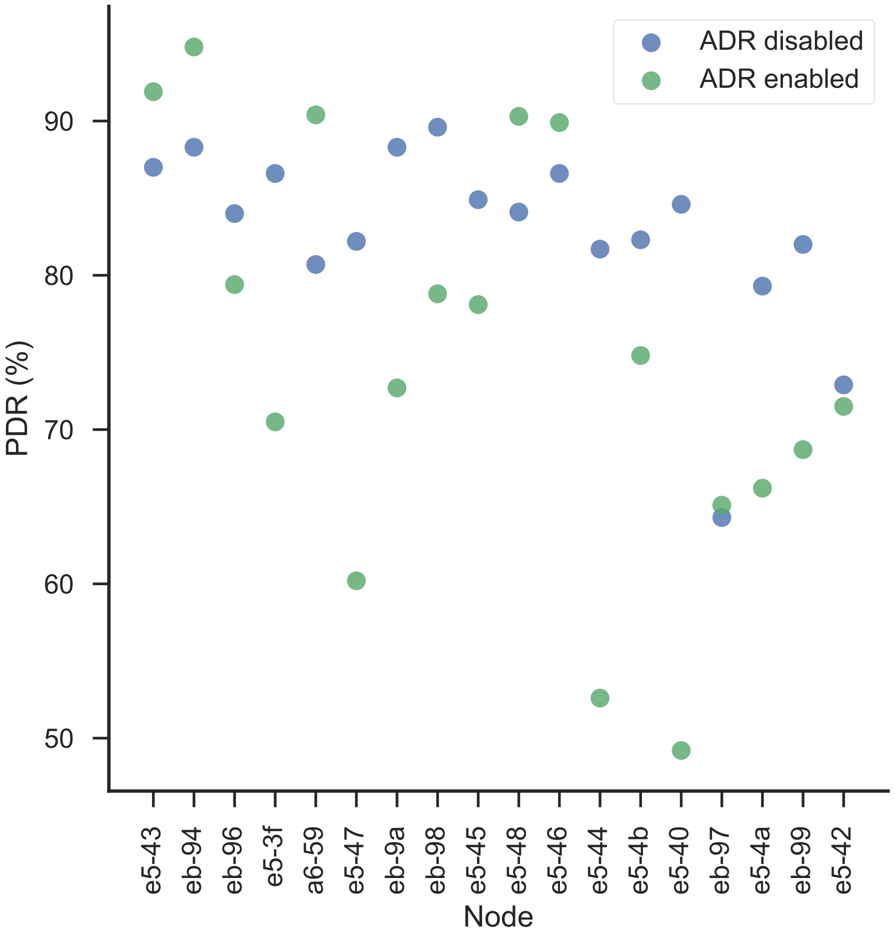


Figure 4.6. The PDR per node when ADR is enabled versus all nodes using SF12.

Figure 4.7 shows a percent stacked barchart of which spreading factors were assigned by the ADR scheme to the nodes in each group. All three groups have a significant percentage of the SF7BW250 configuration, with the Furthest group having the largest variation in assigned configurations. The Near and Far groups only had SF7BW250 and SF12 assigned to their nodes. The Furthest group had a small percentage of SF7BW125 and SF8 configurations. In this group, the two dominant configurations (SF7BW250 or SF12) were approximately equally assigned.

The non-assignment of spreading factors 9, 10 and 11 and the strong preference for either SF7BW250

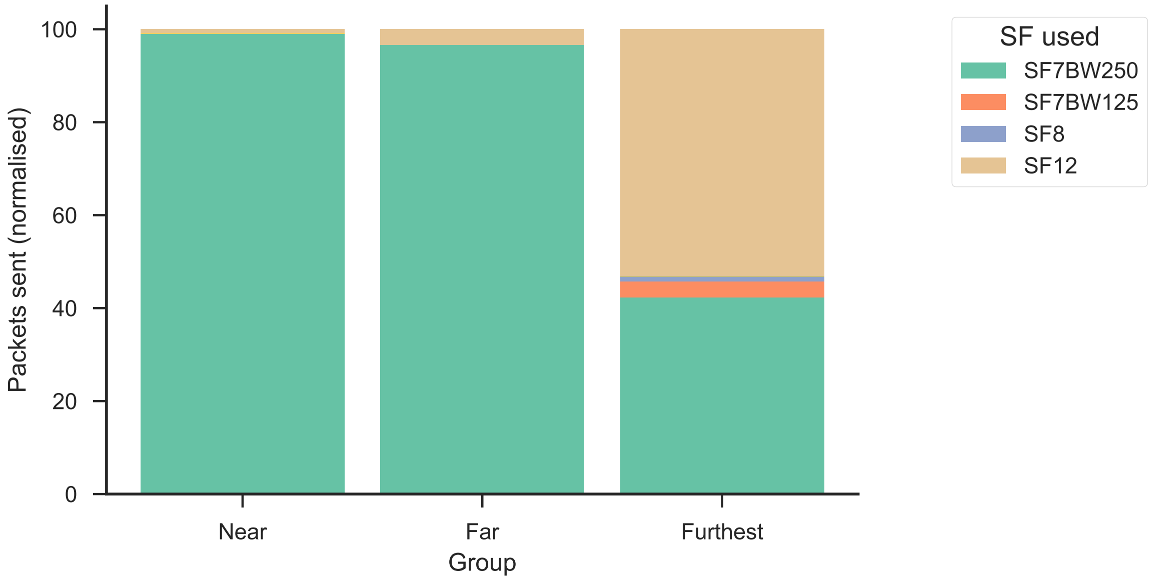


Figure 4.7. Spreading factors chosen by the ADR scheme.

or SF12 seems unusual. To confirm the accuracy of the bar chart, Grafana was used to plot the same data to check if a different data analysis and plotting format would deliver different results, and this is shown in Figure 4.8. This graph matches the results of Figure 4.7 and also shows that the other SFs were not used. It should be noted that Grafana's graph does not distinguish between SF7BW250 and SF7BW125 and plots both as a SF of 7.

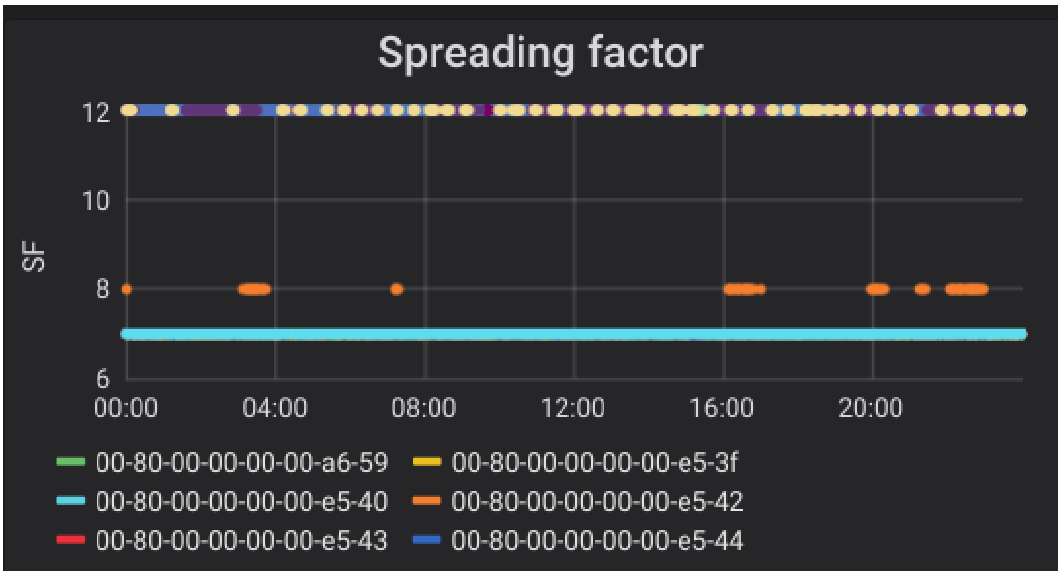


Figure 4.8. Spreading factor plotted using Grafana.

Figure 4.9 shows how node e5-42 (part of the Furthest group) varies between SF7, SF8 and SF12. The graphs show that the ADR scheme did not gradually choose lower and lower spreading factors, but instead the assignment abruptly changes between the three options. Table 4.4, which contains 1000 data points for each node, shows that this dramatic change occurs for all nodes in the Furthest group, and how drastically the difference in the assignment of either SF7BW250, SF12 or SF8 are between these nodes. Particularly, eb-99 has a 60/40 split between SF7BW250 and SF12 whilst e5-42 (170 m away, and with a lower SNR) heavily favours SF12 with almost no packets sent using SF7BW250. With these long distance nodes, SF12 was likely a good choice to help deal with interference and signal strength loss.

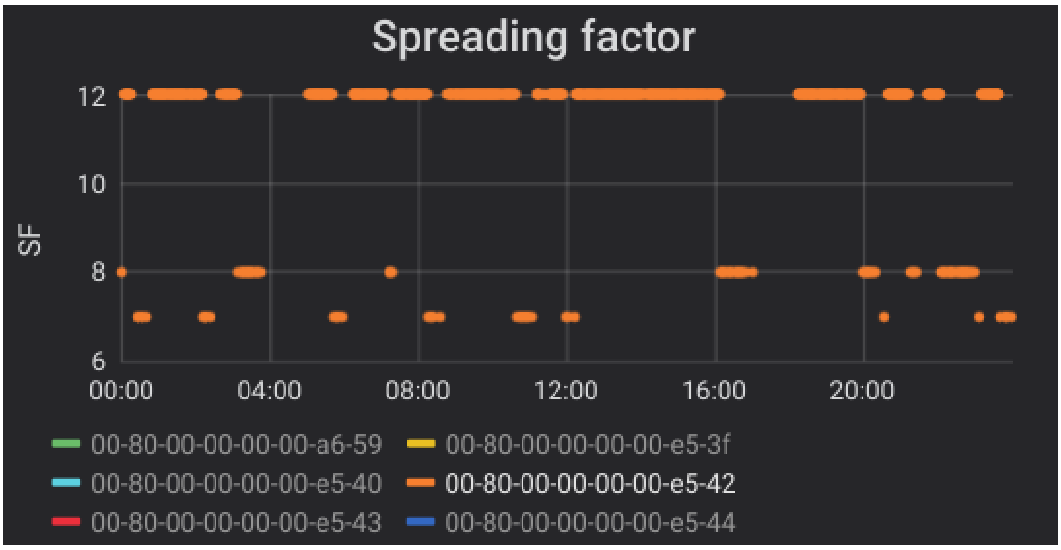


Figure 4.9. Spreading factors used by e5-42.

Table 4.4. Spreading factor assigned to the Furthest group by the ADR scheme.

Node	SF7BW250	SF7BW125	SF8	SF12	Distance (km)
e5-40	596	23	0	381	1.98
eb-97	50	70	0	880	1.98
e5-4a	981	0	0	19	4.90
eb-99	595	0	0	405	5.02
e5-42	6	45	119	830	5.19

4.4.3 Performance over several measurements

So far, this measurement's data shows that the ADR scheme made some dramatic choices, so the question was asked if the ADR continues to act consistently in its pattern of choices over multiple measurements? Figure 4.10 shows that the choices are consistent in terms of the previous pattern of SF and BW assignments. As was the case in Section 4.3.1, only the values for 574 received packets in each measurement was used. There is variation in the ratio between the SF7 and SF12, but it is fairly small.

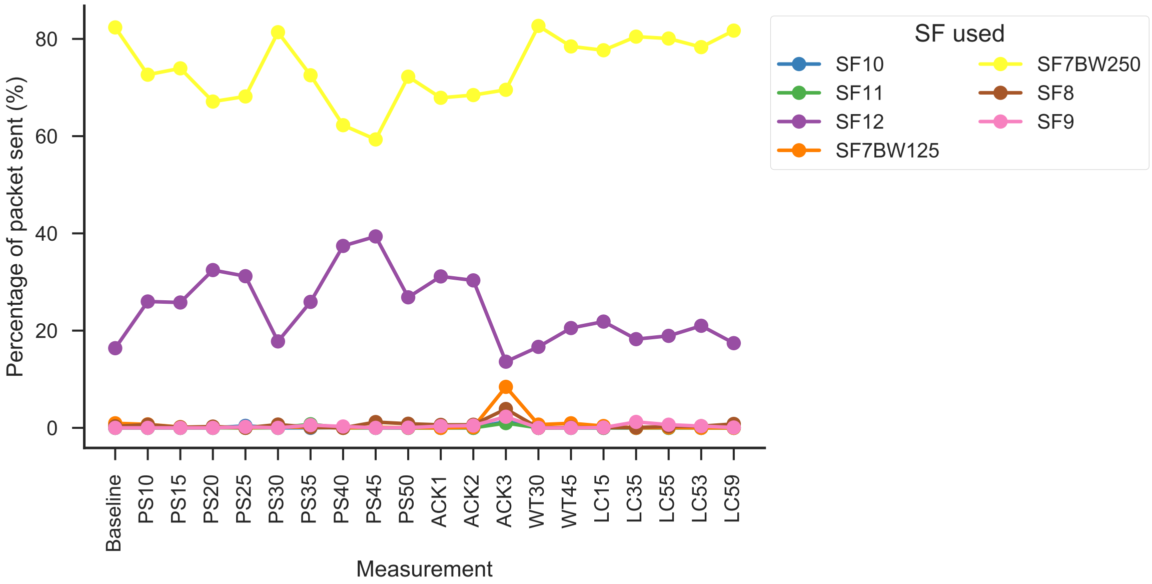


Figure 4.10. Spreading factors used in a selection of 20 measurements.

Table 4.5 details in how many of the measurements used in Figure 4.10 the ADR scheme selected each less utilised SFs. Only in ACK3 did the ADR use all of the available SFs, with SF11 being the least used, that is, 0.98 % of the received packets. Whilst all spreading factors were used in multiple measurements, their assignment percentages were very small. One of the key factors that allows a LoRaWAN to have a large network capacity is the fact that communication on different SFs are orthogonal to each other, thus allowing a gateway to receive them simultaneously. A congested network's PDR will also be improved if communication takes place using different SFs as it reduces the number of packets that are lost due to collisions. In a small network such as the testbed the under utilisation of some SFs would not have had a high impact as the network is not congested.

The experiment examining the enablement of acknowledgements is represented in measurements ACK1 to ACK3. The results for ACK1 and ACK2 follow the overall trend. In the ACK3 measurement, three acknowledgement attempts are allowed before the node stops transmission of a packet. It is uncertain why ACK3 is the only measurement showcasing a larger selection of SF assignments. Environmental conditions may be a potential cause, but Figure 4.4 shows no significant change in the RSSI and SNR values when this test was performed.

Table 4.5

Table 4.5. Spreading factor assignment by the ADR scheme for less utilised SFs.

SF	Found in how many measurements	Highest assignment (%)
SF7BW125	15	8.43
SF9	11	2.28
SF10	3	1.29
SF11	2	0.98

4.5 IMPACT OF PAYLOAD LENGTH ON PDR

4.5.1 LoRaWAN protocol considerations

An investigation was performed to determine how the length of a LoRa packet's application payload impacts the PDR. Depending on the use case, nodes could be required to send 5 bytes or 200 bytes of information. Several tests have been completed in which the application payload length was varied between 5 bytes to 50 bytes in 5 byte increments, and 100, 150, 200 and 241 bytes were tested as well. The lengths referred to here are the number of bytes sent by a user's application running on the mDot processor of the node. The completed LoRa packet will always be longer due to protocol overhead. The LoRaWAN protocol has maximum application payload length limits and these depend on the SF/BW combination used and whether MAC commands are to be transmitted with the payload [8]. The protocol allows for up to 15 bytes of MAC commands to be sent, which could reduce the allowable length of the application payload.

The MAC commands are only attached when required, thus the firmware first determines if any commands will be appended to a payload. The firmware then determines the number of bytes the MAC commands require, before calculating if these additions will cause the maximum length to be exceeded. The application payload is randomly generated and its length will be reduced should the calculation determine that the maximum length would be exceeded. The maximum payload lengths, with no MAC commands, are either 51, 115 or 242 bytes depending on the SF/BW combination used [8].

As an example, if a test specified an application payload of 50 bytes but 4 bytes of MAC commands are to be sent, the node will randomly generate only 46 bytes before transmitting the packet. For a node using SF12, the maximum legal total length is 51, and with 4 bytes of MAC commands it could have generated 47 bytes not 46. However, the mDot library provided by Multitech does not count the link check command when it provides an indication of MAC command length. The link check command is 1 byte long, and therefore the maximum application payload would be decreased by 1 byte. To ensure that packets never exceed their maximum length, the application payload length is reduced by 1, making the maximum allowed lengths 50, 114 and 241 even if the mDot library calculates that no MAC commands are to be sent.

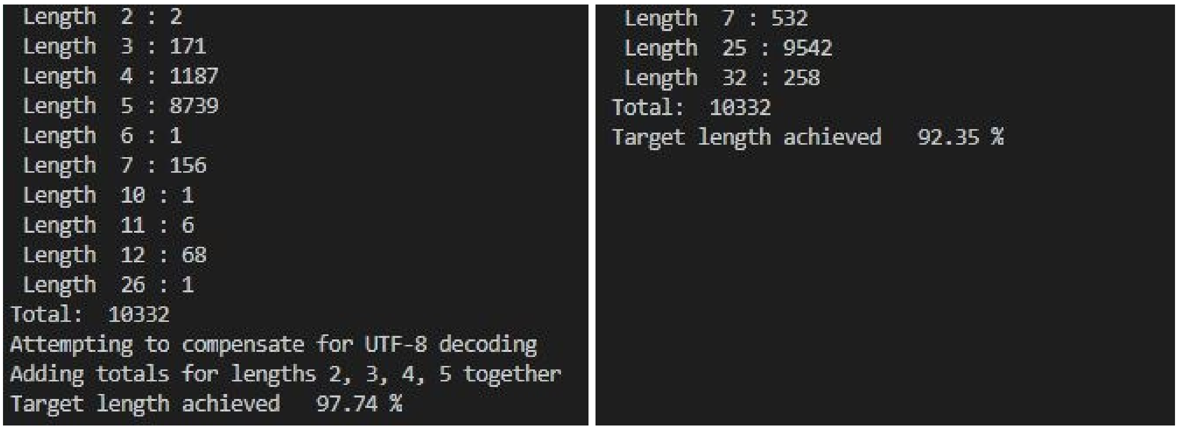
As the payload length is only reduced if the addition of MAC commands would have caused an illegal length, this reduction only appeared in PS50 and PS241 as these are extremely close to the legal limits of 51 and 242 bytes respectively. When the 574 received packets per node were examined in measurement PS50, 0.23 % had slightly reduced lengths to accommodate the MAC commands.

In the case of the Furthest groups of nodes and several nodes of the Far and Near groups, the LoRa protocol automatically assigned higher spreading factors, and therefore these nodes could not be tested with payload lengths of more than 50 bytes. All 18 nodes were used in the tests ranging from 5 to 50 bytes, while a selection of 9 nodes were used in the tests above 50 bytes. The other 9 nodes were set to send payload lengths of 50 bytes, but were not monitored, and the tests stopped once the chosen 9 nodes each sent 1000 packets.

4.5.2 Firmware considerations

In addition to sending data packets with the specified application payload length, nodes will also send status messages. These are short messages with a maximum length of 7 bytes. These messages are included in the PDR calculation. Another variation is the fact that the mDot library indicates a failed transmission by appending the full data packet message to the next data packet sent, thus doubling the total packet length of the next packet, while not exceeding the set limit of maximum packet length. The REJOIN and PAC2B status messages are the only ones that would be present in the data, and the other status messages that indicate that nodes are resetting are not present. This is due to the fact that tests in which node resets appeared were replaced by successful tests.

The frequency with which status messages and appended packets occur in the measurements was determined. A Node.js script was used to determine how frequently these occur, and an example of the results are shown in Figure 4.11. Section 3.5.4 detailed that, when the earliest measurements were conducted, the application payload was incorrectly decoded into UTF-8 when saving the data to the database and, as a result, the sent payload lengths would be incorrect for some measurements. Status messages were, however, correctly decoded. Therefore, the incorrect lengths are those of the application payloads. The Node.js script was designed to deal with the UTF-8 data by grouping the occurrences by lengths with respect to the specified length, that is, lengths 2,3,4 and 5 are combined in the group with 5 bytes specified.



```
 Length  2 : 2
 Length  3 : 171
 Length  4 : 1187
 Length  5 : 8739
 Length  6 : 1
 Length  7 : 156
 Length  10 : 1
 Length  11 : 6
 Length  12 : 68
 Length  26 : 1
Total:  10332
Attempting to compensate for UTF-8 decoding
Adding totals for lengths 2, 3, 4, 5 together
Target length achieved   97.74 %
```

(a) Lengths for Baseline.

```
 Length  7 : 532
 Length  25 : 9542
 Length  32 : 258
Total:  10332
Target length achieved   92.35 %
```

(b) Lengths for PS25.

Figure 4.11. Application payload length results for Baseline and PS25.

Table 4.6 shows the different application payload lengths encountered in each measurement. The length percentages reported for tests greater than PS50 are only for the 9 nodes participating in those tests. The incorrect lengths are caused by packets that contained a REJOIN message or those in which a failed message's payload was appended to a new message. The results show that a high percentage of "normal" packets were transmitted, and hence the occurrence of status messages and appended packets was low enough to not have a significant effect on the calculations.

Table 4.6. Packet lengths achieved in tests, calculated over 574 received packets per node.

Measurement	Specified length achieved (%)	Lengths used to estimate
Baseline	97.74	2-5
PS10	93.74	8- 10
PS15	94.83	12- 15
PS20	93.82	
PS25	92.35	
PS30	97.01	
PS35	95.65	
PS40	89.78	
PS45	89.89	
PS50	94.23	
PS100	91.29	
PS150	90.81	
PS2200	98.06	
PS241	83.47	

4.5.3 Performance per group

Figure 4.12 shows how the PDR changed for various application payload lengths, with the Baseline measurement representing a payload of 5 bytes. For all three groups, the influence of payload length on the PDR is not pronounced. Whilst the Near and Far group's PDR did improve noticeably during PS200 and PS241, the PDR was only calculated over the subset of 9 nodes participating in these tests and is thus not a reflection on the entire Near and Far group's performance.

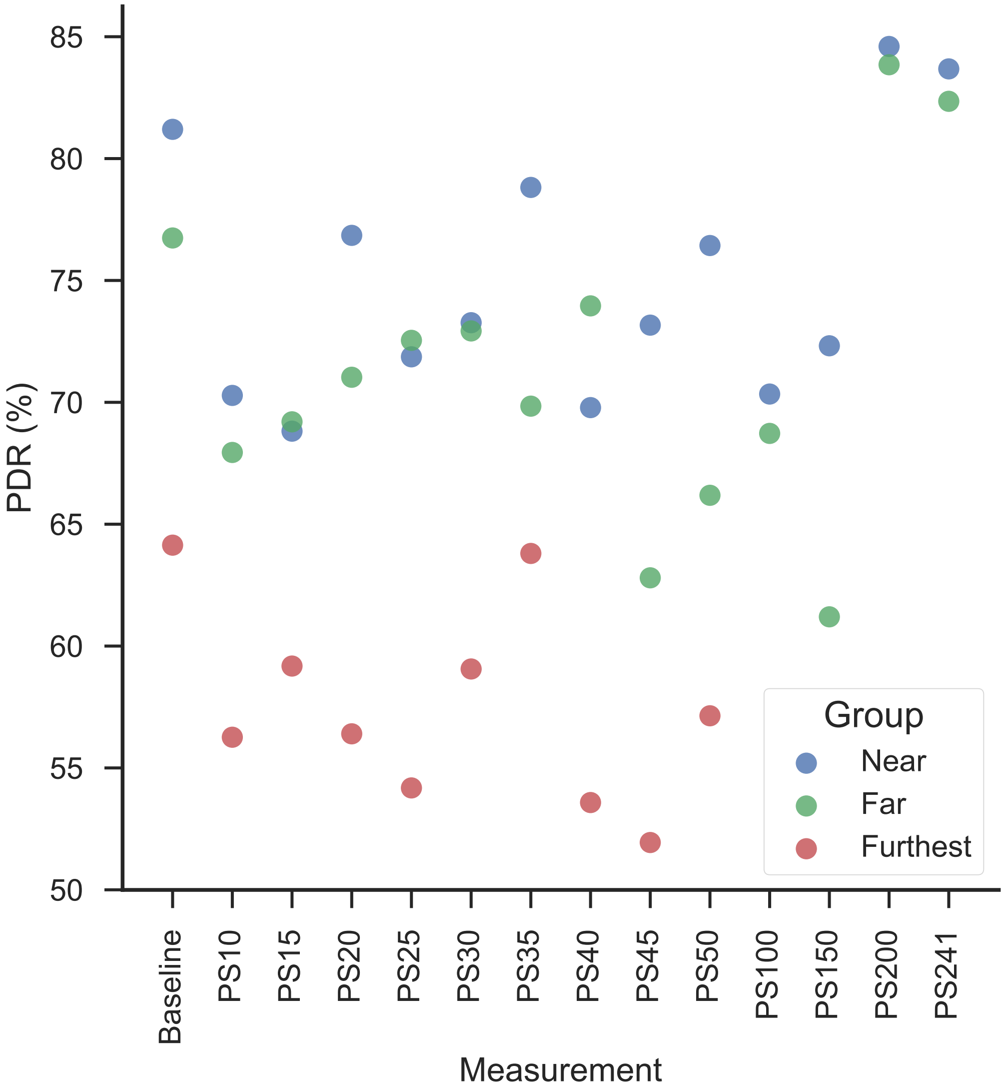


Figure 4.12. PDR over several application payload lengths (Baseline serves as PS5).

This non-reflection is confirmed when the difference between the average of the PDR over all nodes is compared with the average PDR for the subset version of each group as shown in Figure 4.13. Zero on the y-axis therefore represents an absolute 0 % difference from the average of the PDR over all nodes. The calculated PDR using the subset is nearly always higher than when all the nodes are used. The Near group has a lower PDR than the Far group, which is contrary to the PDR trends in Figure 4.12 in which, as expected, the PDR of the Near group is higher than that of the Far group.

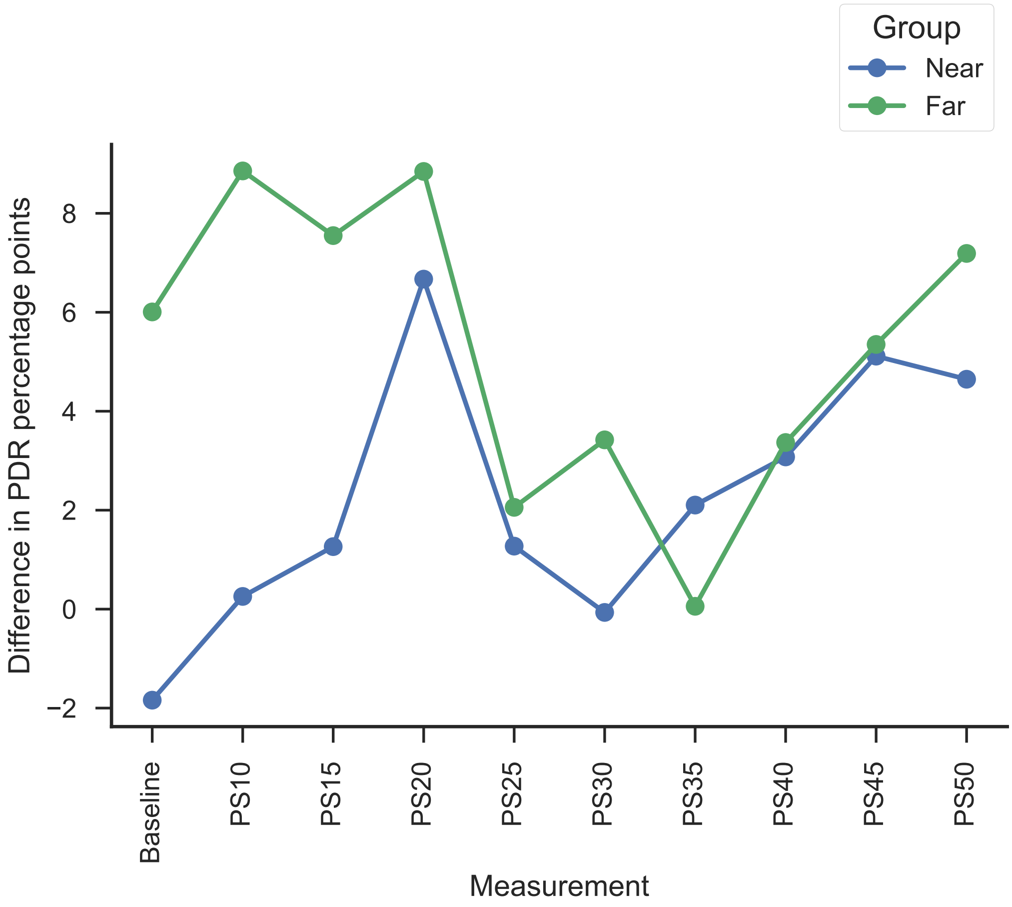


Figure 4.13. Difference in PDR for groups consisting of the chosen 9 nodes and the full groups.

4.5.4 Performance of a subset of 9 nodes

Another approach would be to only plot the PDR for the subset of 9 nodes as they were used in every test. Figure 4.14 shows the results of this analysis. As only the chosen 9 nodes were used, the Furthest group were completely excluded and the Near group excluded a6-59 whilst the Far group excluded eb-98, e5-44 and e5-4b. Whilst the calculated PDR is in general higher, as confirmed by Figure 4.13, no clear patterns can be observed. Both extremes in packet length (5 and 241) did perform quite well for both groups in comparison to other lengths, but this increase in PDR is likely due to a combination of factors and not solely payload length.

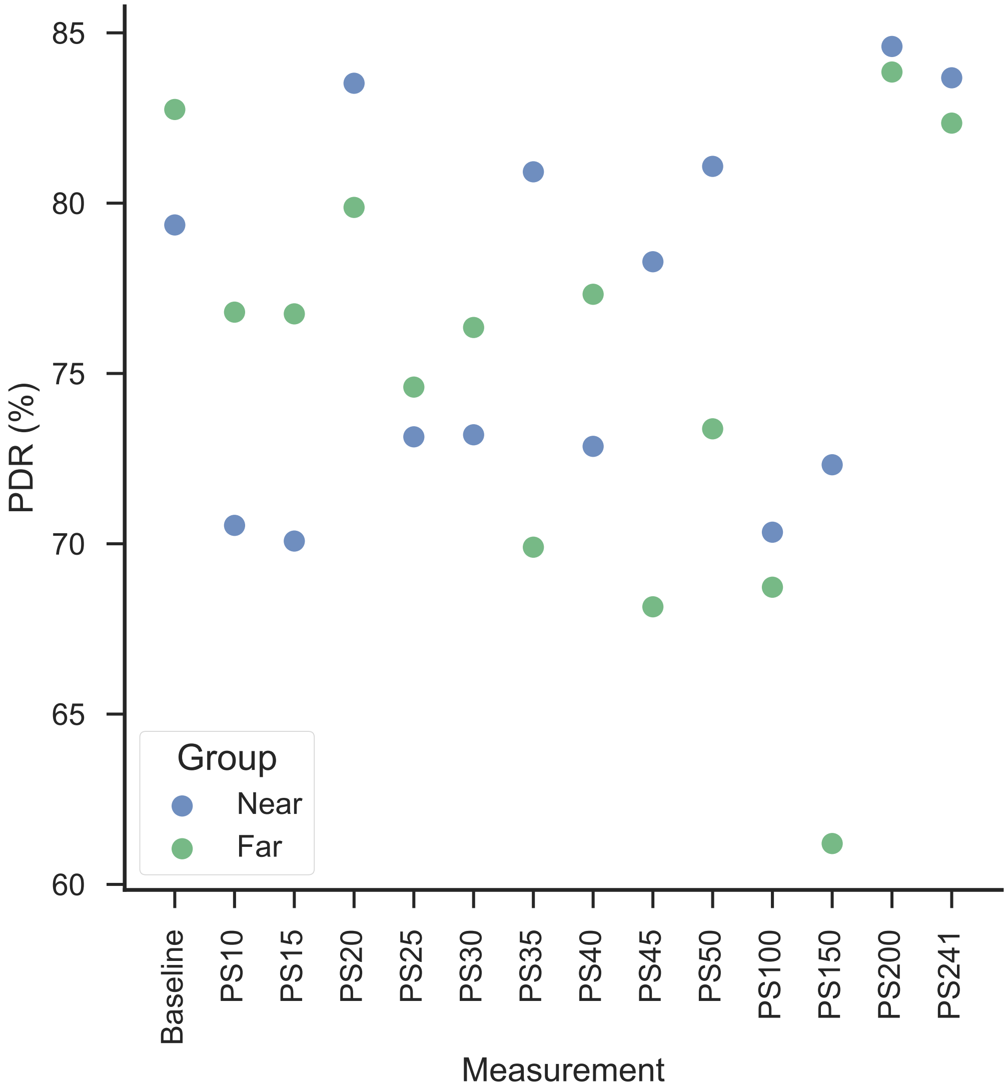


Figure 4.14. PDR for the chosen 9 nodes over several application payload lengths.

4.6 IMPACT OF WAITING TIME ON PDR

The PDR of a node can also be influenced by the interference caused by other nodes that influence the successful reception of the node's packets at the gateway. An experiment was conducted in which the time interval between packets was set to 30, 45 or 60 seconds. The results are plotted in Figure 4.15 and show no significant change in the PDR. The Baseline configuration served as the 60 second measurement. The PDR can be influenced by the gateway attempting to decode colliding signals [49], but as the testbed only has 18 nodes, it cannot serve as a full scale test of collisions. Each gateway can support thousands of mDots [71], but due to cost considerations, simulations are the preferred method of investigating the impact of collisions [49].

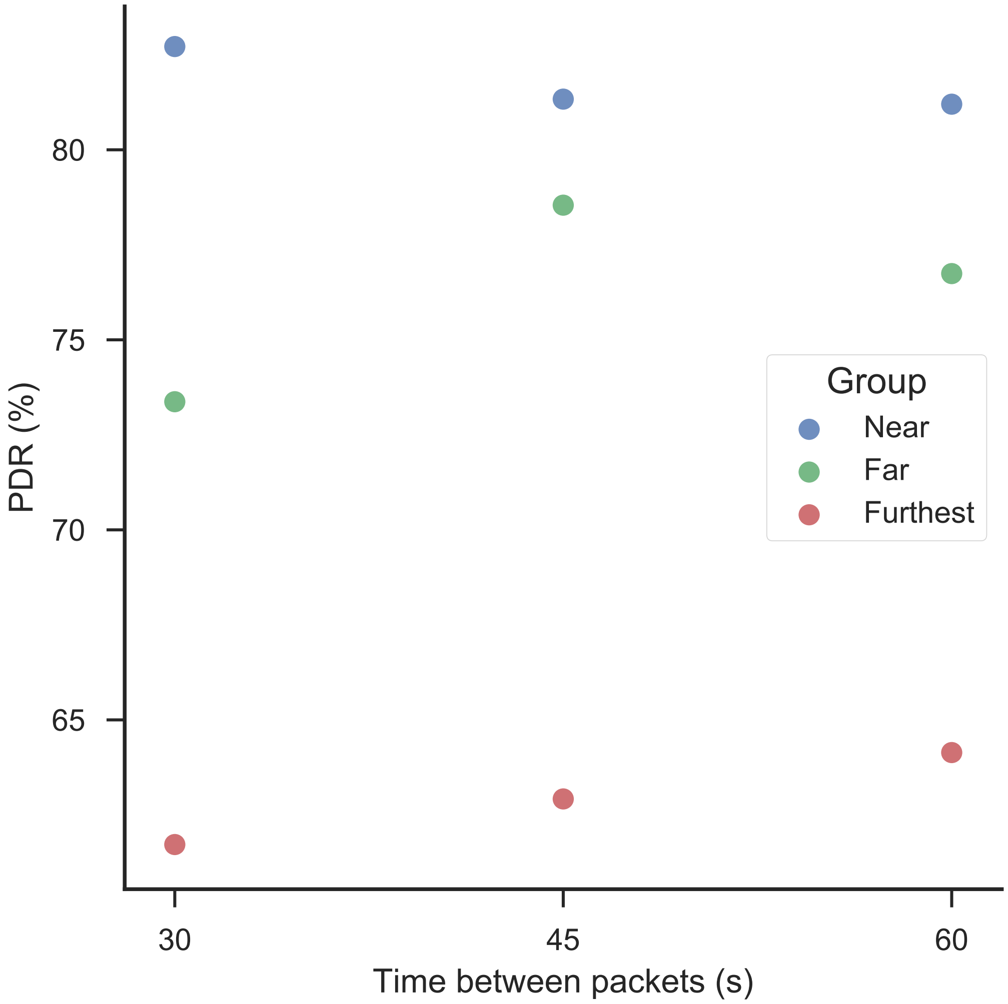


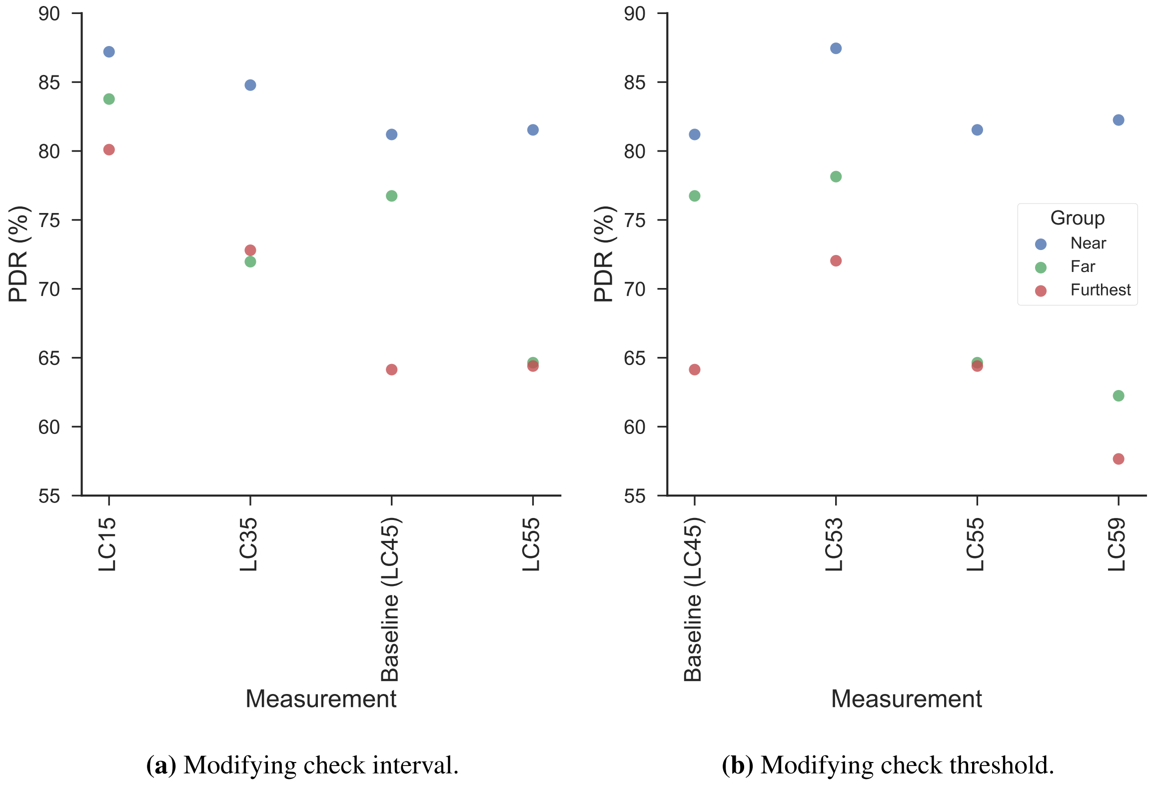
Figure 4.15. PDR over several inter packet time intervals.

4.7 IMPACT OF LINK CHECK ON PDR

To ensure scalability, the LoRaWAN nodes are normally set to minimise the use of acknowledgements when sending packets. As a result, nodes do not receive information from the gateway regarding successful reception of packets. The network could also consider a node to be disconnected without the node being aware of this status. Nodes can, however, perform periodic network link checks to confirm that they are still connected to a LoRaWAN. The node requests a link check and a gateway will respond indicating that the node is still connected [6]. The mDot provides the capability to automate the periodic link checks. The user application must specify how often these checks should be performed, for example, every third packet, and the number of link check failures that must occur before the network connection is considered to be lost. This is due to the fact that the LoRaWAN protocol does not guarantee 100 % packet delivery. In practice, a single link check could fail but the node would still be connected.

The impact of link checks on the PDR was investigated by changing both the link check interval and the number of failed checks before the node is considered disconnected. The Baseline measurement used the default values of checking every 4 packets and considered the node disconnected if 5 of these checks failed. The check interval changed as follows: every single packet, every third packet, and every fifth packet. The check threshold was kept constant at the value used in the Baseline. In addition, the influence of setting the number of checks for failure to 3, 5 and 9 was investigated, with the checking interval kept constant at every fifth packet.

Figure 4.16 shows how the PDR was impacted by changing either the checking interval or the threshold. In 4.16(a), the PDR stays fairly stable for the Near group, with the other groups showing larger deviations from their Baseline performance. In 4.16(b), the PDR again stays similar for the Near group, and the Far group showed a 10 percentage points drop, which was the largest deviation. The Furthest group showed a slightly smaller deviation.



(a) Modifying check interval.

(b) Modifying check threshold.

Figure 4.16. PDR for various link check options.

4.8 IMPACT OF ACKNOWLEDGEMENTS ON PDR

To enable large scale networks, LoRaWAN's approach is to strongly suggest that nodes only send unconfirmed packets, but the protocol does allow a node to send a confirmed data message to which the network server will try to reply with an acknowledgement. Acknowledgements are only sent once only in response to the latest packet received [6].

Data is sent to the gateway via a send function provided by the Multitech mDot library. The function returns a value indicating if the packet was successfully sent, or when ACKs are enabled, if an ACK was received. The Multitech mDot library allows the application to specify how many retry attempts must be performed if an acknowledgement was not received after the first transmission. The send function will retransmit the packet up to the specified amount of times before the send function's return value indicates failure.

Performing retransmissions increases power consumption and reduces throughput. Figure 4.17 shows how the amount of time it takes for all nodes in a group to send 1000 packets increases when the number of retries is increased. The zero options for the number of retries allows the node to request an acknowledgement, but if it is not received, the packet is not automatically retransmitted and the send functions indicate failure. The user application running on the mDot can then decide to resend the application payload or not to resend the payload. For the Near group, the number of retries had little effect on the time duration of the tests, whilst for the Furthest group, a large impact on test duration was observed.

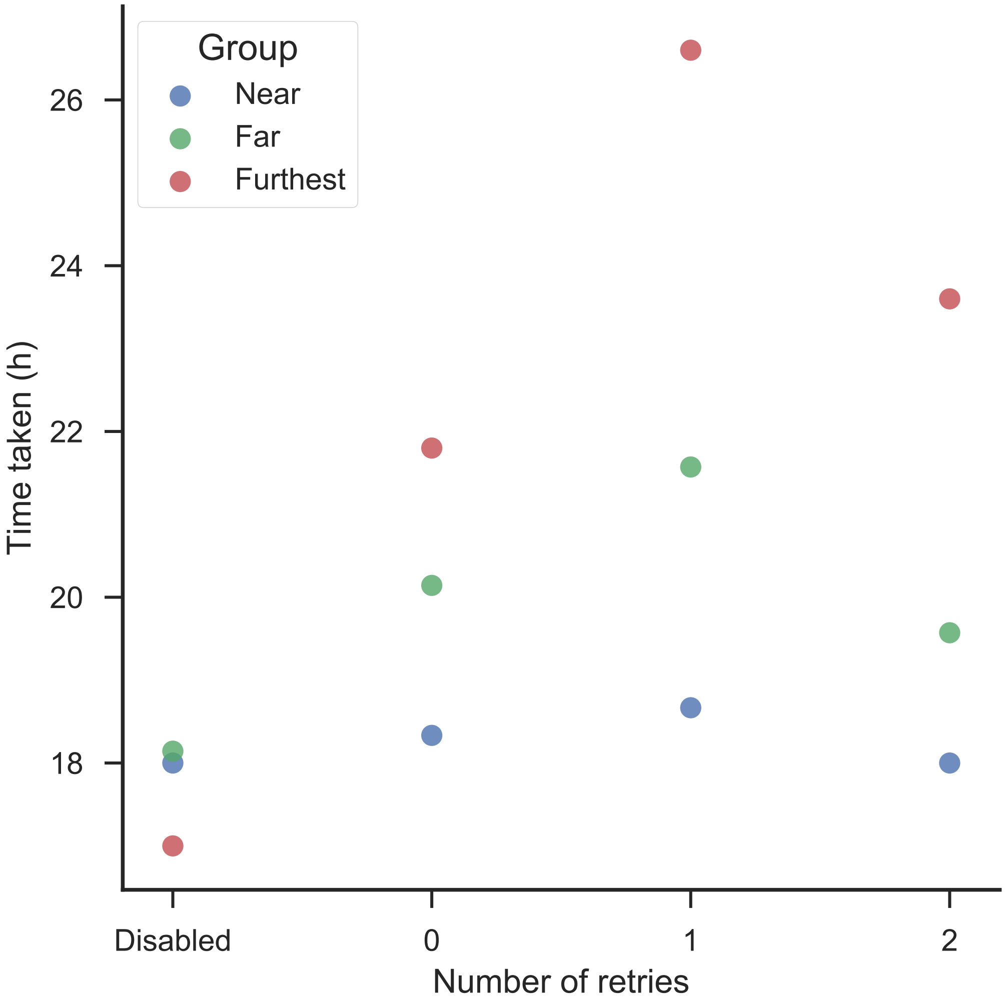


Figure 4.17. Time taken to reach 1000 sent packets for different acknowledgement options.

Figure 4.18 shows how the PDR improved when packet acknowledgement was enabled. The PDR improved for all three groups, with the Furthest group showing the biggest improvement. This group's PDR increased by almost 30 percentage points for 2 retries when compared to disabling acknowledgements. The improvement in PDR comes at a cost: the gateway must transmit an acknowledgement for every incoming packet and during this time it is unable to receive packets. A packet may also be received but the node did not receive the gateway's acknowledgement, causing it to retransmit unnecessarily. The testbed is a small scale experiment relative to the actual use scenario of a LoRaWAN gateway which will likely support thousands of devices. This high load of nodes makes enabling acknowledgements difficult in practice.

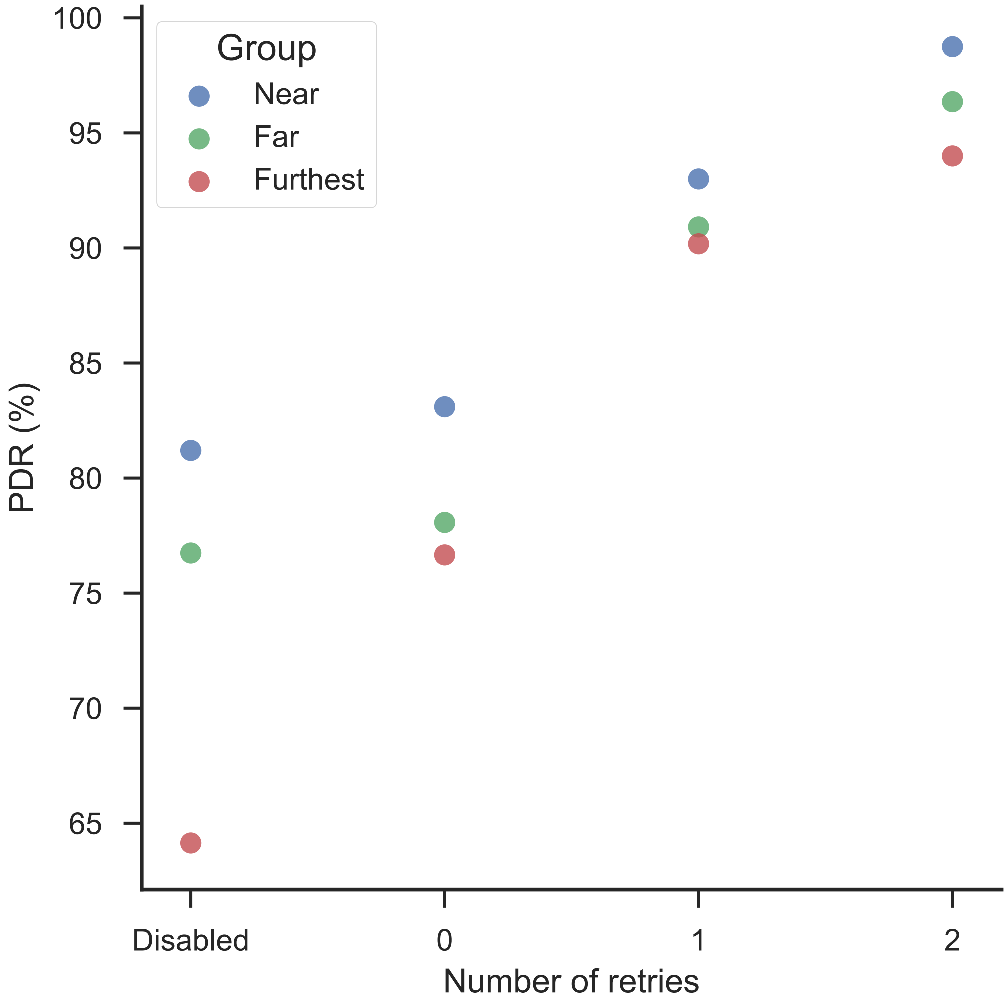


Figure 4.18. PDR for several acknowledgement options.

4.9 CHAPTER SUMMARY

In this chapter, the results of several experiments conducted with the LoRaWAN testbed were presented. An overview of the data obtained from the experiments and the technical issues experienced was provided in Section 4.2. Each experiment consisted of several tests, each stored in a measurement, which is an InfluxDB data structure. A Baseline node configuration with its performance captured in a measurement served as the point of comparison for experiments. This was followed by the results of each experiment.

Section 4.3.1 examined received LoRa packets to develop an understanding of the SNR and RSSI values of successfully received packets. This section found that the impact of distance on signal strength can clearly be observed when the SNR and RSSI values are examined for the different groups. The long whiskers of the box plots showed the wide range of RSSI and SNR values captured. This indicated that determining a node's group based on a single RSSI or SNR value would be highly inaccurate. When the recorded RSSI and SNR values were inspected on an individual node basis, a variety of distribution shapes were observed, with some nodes exhibiting more spread out values with respect to the mean, with others exhibiting small clusters of values. This variation in the distributions at node level was small compared to the overall distribution of values for a group. As expected, when the average RSSI and SNR values of multiple measurements were examined, a consistent trend with respect to distance was observed between the three groups.

ADR is a prominent feature of a LoRaWAN, and this feature's impact on performance was the focus of Section 4.4. ADR allows the network to adjust the data rate (a SF/BW combination) as well as the Tx power on an individual node basis. The PDR was consistently worse when ADR was enabled and the data was examined to determine why. Figure 4.7 revealed that the ADR assigned just two options, namely SF12 or SF7BW250, in the case of the Near and Far groups. These two options were also dominant in the Furthest group's assignments. When the Furthest group's data was examined more closely, it was found that only a few packets were sent using SF8 and SF7BW125, and that the split between the dominant options of SF12 and SF7BW250 differed substantively between nodes in this group.

The consistency of data rates chosen by the ADR scheme was investigated in Section 4.4.3. The results showed that SF12 and SF7BW250 remained the most assigned options in all of the 20 measurements, with a small variation in the split between these two. Whilst the other data rates were rarely used, all of the data rate options were used when all 20 measurements were investigated. The most underutilised spreading factors were SF11 and SF10 whose highest assignment percentages were only 0.98 % and 1.29 % respectively.

The experiment in Section 4.5 into the impact of application payload length on PDR required some protocol and firmware issues to be considered before the experiment could be conducted. The LoRaWAN protocol has maximum application length limits that must be adhered to and the addition of MAC commands to the payload must also be considered. The firmware also sends standard status messages and a Node.js script was used to determine that the impact of these messages on the packet length tests is not significant. Tests of application payloads with lengths of 5 bytes up to 50 bytes were conducted using all 18 nodes. Tests of longer payloads were done with a subset of 9 nodes capable of adhering to the legal limits. The effect of payload length on the PDR was not very pronounced, even when the PDRs of this subset of 9 nodes were considered which were, of course, present in all 14 tests.

The sending interval between packets was the focus of Section 4.6. Three sending intervals, namely 30, 45 and 60 seconds, were examined and no significant change in the PDRs were found. As the testbed only contained 18 nodes, the impact of packet collisions and nodes interfering with each other could not be fully captured.

Link checks are another LoRaWAN feature and their impact on PDR was examined in Section 4.7. Nodes periodically request link checks to determine their connectivity status, and the interval of these checks and the number of failed checks before connectivity is considered lost can be modified. The Near and Far group had little change in the PDR for the various link check configurations examined, but the Furthest group did see a small PDR decrease when the interval was increased.

Section 4.8 examined the impact of nodes sending confirmed packets, which requires an acknowledgement from the gateway when received. The impact of retransmitting a packet either once or twice versus only informing the application that an ACK was not received was also investigated. Retransmitting packets when an ACK is not received takes time, and this was confirmed when the time it took nodes to send 1000 packets was plotted for several acknowledgement options. Retransmissions increase a node's power consumption. The PDR improved for all three groups when ACKs were enabled and the Furthest group showed the largest improvement. Increasing the number of retransmission attempts did result in an increase in the PDR, but would likely be unfeasible in a large LoRaWAN. A gateway cannot receive incoming packets when transmitting acknowledgements, and if a gateway supports thousands of devices, this decrease in effective receiving availability will reduce the total number of nodes that can be accommodated.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

This research project set out to build a LoRaWAN testbed and to execute a performance evaluation of this new technology. Chapters 3 and 4 described how the testbed was built, and presented the experimental results gathered with this testbed. This chapter discusses the main findings of the completed work, compares it with existing literature examining performance and relates it back to the research questions posed in Chapter 1.

The research objectives completed in this work are as follows. The first was to determine how effective a LoRaWAN would be for nodes at larger distances (> 1.5km) from a gateway in an urban environment. The second was to determine how the Adaptive Data Rate (ADR) scheme impacts the performance of a LoRaWAN. The final objective was to determine the impact of other LoRaWAN parameters on the performance of a LoRaWAN.

A baseline node configuration was developed, and a set of experiments was executed in which each experiment revealed the performance impact of variation in a single LoRaWAN parameter. The first experiment measured and reported RSSI and SNR values, and this was followed by experiments examining the impact of ADR, payload length, waiting time, link checks and, finally, acknowledgements. To see the effect of distance on performance, the results were examined on a group basis (Near, Far and Furthest). In some experiments, the results for individual node(s) were also investigated.

5.2 PERFORMANCE OF LONG RANGE NODES

5.2.1 The experimental context

An objective of the testbed was to not only evaluate the performance impact of LoRaWAN parameters, but to also examine the performance of long range nodes in urban environments. When this testbed was constructed, the existing work was dominated by close range and very brief performance evaluations. The testbed thus was designed to be a permanent installation to allow for evaluations over long time periods and with some nodes deliberately placed at further distances from the gateway. Due to cost constraints, the testbed was limited to 18 devices and these were carefully placed as to maximise their effectiveness when evaluating the impact of distance on performance.

Performance can be measured using various metrics. As the testbed only consists of a few devices, using it to measure a LoRaWAN's scalability by accurately capturing congestion was not feasible. Other performance metrics such as energy efficiency require specialised measuring equipment and are heavily impacted by other component choices, for example, solar panel and battery, in addition to the energy consumption of the LoRa radio. Instead, the Packet Delivery Ratio (PDR) was chosen as the best performance metric as this could be accurately measured and the delivery of data is one of, if not the, primary goal of a WSN.

In addition to examining the PDR, the RSSI and SNR values captured of received packets can also be used when evaluating the performance of long range nodes. It is known that, in WSNs, a transmitted signal's strength will not only decrease due to free space path loss but also by multipath propagation. LoRa's use of CSS modulation improves its resistance to not only multipath but also interference and the Doppler effect [75]. The RSSI and SNR values can provide an indication of a node's immediate environment which helps when examining its PDR.

5.2.2 Findings

The RSSI and SNR experiment, detailed in Section 4.3, demonstrated the receiver sensitivity achievable with LoRaWAN. The lowest RSSI and SNR values were -125 dBm and -19.2 dB. By first presenting the results on a per group basis, the impact of distance on node transmissions can be seen more clearly than examining each node individually. This impact was observed when the SNR and RSSI values were examined for the different groups. The long whiskers of the box plots showed the wide range of RSSI and SNR values captured. When the RSSI and SNR values for individual nodes were examined, a variety of distribution shapes were observed, with some nodes exhibiting more spread out values with respect to the mean, and others exhibiting small clusters of values. The Furthest group, consisting of 5 nodes, can be considered as long range nodes and this is reflected in their low RSSI and SNR values.

Figures 4.2 and 4.3 showed how a node's immediate environment can impact signal strength. As an example, e5-40 is 2.92 km closer to the gateway than e5-4a, but due to other factors such as position, height, and the immediate surroundings, node e5-40 has lower RSSI values in general. In the Furthest group, the difference between e5-40 and e5-4a RSSI values is repeated in their SNR values (Figure 4.3). Another reason why a node that is located further away would perform better than a closer one is due to constructive/destructive interference. The constructive or destructive effects of multipath on the signal is dependent on the obstructions between the node and the gateway and not on the distance [76]. Whilst eb-99 and e5-42 are within 170 m of each other and e5-42 is on a considerably higher building (8 floors higher), its location behind a high parapet on the roof compared to a clear roof for eb-99 had a significant negative impact on its SNR.

The impact of distance on the PDR was also seen in Figure 4.6, which shows that most nodes closer to the gateway had higher PDRs than nodes much further away. The difference in PDR between the Near and Far groups was small as all of these nodes were less than 1.3 km from the gateway. This figure also shows that distance does not necessarily cause the biggest drop in PDR. Activating the ADR scheme caused the average PDR for the Furthest group to drop by 12.5 percentage points from 76.62 to 64.14, and Section 4.4 revealed the cause to be the assignment of SFs to these nodes.

Whilst distance certainly has an effect on long range nodes PDR, sub-optimal LoRaWAN settings could have a much bigger impact on network performance. Whilst activating the ADR scheme did reduce the average PDR for the Furthest group substantially, the decrease was not the same for all nodes. The biggest decrease was 35.4 percentage points for e5-40, resulting in a PDR of only 49,2 %. This is a stark contrast to eb-97 and e5-42, which showed a change of less than 2 %. A node's PDR cannot be predicted accurately by simply examining the RSSI and SNR values of received packets. The correlation between RSSI and PDR is not high, although SNR correlates to a larger extent with

PDR [77, 76].

Another consideration, especially for longer range nodes is that the testbed's performance was measured by calculating the PDR for packets sent from the nodes to the gateway (uplink traffic). However, wireless networks are known to have link asymmetry, that is, the amount of connectivity differs between the uplink and the downlink [76]. Therefore, the PDR calculated in each experiment is a reflection of uplink network performance and the downlink performance would differ. The LoRaWAN protocol does allow for downlink transmissions, but these should be kept to a minimum to ensure the overall network performance does not suffer. When a LoRaWAN is deployed, care should be taken to ensure that connectivity levels are acceptable for uplink and downlink traffic. Firmware over the air updates requires that the downlink performance is sufficient.

The results showed that, whilst nodes located at distances greater than 1.5 km for the gateway do have lower PDRs than closer nodes, they can still communicate fairly effectively. Each IoT use case has its own unique challenges, and if long range communication is required, LoRaWAN should only be used if a PDR of between 60 % and 80 % is acceptable. Furthermore, careful network planning should be performed to ensure that gateways are as high as possible and that node placing are performed with radio performance in mind. Finally, distance is not the only factor impacting performance; all aspects, especially LoRaWAN parameters, should be considered when deploying a network with long range nodes.

In the testbed's case, disabling the ADR scheme and using a fixed SF of 12 would have resulted in better performance. Additionally, as the testbed was a small network enabling acknowledgements and allow for two retries would also have increased the PDR for long range nodes.

Obtaining the optimum LoRaWAN parameters would have to be performed on an individual network basis, as the radio environment and geographical layout will differ between networks. As highlighted in [12], automated mechanics will allow networks to determine their own optimum settings.

5.3 PERFORMANCE IMPACT OF ENABLING THE ADR SCHEME

The PDR was consistently worse when ADR was enabled and the data was examined to determine why. It was found that the ADR dominantly assigned certain spreading factors, mainly SF12 and SF7BW250, whilst almost ignoring the other options. The ADR algorithm would also have made adjustments to the Tx power used by nodes, but unfortunately the power with which each packet was transmitted was not logged as the Multitech gateway does not have this as an option. When the data from multiple measurements were examined, they showed that the ADR scheme remained consistent in its SF assignments. This suggests that this behaviour is not the result of a specific packet size, acknowledgement setting, link check setting or packet sending interval (at least not the ones that were examined).

When the spreading factors used by e5-42 from the Furthest group were examined in Figure 4.9, it showed how the algorithm abruptly changes between SF7, SF7 and SF12. This dramatic change occurs for all nodes in the Furthest group but the split between SFs differs.

A likely cause of the ADR scheme's poor performance was that its algorithm only used one input variable, namely SNR, when determining a node's configuration. Section 4.3 demonstrated how variable a node's SNR can be and thus SNR should not be the sole input for the algorithm.

Whilst the data showed a reduction in PDR when ADR is enabled, it remains a useful feature. When ADR is disabled and long range nodes are incorrectly setup with a high data rate, which is not designed for long range, a situation occurs in which a node can join the network but cannot effectively communicate. The joining procedure will adjust the SF used during the joining process [6], but once the node is joined, it will revert back to the user defined SF. With this high data rate, it experiences very high packet loss and will attempt to rejoin the network. This cycle repeats with the node being able to join the network but not being able to reliably send data.

The algorithm used by The Things Network was studied in detail in [66], which also includes suggested improvements. One addition proposed to the publicly available algorithm used by The Things Network is to add a component to help combat hysteresis. It was noted that when the link margin estimate is marginal between two assignment decisions oscillations can occur as ADR requests are repeatedly sent instructing the node between different configurations.

This oscillation was observed in the testbed; Figure 4.9 shows how node e5-42 abruptly varies between SF7, SF8 and SF12. Their identified weakness appears to be also present in Multitech's implementation of ADR.

The effective use of SFs is key to ensure network scalability, because communication on different SFs are orthogonal to each other. This allows a gateway to receive them simultaneously and thus increases the capacity of a gateway. However, scalability requires more than just ensuring that spreading factor use is balanced evenly. Long range nodes can not use all of the available SFs, as the choice of spreading factor provides a trade-off between range and throughput.

Improvements to the ADR scheme is an active research area. Researchers are investigating different approaches such as providing ADR algorithms with more inputs or changing the algorithms themselves. Additionally, the optimisation attempts are generally focused on one optimisation goal such as scalability, and might decrease other areas of performance such as energy efficiency or throughput.

5.4 IMPACT OF OTHER LORAWAN PARAMETERS

In addition to the ADR scheme, the performance impact of the following LoRaWAN parameters was also investigated: payload length, waiting time, link checks and acknowledgements. The effect of payload length, link checks and waiting time on the PDR was not very pronounced. Enabling acknowledgements did improve the PDR for all three groups but has some caveats around scalability. The impact of each parameter will now be expanded upon.

The main factor that had to be considered when the impact of payload length was investigated was the LoRaWAN protocol's maximum application payload lengths. These maximums depend on the SF/BW combination used and if MAC commands need to be transmitted alongside the payload. As a result, not all nodes could participate in tests of payload lengths longer than 50 bytes, thus tests of longer payloads used a subset of 9 nodes. These results showed no significant PDR changes between different payload lengths. Increasing the payload length would increase a node's transmission time, which can be reduced by using a faster data rate. When the impact of two payload lengths were evaluated in [64], a definite PDR improvement was observed. This improvement was, however, not consistent over the range of data rates evaluated and the payload lengths experiments conducted in [57] noted similar PDRs for 10 and 100 bytes but a decrease for 50 bytes.

As the results showed no negative impact of sending longer payloads, the data can be aggregated and sent as one large payload if the use cases allow for delayed data. The testbed consisted of only 18 devices so the interference impact of longer but less frequent transmissions versus smaller but more frequent transmissions could not be investigated.

The frequency with which nodes send packets (waiting time) was also investigated; sending packets more frequently would increase power consumption and increase the chance of collisions. The experiment revealed no major PDR changes, but as the testbed only contained 18 devices, the congestion levels cannot be fully captured. Simulations performed in [14] revealed that the PDR impact remained small in networks of less than 5000 devices when waiting times of 600 s, 6 000 s and 60 000 s were evaluated. Congestion, scalability and collision based research is best done with simulators as building a LoRaWAN testbed with thousands of devices would be costly. Research performed through simulations found that performance drops significantly for event-driven traffic [50], downlink traffic has a high scalability cost [47], scalability should be examined on a use cases basis [48] and that deploying multiple gateways can better mitigate the performance impact of interference than the use of directional antennae [36].

The LoRaWAN protocol allows nodes to validate their connectivity periodically through link check commands. The frequency of these checks (every <x> packets) and how many must fail before the network connection is considered lost can be configured. Several values for both of these options were tested and showed no significant changes in PDR. There is a trade-off between immediately identifying connection issues and being too strict with the disconnectivity threshold value. As the LoRaWAN protocol was not designed for > 99 % packet delivery, some packet loss is expected. A too strict criteria would cause nearly any packet loss to be identified as connectivity loss. There is a possibility that the link check requests to the gateway or its response are not successfully received by their intended recipient, causing additional time-consuming reconnection attempts. The reconnection procedure takes a lot of time and energy, not only reducing the time available to sent packets but also wasting energy.

Acknowledgements were the last LoRaWAN parameter examined and it did show that a small network, such as the testbed, can improve node PDRs using acknowledgements. To ensure scalability, nodes normally send all uplinks without requiring that the network acknowledge reception (unconfirmed). Acknowledgements can however be requested (confirmed uplinks) and the network will attempt to send an acknowledgement. The mDot library supplied by Multitech allows the number of retries if an ACK was not received to be set, and increasing this improved the PDR. Power consumption would have increased as nodes increased retransmission attempts. This improvement matches the simulation results presented in [14], however the researchers note that this improvement is only present in small networks which send data infrequently.

A LoRaWAN gateway can receive multiple transmissions simultaneously (if on different SFs) but can only transmit on one channel at a time and cannot receive any transmission whilst doing so. Thus, acknowledgement requests will negatively impact the PDR of the network as a whole if the gateway must frequently enter transmission mode while serving a large number of nodes. The testbed consisted of only 18 nodes, and thus the PDR improvement of enabling acknowledgements was seen without the downside of enabling acknowledgements.

5.5 CHAPTER SUMMARY

This chapter set out to discuss the experimental work performed with a LoRaWAN testbed in order to achieve three research objectives. These objectives were:

- Determine how effective a LoRaWAN would be for nodes at larger distances (> 1.5km) from a gateway in an urban environment.
- Determine how the Adaptive Data Rate (ADR) scheme impacts the performance of a LoRaWAN.
- Determine the impact of other LoRaWAN parameters on the performance of a LoRaWAN.

A performance metric was required before the research objectives could be met. The Packet Delivery Ratio (PDR) was chosen as the best performance metric as this could be accurately measured and the delivery of data is one of, if not the, primary goal of a WSN. Additionally, a baseline node configuration was developed, and a set of experiments was executed in which each experiment revealed the performance impact of the variation of a single LoRaWAN parameter.

The impact of distance on signal strength was investigated by measuring the RSSI and SNR values of the groups of nodes at three different distances. The Furthest group contained 5 nodes placed more than 1.5 km away from the gateway and were considered to be long range nodes. A wide range of RSSI and SNR values were captured which showed a variety of distribution shapes. The data showed that the nodes located furthest away could have better signal strength than nodes slightly closer due to factors such as position, height and multipath influences. Enabling the ADR scheme resulted in the PDR for this group to drop by 12.5 percentage points as the scheme assigned SFs to optimise throughput and not improve the success of individual packet delivery. Distance did have an impact on the PDRs, but this is just one performance metric among many. The complete set of LoRaWAN parameters and the relevant performance metrics should be considered for networks containing long range nodes. The Furthest group was able to communicate fairly successfully. The definition of an acceptable level of success is dependent on the IoT use case. For many use cases, a LoRaWAN can be used with a PDR of between 60 % and 80 %.

In general, as shown in the literature review, there are performance trade-offs required in the development of LPWANs and LoRaWAN is no exception. Long range nodes require not only large coverage ranges, but frequently also high energy efficiency, as nodes may be placed in remote and difficult to access areas. In the LoRaWAN protocol's case, these features are traded-off against non-guaranteed packet delivery and a low throughput.

The LoRaWAN protocol's ADR scheme was the focus of the second research objective. This scheme aims to optimise network throughput by individually adjusting the data rate and transmit power of a network's nodes. LoRa's CSS modulation has the highly desirable feature that packets sent at different data rates can be received simultaneously by the gateway, increasing the network's scalability.

The ADR scheme sends MAC commands to nodes requesting adjustments and an acknowledgement will be sent by a node to confirm its new settings. The LoRaWAN specification does not specify how the network manager must determine the settings that should be sent as this could be a competitive advantage and vendors keep their ADR algorithm implementations private. Multitech's description of their implementation states that it uses the SNR from each packet to calculate a potential data rate. The calculated values for the most recent 11 uplinks of a node are used in conjunction with a threshold value to determine the maximum data rate that meets the threshold and will thus be sent to the node.

The captured data from the testbed showed that the PDR was consistently worse for all groups when the ADR scheme was enabled. The scheme dominantly assigned either SF12BW125 (the minimum data rate) or SF7BW250 (the maximum data rate) and this behaviour remained fairly consistent and was not influenced by a specific packet size, acknowledgement settings, link check settings or packet sending interval.

The ADR scheme is a popular topic in LoRaWAN research. Several researchers have proposed improvements with the aim to optimise a certain aspect of the network such as scalability, throughput or PDR. When the ADR scheme implemented by The Things Network was examined in [66], a potential oscillation in the decision between data rates was identified. This oscillation was observed in the testbed, and Figure 4.9 showed how a node abruptly changes between SF7, SF8 and SF12. This suggests that Multitech's implementation could also suffer from this oscillating behaviour in assigning data rates.

As the testbed showed, there is definitely room for improvement in Multitech's current ADR scheme (at least in the version used by the testbed). The data showed that SF assignments were not equal, the assignments showed an oscillation between SF selections and the PDR of several nodes suffered as a result. For a small scale network such as the testbed, disabling ADR would have been an improvement. However, this is not scalable and this key feature of a communication protocol should not be ignored but rather improved. Other researchers have also engaged with this element of the LoRaWAN protocol and multiple approaches are being developed and tested as referenced above.

The performance impact of individual LoRaWAN parameters was investigated and the effects of payload length, link checks and waiting time on the PDR were not significant. Enabling acknowledgements did show a significant improvement in the PDR but is a result of the low number of nodes in the testbed and acknowledgements, which force the gateway to transmission mode, would be detrimental to the performance in larger networks. Therefore, the use of acknowledgements to improve the PDR would depend on the network's size and if gateways experience congestion. A public LoRaWAN network might need to support thousands of nodes but a private network operating in a remote area with only a few nodes could turn on acknowledgements to achieve a high PDR.

Further insights can be found by testing combinations of the LoRaWAN parameters but this is outside the scope of the current research. As would be expected, finding the right combination of parameters to satisfy performance requirements is difficult [12].

CHAPTER 6 CONCLUSION

6.1 SUMMARY OF WORK CONDUCTED

The IoT aims to improve multiple sectors of our society through the use of networked sensors and actuators which enables optimisation and automation in these sectors.

This work set out to examine LoRaWANs, a newly released LPWAN technology purposefully designed for IoT devices. Specifically, this work focused on doing a performance evaluation of a LoRaWAN by building an outdoor testbed and performing experiments.

A performance evaluation of a new wireless technology can be performed in various ways and, to focus the research, three research questions were developed and answered. These questions were as follows. How effective is a LoRaWAN for nodes at larger distances (> 1.5km) from a gateway in an urban environment? How does the ADR scheme impact the performance of a LoRaWAN? What are the impact of other LoRaWAN parameters on the performance of a LoRaWAN?

To answer these questions, a literature study was conducted and an outdoor testbed was built which consisted of 18 nodes and one gateway. Multitech is a prominent LoRaWAN vendor and their mDot modules were used in the nodes and their MultiConnect Conduit product used as the gateway. The nodes were split into three groups based on distance from the gateway. Six nodes formed the Near group (less than 0.5 km), seven nodes formed the Far group (between 0.54 km and 1.25 km) and the remaining five formed the Furthest group (between 1.98 km and 5.19 km). The nodes use a custom firmware that allows nodes to be sent AT command style commands via the gateway in order to remotely configure experiments. Additionally, a web server running Node-RED, InfluxDB and Grafana with the Node-RED application provided the user with a convenient control interface to set up and monitor experiments.

Experiments were conducted with PDR chosen as the primary performance metric. The PDR was calculated by an algorithm examining the sequence numbers of packets received by the gateway. Initial data analysis revealed that experiments could be conducted at any time and on any day of the week and each experiment was executed until each node sent 1000 packets. Each experiment consisted of several tests to evaluate multiple options for the experimental parameters. A Baseline node configuration with its performance captured in the database served as the point of comparison for experiments.

The experiments that were conducted were an investigation into RSSI and SNR values, impact of ADR, impact of payload length, impact of waiting time, impact of link checks and, finally, the impact of acknowledgements. All experiments investigated performance by examining each group of nodes (Near, Far and Furthest) independently. In some experiments, individual node(s) were also examined in more detail.

6.2 CONCLUSIONS

The first research objective dealt with how effective a LoRaWAN is when providing connectivity to nodes further than 1.5 km from a gateway in an urban environment. The testbed had 5 nodes placed at locations ranging from 1.98 km to 5.19 km from the gateway. The distance from the gateway did have a negative performance impact on this Furthest group compared to the other groups. However, distance does not necessarily cause the biggest drop in PDR. Activating the ADR scheme caused the average PDR for the Furthest group to drop by 12.5 percentage points from 76.62 % to 64.14 %. This decrease was not the same for all nodes and the reason for this decrease is discussed as part of the second research objective. As an example, the biggest decrease was 35.4 percentage points for e5-40, resulting in a PDR of only 49,2 %. This is a stark contrast to eb-97 and e5-42, which showed a change of less than 2 %.

The results show that a LoRaWAN should only be used if a PDR of between 60 % and 80 % is acceptable for long range nodes. Furthermore, there can be significant differences in the PDRs between nodes and these differences can become bigger over long distances. When a LoRaWAN is deployed, gateways should be placed as high as possible and all aspects, such as the specific physical environment of each node, but especially the LoRaWAN parameters, should be considered for long range nodes.

The second research objective was to determine how the ADR scheme impacts the performance of a LoRaWAN. When the ADR scheme was enabled, the PDR was consistently worse for all three groups. The PHY layer (LoRa) has several SF options available and choosing between these requires a trade-off between range and throughput. The ADR scheme aims to optimise throughput and, in the case of the testbed, dominantly assigned either SF12 or SF7BW250. SF12 has the greatest range and the lowest throughput whilst SF7BW250 is the reverse. As the scheme optimises for throughput, its use of SF7BW250 is expected, but attempting to maximise the throughput is futile when the data does not reach the gateway. The ADR scheme has a mechanism in which different data rates are tested to ensure that nodes do not lose connectivity when a too high data rate is used, but this does not appear to be working effectively.

How the network determines any required adjustments is not specified and vendors keep their ADR algorithms private. Due to LoRa's use of CSS modulation, the assigned data rates have an impact on network scalability and a superior algorithm is a competitive advantage. Multitech's algorithm uses the SNR from each packet to calculate a potential data rate. The calculated values for the most recent 11 uplinks are used in conjunction with a threshold value to determine a data rate which will be sent to the node.

A literature search revealed that the ADR scheme is under investigation by several researchers aiming to optimise certain performance aspects. When the ADR scheme implemented by The Things Network was examined in [66], a potential oscillation in the decision between data rates was identified. This oscillation was observed in the testbed and abrupt changes between SF7, SF8 and SF12 were observed. This suggests that Multitech's implementation could also suffer from this oscillating behaviour in assigning data rates.

In summary, enabling the ADR scheme had a negative impact on packet delivery. Nodes close to the gateway was not as severely affected as long range nodes. Certain parts of the ADR scheme's implementation is vendor specific and thus the results gathered here should be evaluated with this in mind. The ADR scheme is an active research area and the work performed here shows that more work should be done to better balance PDR and throughput.

The last research objective was to determine the impact of other LoRaWAN parameters on the performance of a LoRaWAN. There are numerous parameters and combinations, however, only the impact of payload length, link checks and waiting time on the PDR were examined and these were found to be insignificant. Enabling acknowledgements did show a significant improvement in the PDR but this was a result of the small size of the testbed (18 nodes). The acknowledgement feature forces the gateway to transmission mode, which would be detrimental to the performance in larger networks where the gateway is more congested. LoRaWAN gateways cannot transmit and receive simultaneously, and in an 18 device network the number of packets lost when the gateway is in transmission mode was very small, but this would not be the case in a large (several thousand) node network. Therefore, the use of acknowledgements to improve the PDR would depend on the network's size and if gateways experience congestion.

6.3 FUTURE WORK

The constructed permanent outdoor LoRaWAN testbed could serve as the basis for future work. The testbed currently has 18 nodes and one gateway, but can be expanded with more nodes and gateways to form a larger network. Adding additional gateways is a high priority, as the LoRaWAN protocol has several mechanisms which can only be evaluated if nodes are covered by multiple gateways.

The research evaluated performance by calculating the PDR for uplink traffic (packets sent from the nodes to the gateway). However, a LoRaWAN will also have downlink traffic in the form of acknowledgements, firmware updates and the gateway sending MAC commands to nodes. As the testbed is a wireless network, there will be link asymmetry (a difference in connectivity between the uplink and the downlink performance). A LoRaWAN's downlink performance is an important performance aspect that can be investigated using the testbed. Additionally, battery life is another performance metric that can be explored with the existing testbed.

The LoRaWAN protocol is in active development and the latest version (1.1) added support for handover during roaming and several security changes. Multitech also regularly releases updates for the gateway and the mDot library. These new changes and features can all be evaluated once the testbed is updated.

This work focused on evaluating the LoRaWAN MAC layer, the testbed is also a suitable platform to evaluate the LoRa PHY layer. Items that could be investigated include the impact of different antenna configurations and different FEC rates.

The second research objective revealed that there is room for improvement to the ADR scheme. This is an active research area and multiple researchers have suggested new techniques that should be investigated via the testbed to evaluate their performance and develop recommendations for improvement.

REFERENCES

[1] R. Sanchez-Iborra and M.-D. Cano, “State of the Art in LP-WAN Solutions for Industrial IoT Services,” *Sensors*, vol. 16, no. 5, 2016.

[2] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[3] X. Xiong, K. Zheng, R. Xu, W. Xiang, and P. Chatzimisios, “Low power wide area machine-to-machine networks: Key techniques and prototype,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 64–71, 2015.

[4] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, “Operating Systems for Low-End Devices in the Internet of Things: A Survey,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 720–734, 2016.

[5] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations,” in *IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications*, Rome, Italy, 2017.

[6] LoRa Alliance, “LoRaWAN Specification V1.1,” pp. 1–101, October 2017.

[7] M. Bor, J. Vidler, and U. Roedig, “LoRa for the Internet of Things,” in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, Graz, Austria, February 2016, pp. 361–366.

[8] LoRa Alliance, “LoRaWAN 1.0.2 Regional Parameters,” 2017.

[9] B. Reynders, W. Meert, and S. Pollin, “Range and coexistence analysis of long range unlicensed communication,” in *2016 23rd International Conference on Telecommunications, ICT 2016*, Thessaloniki, Greece, 2016.

[10] M. Lauridsen and B. Vejlgaard, “Interference Measurements in the European 868 MHz ISM Band with Focus on LoRa and SigFox,” in *IEEE Wireless Communications and Networking Conference*, San Francisco, CA, USA, 2017.

[11] J. P. Bardyn, T. Melly, O. Seller, and N. Sornin, “IoT: The era of LPWAN is starting now,” in *European Solid-State Circuits Conference*, 2016, pp. 25–30.

[12] M. Bor and U. Roedig, “LoRa transmission parameter selection,” in *Proceedings - 2017 13th International Conference on Distributed Computing in Sensor Systems, DCOSS 2017*, Ottawa, ON, Canada, 2018, pp. 27–34.

[13] Semtech, “SX1272/73 datasheet Rev. 3,” 2015.

[14] F. Van Den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.

[15] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, “Impact of LoRa Imperfect Orthogonality: Analysis of Link-level Performance,” in *IEEE Communications Letters*, 2018.

[16] B. Reynders and S. Pollin, “Chirp Spread Spectrum as a Modulation Technique for Long Range Communication,” in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, no. 2, Mons, Belgium, November 2016, pp. 1–5.

[17] M. Cattani, C. A. Boano, and K. Römer, “An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication,” *Journal of Sensor and Actuator Networks*,

vol. 6, no. 2, p. 7, 2017.

[18] R. Kirichek, V.-D. Pham, A. Kolechkin, M. Al-Bahri, and A. Paramonov, “Transfer of Multimedia Data via LoRa,” in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 17th International Conference, NEW2AN 2017*, St. Petersburg, Russia, 2017, pp. 708—-720.

[19] K. Mikhaylov, J. Petäjäjärvi, J. Haapola, and A. Pouttu, “D2D Communications in LoRaWAN Low Power Wide Area Network : From Idea to Empirical Validation,” in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, Paris, France, 2017, pp. 737–742.

[20] F. Orfei, C. B. Mezzetti, and F. Cottone, “Vibrations powered LoRa sensor An electromechanical energy harvester working on a real bridge,” in *2016 IEEE SENSORS*, Orlando, FL, USA, October 2016, pp. 1–3.

[21] A. Rahman and M. Suryanegara, “The development of IoT LoRa: A performance evaluation on LoS and Non-LoS environment at 915 MHz ISM frequency,” in *Proceedings - International Conference on Signals and Systems, ICSigSys 2017*, Sanur, Indonesia, 2017, pp. 163–167.

[22] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, “Enabling RPL multihop communications based on LoRa,” in *International Conference on Wireless and Mobile Computing, Networking and Communications*, Rome, Italy, 2017.

[23] K.-h. Ke, J.-h. Lin, and H.-c. Lee, “Demo Abstract : A LoRa Wireless Mesh Networking Module for Campus-Scale Monitoring,” in *Information Processing in Sensor Networks (IPSN), 2017 16th ACM/IEEE International Conference on*, Pittsburgh, PA, USA, 2017, pp. 259–260.

[24] S. Barrachina-Muñoz, B. Bellalta, T. Adame, and A. Bel, “Multi-hop Communication in the Uplink for LPWANs,” *Computer Networks*, vol. 123, pp. 153–168, 2017.

[25] H. C. Lee and K. H. Ke, “Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 9, pp. 2177–2187, 2018.

[26] J. Kim and J. Song, "A Secure Device-To-Device Link Establishment Scheme for LoRaWAN," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2153–2160, 2018.

[27] C. Trasviña-Moreno, R. Blasco, Á. Marco, R. Casas, and A. Trasviña-Castro, "Unmanned Aerial Vehicle Based Wireless Sensor Network for Marine-Coastal Environment Monitoring," *Sensors*, vol. 17, no. 3, p. 460, 2017.

[28] V. Sharma, I. You, G. Pau, M. Collotta, J. Lim, and J. Kim, "LoRaWAN-Based Energy-Efficient Surveillance by Drones for Intelligent Transportation Systems," *Energies*, vol. 11, no. 3, p. 573, 2018.

[29] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Malta, Malta, November 2016, pp. 59–67.

[30] B. C. Fargas and M. N. Petersen, "GPS-free geolocation using LoRa in low-power WANs," in *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, Geneva, Switzerland, 2017.

[31] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the security vulnerabilities of LoRa," in *2017 3rd IEEE International Conference on Cybernetics, CYBCONF 2017*, Exeter, UK, June 2017.

[32] S. Kwon, G. Choudhary, and V. Sharma, "An Enhanced LoRaWAN Security Protocol for Privacy," *Sensors*, vol. 18, no. 6, 2018.

[33] P. Neumann, J. Montavont, and T. Noël, "Indoor Deployment of Low-Power Wide Area Networks (LPWAN): a LoRaWAN case study," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY, USA, October 2016, pp. 1–8.

[34] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, "Thorough IoT testbed characterization: From proof-of-concept to repeatable experimentations," *Computer Networks*, vol. 119, no. June, pp. 86–101, 2017.

[35] A. G. Dludla, A. M. Abu-mahfouz, C. P. Kruger, and J. S. Isaac, “Wireless Sensor Networks TestBed : ASNTbed,” in *2013 IST-Africa Conference Exhibition*, Nairobi, Kenya, May 2013, pp. 1–10.

[36] T. Voigt, M. Bor, U. Roedig, and J. Alonso, “Mitigating Inter-network Interference in LoRa Networks,” in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, Uppsala, Sweden, February 2017, pp. 323–328.

[37] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, “Poster: A Fair Adaptive Data Rate Algorithm for LoRaWAN,” in *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*, Madrid, Spain, 2018, pp. 169–170.

[38] M. O. Farooq, D. Pesch, M. O. Farooq, and D. Pesch, “Poster : Extending LoRaSim to Simulate Multiple IoT Applications in a LoRaWAN,” in *International Conference on Embedded Wireless Systems and Networks*, Madrid, Spain, 2018.

[39] A. I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, “Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, Singapore, 2017, pp. 1–6.

[40] S. Kim and Y. Yoo, “Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN,” *Sensors*, vol. 18, no. 6, 2018.

[41] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, “Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN,” *Sensors*, vol. 18, no. 7, 2018.

[42] O. Georgiou and U. Raza, “Low Power Wide Area Network Analysis: Can LoRa Scale?” *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.

[43] D. Bankov, E. Khorov, and A. Lyakhov, “Mathematical model of LoRaWAN channel access with capture effect,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, Montreal, QC, Canada, 2018, pp. 1–5.

[44] J. Haxhibeqiri, A. Karaagac, F. Van Den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa Indoor Coverage and Performance in an Industrial Environment: Case Study," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017.

[45] K. H. Phung, H. Tran, Q. Nguyen, T. T. Huong, and T. L. Nguyen, "Analysis and assessment of LoRaWAN," in *Proceedings - 2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications and Computing, SIGTELCOM 2018*, Ho Chi Minh City, Vietnam, 2018, pp. 241–246.

[46] G. Ferré, "Collision and packet loss analysis in a LoRaWAN network," in *25th European Signal Processing Conference, EUSIPCO 2017*, Kos, Greece, 2017, pp. 2586–2590.

[47] N. Varsier and J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications," in *IEEE International Conference on Communications*, Paris, France, 2017.

[48] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa Scalability: A Simulation Model Based on Interference Measurements," *Sensors*, vol. 17, no. 6, p. 1193, 2017.

[49] N. E. Rachkidy, A. Guitton, and M. Kaneko, "Decoding Superposed LoRa Signals," *CoRR*, vol. abs/1804.00503, 2018. [Online]. Available: http://arxiv.org/abs/1804.00503

[50] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT Traffic and Its Impact on LoRaWAN," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, Singapore, 2017, pp. 1–6.

[51] R. B. Sorensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of Latency and MAC-Layer Performance for Class A LoRaWAN," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, 2017.

[52] L. Li, J. Ren, and Q. Zhu, "On the Application of LoRa LPWAN Technology in Sailing Monitoring System," in *2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jackson, WY, USA, 2017, pp. 77–80.

[53] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, “LoRa and LoRaWAN Testbeds : a Review,” in *2017 IEEE AFRICON*, Cape Town, South Africa, 2017, pp. 1544–1549.

[54] L. Gregora, L. Vojtech, and M. Neruda, “Indoor Signal Propagation of LoRa Technology,” in *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, Prague, Czech Republic, 2016, pp. 1–4.

[55] A. Wixted, P. Kinnaird, A. Tait, A. Ahmadinia, and N. Strachan, “Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks,” in *2016 IEEE SENSORS*, October 2016, pp. 1–3.

[56] 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, October 2016.

[57] M. Aref and A. Sikora, “Free space range measurements with Semtech LoRa$^{\text{TM}}$ technology,” in *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2014*, Offenburg, Germany, September 2014, pp. 19–23.

[58] Lukas, W. A. Tanumihardja, and E. Gunawan, “On the application of IoT: Monitoring of troughs water level using WSN,” in *2015 IEEE Conference on Wireless Sensors, ICWiSE 2015*, Melaka, Malaysia, August 2016, pp. 58–62.

[59] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hänninen, and M. Pettissalo, “On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology,” in *2015 14th International Conference on ITS Telecommunications, ITST 2015*, Copenhagen, Denmark, December 2016, pp. 55–59.

[60] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, “Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring,” in *International Symposium on Medical Information and Communication Technology, ISMICT*, Worcester, MA, USA, March 2016, pp. 1–5.

[61] T. Wendt, F. Volk, and E. Mackensen, "A benchmark survey of Long Range (LoRa $^{\text{TM}}$) Spread-Spectrum-Communication at 2.45 GHz for safety applications," in *Wireless and Microwave Technology Conference (WAMICON), 2015 IEEE 16th Annual*, Cocoa Beach, FL, USA, April 2015, pp. 1–4.

[62] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, 2016.

[63] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, pp. 1–16, 2017.

[64] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. Skarmeta, "Performance evaluation of LoRa considering scenario conditions," *Sensors*, vol. 18, no. 3, 2018.

[65] P. J. Radcliffe, K. G. Chavez, P. Beckett, J. Spangaro, and C. Jakob, "Usability of LoRaWAN Technology in a Central Business District," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, Sydney, NSW, Australia, 2017, pp. 1–5.

[66] V. Hauser and T. Hegr, "Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, Prague, Czech Republic, 2017, pp. 85–90.

[67] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, "Adaptive data rate control in low power wide area networks for long range IoT services," *Journal of Computational Science*, vol. 22, pp. 171–178, 2017.

[68] F. Cuomo, J. C. C. Gamez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "Towards traffic-oriented spreading factor allocations in LoRaWAN systems," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2018*, Capri, Italy, 2018, pp. 1–8.

[69] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks

for dense IoT deployments," in *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, Taipei, Taiwan, 2018, pp. 1–9.

[70] B. Reynders, W. Meert, and S. Pollin, "Power and Spreading Factor Control in Low Power Wide Area Networks," in *Communications (ICC), 2017 IEEE International Conference on*, Paris, France, 2017, pp. 1–5.

[71] Multi-Tech Systems Inc. "Multitech's MultiConnect Conduit Breaks New Ground at Mobile World Congress". [Online]. Available: https://www.multitech.com/news-and-events/press-releases/MultiTech%E2%80%99s%20MultiConnect%20Conduit%20Breaks%20New%20Ground%20at%20Mobile%20World%20Congress_o23155. [Accessed 29 August 2018].

[72] R. Zhao, Jerry and Govindan, "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks," in *2010 International Conference on Multimedia Technology*, Los Angeles, California, USA, 2010, pp. 1–4.

[73] The Things Network. "LoRaWAN Adaptive Data Rate". [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/adr.html. [Accessed 04 October 2018].

[74] Multi-Tech Systems Inc. "Introduction to LoRa". [Online]. Available: http://www.multitech.net/developer/software/lora/introduction-to-lora/. [Accessed 04 October 2018].

[75] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, and J. Melia, "Understanding the limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[76] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, pp. 1–33, 2012.

[77] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov, "Measurements, Performance and Analysis of LoRa FABIAN, a real-world implementation of LPWAN," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2016, pp. 1–7.

APPENDIX A HARDWARE

A.1 PCB SCHEMATIC

Figure A.1 provides a block diagram of the developed node. Figure A.2 provides the node's PCB layout, and a schematic for the PCB is shown in Figure A.3.

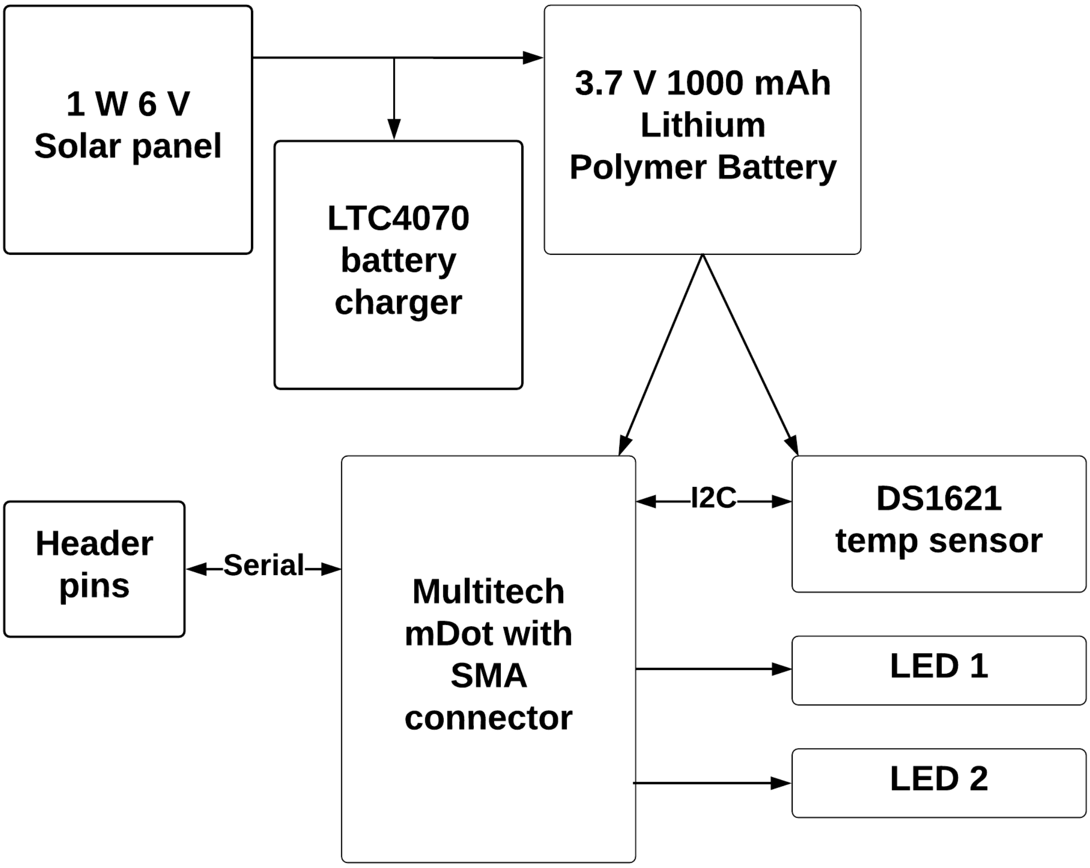


Figure A.1. Block diagram of developed node.

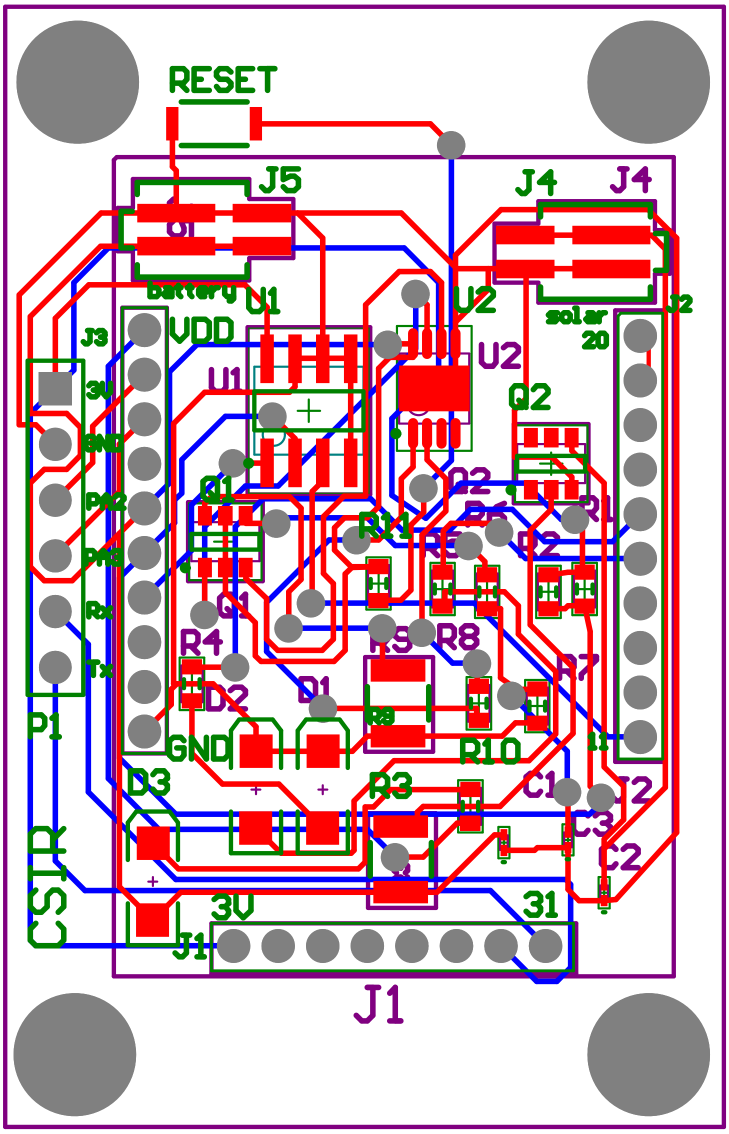


Figure A.2. PCB print.

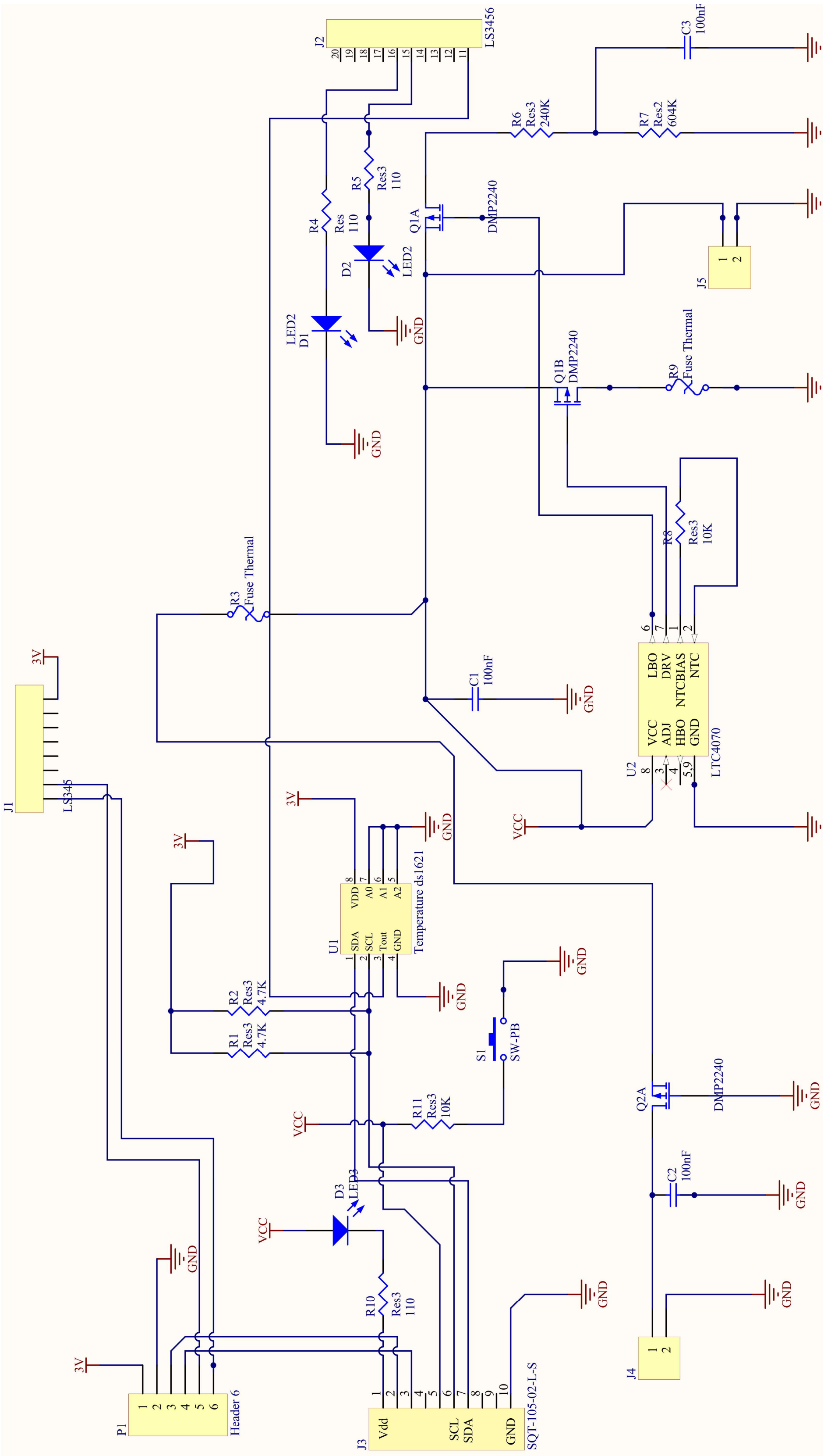


Figure A.3. Node's circuit diagram.

APPENDIX B SOFTWARE

B.1 FIRMWARE

The complete command set of the testbed is shown in Table B.1. As reprogramming the nodes would be time consuming, the command set was written to expose as much of Multitech's mDot library as potentially needed. The AT+TP command show only be sent to nodes which have a temperature sensor on their PCB.

The AT+GI/GK/GP commands were written to allow a node to receive the necessary information to connect to a new gateway. This would allow experiments with some nodes connected to gateway <x> and others connected to <y>. Once a node has been provided with the details for a new gateway it would store the information before restarting and attempting to connect to this gateway. This approach was abandoned as the values stored in flash memory became corrupted over time. Multitech warns that this can occur when the flash memory is accessed at a voltage level below 2.7 V. A new approach was followed in which three sets of gateway ID, key and network mode were hard coded. and AT+GW instructs nodes to switch between these.

A flowchart of how the command string is processed and how payloads are generated and sent is shown in Figure B.1.

Table B.1. Node command set.

Command	Purpose	Options	Valid values
AT+AC	Set/Query acknowledgements	? or =	0 to disable, 1-8=num of retries
AT+AD	Set/Disable adaptive data rate	? or =	0 or 1
AT+AN	Set/Query Antenna gain (dBi)	? or =	0-255, default 3
AT+CG	Query node config		
AT+DR	Set/Query data rate	? or =	DR0-DR5
AT+GI	Set gateway ID	=	ID (8 bytes) 0x99,0x45 should be sent as 9945
AT+GK	Set gateway KEY	=	KEY (16 bytes) 0x99,0x45 should be sent as 9945
AT+GP	Set if gateway's a public network	=	1 or 0
AT+GW	Set which of the 3 saved gateways to use	=	0-3
AT+LC	Set/Query link check	? or =	0-9 and 0-9, send together e.g. 34
AT+PL	Query max payload size	?	num of bytes, depends on SF
AT+PS	Set/Query payload size	? or =	size in bytes
AT+RS	Restart mDot		
AT+TP	Will query onboard I2C temp sensor	?	Sends back temp as an int
AT+TX	Set/Query transmit power (dBm)	? or =	2-20, default 14
AT+WT	Set/Query inter packet waiting time	? or =	in seconds (max 32 bits)

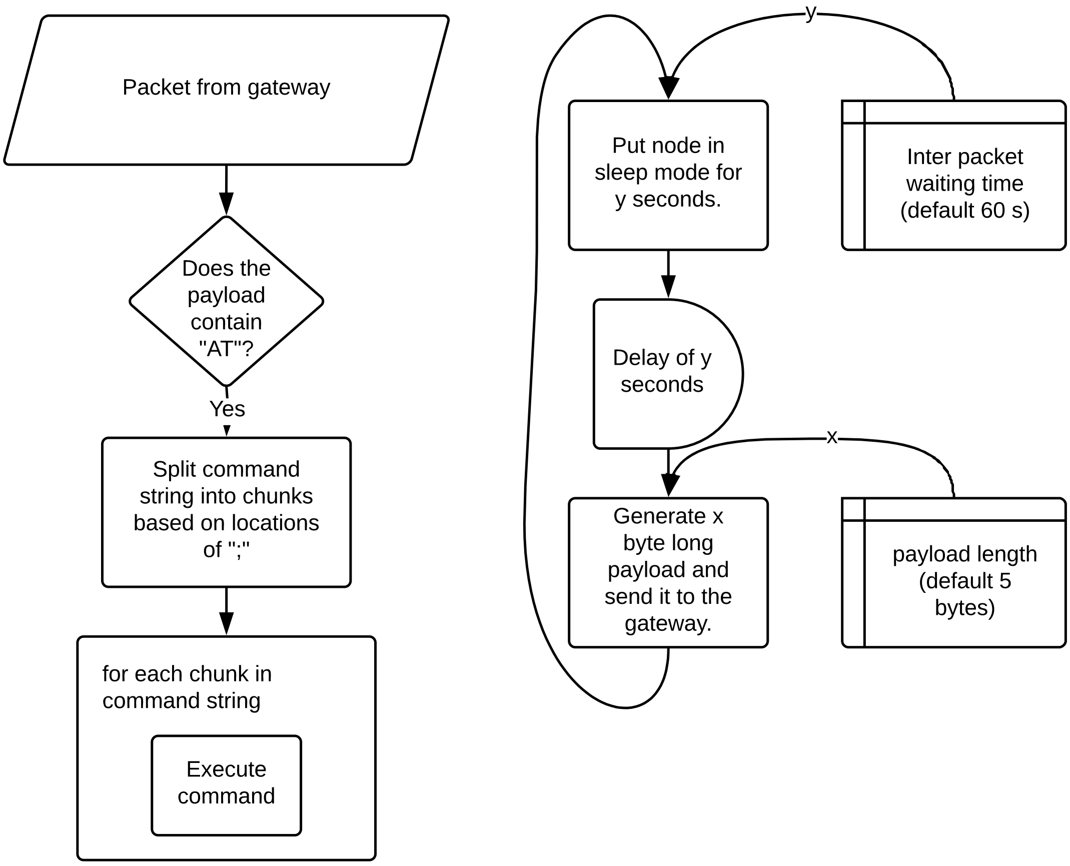


Figure B.1. Flow diagram of how the command string is processed in the firmware.