Chapter 1

Introduction

Decentralisation of wireless control and data sharing systems allows flexible deployment structures over large areas. Conversely, using a single centralised node, deployments are limited by that node's placement and its maximum communication range. This paper studies the application of LoRa, an emerging long range, low power, radio frequency (RF) technology, for the decentralised use case of sparse robot swarms.

Swarm robotics is the coordination of multi-robot systems such that a common goal can be achieved. Capabilities of a swarm should exceed that of any single robot in the swarm; be that attributed to increased coverage [1] or self-assembly methods [2]. Although, spreading robots across large areas opens up potential for many practical applications, including terrain mapping, and search and rescue, the sophistication of robots required in these real-world scenarios can make them prohibitively expensive, which can lead to limited robot density. These are referred to as sparse swarms. For the sake of perspective this paper assumes distances that are beyond the capabilities of typical short-range communications such as Wi-Fi.

Unlike a centralised control approach, swarms rely on robots sharing data directly so that all instances can build a combined interpretation of the environment. Although some data may need to be decimated to many or all robots in the network, the vast majority will only be of interest to physically local neighbours. Data of global interest may be for swarm management, e.g. voting decisions, or be generic, e.g. battery usage figures for specific terrain fingerprints. Whereas, data of local interest may consist of local area features, e.g. robot routes and found obstacles. In critical scenarios, for example when a robot failure is impending, large fast data dumps may

be required. Alternatively, data can be continuously aggregated and distributed in a BitTorrent-like fashion, though, this still requires a transfer mechanism [3].

For concentrated deployments, these scenarios are trivial to implement using high-data-rate technologies such as Wi-Fi. However, in a real-world scenario, when inter-robot distance is significant, and there are line of sight (LOS) obstructions (e.g. trees), an alternative physical medium is required. This leads to the choice of LoRa, detailed in Section 2.1. The system can be described as a mobile-ad-hoc-network (MANET), due to the ever changing topology caused by internal system changes (e.g. robot movement), and external system changes (e.g. weather).

Although LoRa is fundamentally ideal for long-range applications and operates in the low attenuation Sub-1GHz band, scenario specific conditions of ground-level transmissions and high-propagation environments are not ideal for any RF communications. Therefore this project initially covers real-world testing in free-space and forests to assess how sparse swarm deployment scenarios may affect LoRa's physical radio performance. From this, demodulation models are extracted and combined with LoRa collision models from literature to create a simulator for testing LoRa ad-hoc scenarios. This simulator is then used to analyse the effectiveness of three MAC protocols with the aim of prioritising data distribution to physically close neighbours. Two of which utilise common approaches (ALOHA and CSMA), and a third which attempts to exploit hardware parameters and regional RF regulations; this is defined as the LoRa Local Broadcast Protocol (LLBP).

Chapter 2

Background

2.1 LoRa

2.1.1 Overview

LoRa is a physical long-range, low-power, communication technology developed and patented by Semtech[1]. It is designed to operate inside either the Sub-1GHz or 2.4GHz unlicensed ISM bands worldwide. Consequently, under the proviso that local regulatory standards are obeyed (see Section 2.2), it can be used for wide area deployments without being tied to expensive licensed carriers. Fundamentally, even using its fastest configuration, it is a low-data rate modulation technique.

Long-range communication can be a challenge in the ISM bands as the heavy congestion can result in a high physical noise floor. That is the sum of all signals in the band from sources such as the atmosphere or radio devices, excluding that of the signal being monitored. LoRa functions using a unique spread spectrum modulation technique that can operate below the noise floor. Spread spectrum techniques allow signal-to-noise degradation in a single channel to be compensated for by spreading across other channels. Unlike other spread spectrum techniques that use a fixed chip sequence to carry out spreading, LoRa modulation uses a chirp signal that varies in frequency continuously. This is referred to as chirp spread spectrum (CSS) modulation and allows the complexity of the receiver design to be greatly reduced, resulting in a reduced and accessible hardware cost [4].

[1]Semtech, USA, https://www.semtech.com/

The link budget of a RF system is defined as the measure of all gains and losses incurred by a signal passing through the transmitter, the receiver and the propagation channel. The equation in its simplest form is:

$$RX\ Power\ (dB)\ = TX\ Power\ (dB)\ +\ Gains\ (dB)\ -\ Losses\ (dB) \quad (2.1)$$

A system is said to be link limited when the channel losses result in a receive power (and therefore SNR) lower than can be demodulated by the receiver. LoRa modulation boasts a high and adapative receiver sensitivity compared to frequency shift keying (FSK) and other modulation types, allowing it to make far more efficient use of its link budget [4].

2.1.2 Parameters

LoRa hardware has many parameters that can be configured to extend range or increase reliability at the expense of air time, data-rate or energy consumption. These are independent from any external hardware and include:

- **Bandwidth (BW)**: The range of the chirps around the CF. Increasing bandwidth increases data-rate but decreases receiver sensitivity [5].
- **Carrier Frequency (CF)**: The centre frequency of chirps. Current hardware targets some sub-range of 137MHz to 1020MHz at a resolution of 61Hz [6].
- **Coding Rate (CR)**: The amount of redundant information encoded in symbols for forward error correction (FEC); trade-offs can be seen in Table 2.1. FEC is most effective in the presence of burst interference [4].
- **Preamble Symbols (PS)**: The number of programmed preamble symbols sent for receiver synchronisation. Packet receive percentage (PRP) has been shown to increase with an increased PS up to a certain threshold [7]. Increasing length can also improve performance of CAD (see Section 2.1.5).
- **Spreading Factor (SF)**: The ratio of chip rate to bit rate, where chips per symbol is given as 2^{SF}. Receiver sensitivity increases in line with spreading factor and is therefore a factor in the link budget [6]. The trade-offs can be seen in Table 2.2.
- **Transmission Power (TP)**: The radio output power. Equation 2.1 highlights how an increased TP directly increases link budget with the obvious drawback of higher power usage.

TABLE 2.1: Effect of CRs on LoRa transmissions. Recovery performance is defined as the best case percentage of bits that can be lost for a successful receive. Compiled from data in [7].

Coding Rate	**Data Overhead**	**Recovery**
4/5	×1.25	20%
4/6	×1.50	33%
4/7	×1.75	43%
4/8	×2.00	50%

TABLE 2.2: Effect of SF on LoRa transmissions (BW=125KHz). Compiled from data in [4] and [7].

SF (LoRa Mode)	**SF (Chips / Symbol)**	**Bit Rate (bits / sec)**	**Demodulation Limit (SNR dBm)**
SF7	128	5469	-7.5
SF8	256	3125	-10.0
SF9	512	1758	-12.5
SF10	1024	977	-15.0
SF11	2048	537	-17.5
SF12	4096	293	-20.0

Selection of these parameters is often manual, however mechanisms such as LoRaWAN's adaptive data-rate (ADR) [8], or "probing algorithms", as proposed by [9], can be used to choose these parameters such that transmission energy is reduced whilst maintaining an adequate throughput and link budget. LoRaWAN abstracts SFs and BWs into a set of orthogonal data-rates (DRs) to simplify selection, where lower data-rates have higher range [10]. It has been suggested that ADR has low-scalability due to packet count requirements [11] and is slow to converge [12]. Likewise [9] requires a large number of initial transmissions, making both only suitable for static nodes in a gateway controlled network.

2.1.3 Airtime

To understand a configuration's performance, it is critical to know its airtime; this defines bitrate and affects channel contention methods. The total airtime of a transmission (Equation 2.3) is dependent on the number of symbols being transmitted

(Equation 2.2) and the time each symbol takes to send. Preamble and payload are calculated individually as it can be useful to consider them as separate components for purposes such as CAD. Equations are adapted from [6] to suit the notation and formats used by this paper.

$$S_{preamble} = \mathrm{PS} + 4.25 \tag{2.2a}$$

$$S_{payload} = 8 + \max\left(\mathrm{ceil}\left[\frac{8\mathrm{PL} - 4\mathrm{SF} + 16\mathrm{CRC} + 20\mathrm{EH} + 8}{4(\mathrm{SF} - 2\mathrm{LDR})}\right]\frac{1}{\mathrm{CR}},\ 0\right) \tag{2.2b}$$

$$S_{total} = S_{preamble} + S_{payload} \tag{2.2c}$$

The unmentioned factors in Equation 2.2 are: the number of bytes in the packet (PL), whether the CRC check is enabled (0 or 1), whether the explicit header (EH) is enabled (0 or 1) and whether low-data-rate optimisation (LDR) is enabled (0 or 1). LDR is used when $T_s > 16ms$ to aid stability over long transmits [6].

$$T_s = \frac{1}{S_R} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{2.3a}$$

$$T_{preamble} = S_{preamble} \cdot T_s \tag{2.3b}$$

$$T_{payload} = S_{payload} \cdot T_s \tag{2.3c}$$

$$T_a = T_{packet} = T_{preamble} + T_{payload} \tag{2.3d}$$

Equation 2.3 highlights that each SF increment will cause T_a to double for the same number of symbols. Likewise, doubling BW will half T_a. Figure 2.1 shows the structure of a LoRa transmission and therefore where each airtime component comes from.

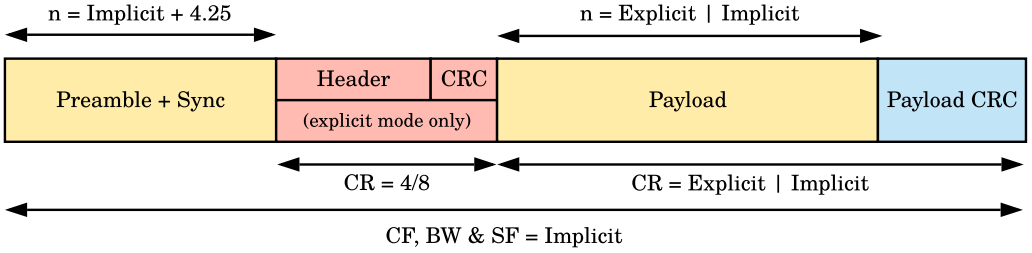


FIGURE 2.1: Structure of a standard LoRa transmission. All transmissions contain a preamble, sync words, and a payload. The header section (in red) is optional but if present, contains information such as the payload length, the payload CR, and whether the CRC is present. If the header is not present this information must be fixed implicitly by the receiver. Parameters that are not in the header are always implicit and must match between transmitter and receiver (i.e. PL, SF, BW and CF). Adapted from [6].

2.1.4 Receive Behaviour

LoRa radios, like all current consumer radios, are half-duplex, meaning that they are unable to receive for the duration of transmissions. Additionally, end device radios will usually be transceivers, which can only demodulate one incoming signal at a time [6]. In the presence of multiple signals, some transmissions may be missed, or collisions may occur, causing all transmissions to be missed. Collision scenarios specific to LoRa are identified in Table 2.3.

TABLE 2.3: Collation of LoRa collision scenarios as defined by [13] and [14]. [14]'s definition of the important preamble (IP) is used: the four fixed preamble symbols and proceeding two symbols of the programmed preamble. Situations use two transmission sources (A and B) and one receive source (C).

ID	**Time**	**Power**	C **Result**
A	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} \geq B_{\text{RPS}}$	Receives A
B	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} < B_{\text{RPS}}$	Receives A
C	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} \ll B_{\text{RPS}}$	CRC Fail A
D	B_{start} *inside* A_{IP}	$A_{\text{RPS}} \leq B_{\text{RPS}}$	Collision
E	$B_{\text{IP}} \approx A_{\text{IP}}$	$A_{\text{RPS}} \approx B_{\text{RPS}}$	Collision
F	$B_{\text{IP}} \approx A_{\text{IP}}$	$A_{\text{RPS}} \gg B_{\text{RPS}}$	Receives A

It should be noted that a different technology exists in gateways (a LoRa concentrator block), allowing demodulation of up to eight signals concurrently, provided they use unique spreading factors [15]. Although gateways are clearly more powerful than transceivers, their cost and power usage make them hard to deploy on scale. However, Pycom's[2] newly released Pygate gateway is a fraction of the cost of existing implementations and may be feasible for ad-hoc scenarios.

Most LoRa applications consist of many sensor nodes infrequently sending data on different spreading factors to a single gateway with very little downlink present; this means collisions and missed receives are rare. Unfortunately, in ad-hoc networks, these events are very likely and can be detrimental to a network's throughput; this is further explored in Section 2.3.2.

[2]Pycom, UK, https://pycom.io/

2.1.5 Channel Activity Detection (CAD)

Carrier-sensing is a helpful mechanism for radios to check whether a channel is busy or idle. Usually, this is achieved by checking the power present in the channel using the received signal strength indicator (RSSI). This is a very unreliable method for LoRa because the RSSI includes channel noise, and LoRa signals can operate below the noise floor. For this reason, LoRa radios offer a specialised CAD method, which searches the channel for a single LoRa packet preamble symbol. CAD is at least 97% reliable in the presence of preamble with false positives occuring just 0.1% of the time [13]. It has been shown that CAD can in fact detect non-preamble symbols when there is high signal strength, although this ability quickly becomes unreliable in a real world scenario [16].

2.1.6 Signal Orthogonality

The manner of LoRa's modulation allows multiple signals to co-exist in the same channel provided they have a different chirp rate, where $C_R = \mathrm{BW} \cdot S_R$, otherwise written as $C_R = \frac{\mathrm{BW}^2}{2^{\mathrm{SF}}}$. This clearly demonstrates that for a single bandwidth, all SFs must be orthogonal to one another. However, in the case that different BWs are used, different SFs may have the same chirp rate and could interfere; this is demonstrated in Figure 2.2.

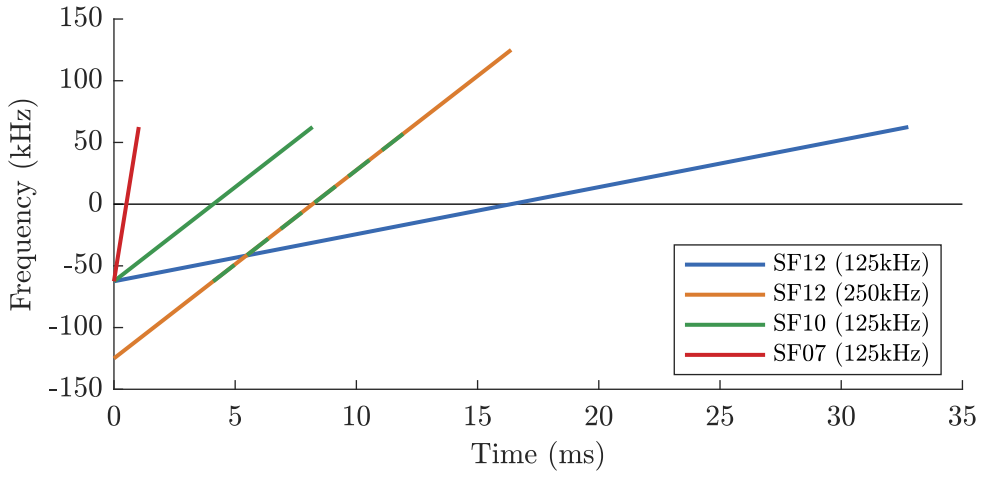


FIGURE 2.2: Demonstration of signal orthogonality for different SFs. The SF10 (125kHz) chirp is duplicated and shifted to overlap with the SF12 (250kHz) chirp to highlight that they have the same chirp rate and are therefore not orthogonal.

2.2 ISM Band Regulation

The industrial, scientific and medical (ISM) bands are portions of the radio spectrum, which can be used without a license, subject to local regulatory standards. These standards vary around the world but often define maximum power outputs, duty cycles, and bandwidths. Though limits can be problematic, they help reduce the chance of internal and external system interference. Some of the most stringent regulations are in Europe, where they are controlled firstly by the European Telecommunications Standards Institute (the ETSI[3]), and then by country specific authorities. The United States' regulations are managed by the Federal Communications Commission (the FCC[4]), with many other countries following their example.

Sub-1GHz LoRa hardware operates around the 433MHz and 900MHz bands, but as the former is very heavily regulated for ad-hoc scenarios in the US [17], only the 900MHz area is considered viable to the use case. It should be noted however that the specific frequencies used will still differ between region, as is demonstrated by LoRaWAN's use of the 863-870 MHz and 902-928 MHz bands for Europe and the US respectively [10]. A comparison of the regulations can be seen in Table 2.4. As ETSI limits vary heavily on a by band basis, they are broken down further in Table 2.5. An overview of each regulation type follows.

Table 2.4: Regional regulation comparison for 900MHz band radio [18], [19].

	FCC	**ETSI**
Band	902–928MHz	863–870MHz
EIRP	36dBm *30dBm transmit power*	0.25MHz @ 27dBm 6.40MHz @ $\leq$ 14dBm
Duty Cycle	None	0.1%–10%
Bandwidth	26 MHz	6.65MHz
Narrowband	400ms airtime per transmit	None
CSS	None	Varies

[3]ETSI, EU, https://www.etsi.org
[4]FCC, US, https://www.fcc.gov

TABLE 2.5: ETSI 868MHz sub-bands for short range devices (adapted from [19]). Only bands relevant for CSS are included. Band numbers correspond to CEPT-ERC-REC 70-03 definitions [20]

Band	**Frequency** (MHz)	**EIRP** (dBm)	**Duty Cycle**	**Max BW** (kHz)
h1.2	863.00–870.00	14	0.1% (or PSA)	7000
h1.3	865.00–868.00	14	1% (or PSA)	300
h1.4	868.00–868.60	14	1% (or PSA)	600
h1.5	868.70–869.20	14	0.1% (or PSA)	500
h1.6	869.40–869.65	27	10% (or PSA)	250
h1.7a	869.70–870.00	7	None	300
h1.7b	869.70–870.00	14	1% (or PSA)	300

Duty cycle limits greatly reduce the amount of airtime a radio is allowed. For example, in Band h1.3, there is a 1% duty cycle, which indicates a maximum of 36 seconds of airtime, over all the band's channels, over a rolling one hour period. Duty cycles are considered within a band so a multi-band implementation may obey multiple duty cycles separately. Alternatively, the polite spectrum access (PSA) policy can be used; this is a defined regulation for listen before talk (LBT) implementations PSA allows airtime of up to 100 seconds per 200kHz of spectrum, per hour, regardless of duty cycle [21]. It does however require a clear channel assessment to be carried out before every transmission, this would be independent of LoRa's CAD.

Regulations consider power as equivalent isotropically radiated power (EIRP). This value is directly related to radio transmission power (P_t) and can be calculated as $EIRP = P_t - L + G$ where L is cable loss and G is antenna gain. The latter occurring from transmit power being concentrated into a smaller area, all antenna will have some form of gain as isotropic antennas are only hypothetical. For example, a radio operating in Band h1.3, using a typical omnidirectional antenna that has 3dBi of gain, can only transmit at 11dBm if no cable loss occurs. Realistically, some cable loss will occur but this must be considered on an implementation by implementation basis. Directional antenna compound these issues due to their high-gains and the relatively low EIRP limits.

The ETSI regulations are the limiting factor in transmit power, duty cycle and overall available bandwidth. LoRa being a CSS signal means FCC narrowband limitations need not be a concern. Under this consideration, the ETSI regulations are used as the worst-fit scenario from here on in. When considering national regulation,

it is common that either all bands are implemented, or none at all [20]. Of the ETSI bands, h1.3, h1.4 and h1.6 are of most interest. h1.3 and h1.4 giving a balanced offering of bandwidth, EIRP and duty cycle. Whilst h1.6 offers a far greater duty cycle and EIRP but limited bandwidth; a further regulatory limit means only allows a single wide-band channel can operate within this band. The maximum number of possible LoRa channels in each of these bands can be seen in Table 2.6.

TABLE 2.6: Maximum channel count breakdown for relevant ETSI bands, calculated for LoRa's most common operating bandwidths. Calculated assuming channel spacing is 120% of the bandwidth to avoid inter-channel interference.

	Channel BW		
Band	**125kHz**	**250kHz**	**500kHz**
h1.3	19	8	N/A
h1.4	4	2	1
h1.6	1	1	0

2.3 Ad-Hoc Networks

2.3.1 Routing

An ad-hoc network is a type of wireless network that does not rely on any managed infrastructure, such as hard-wired routers. The network's nodes are responsible for determining their own routing paths and forwarding other nodes packets (i.e. acting as the routers). As explored by [22], a single network could make use of multiple transmission mediums to reach the destination node. A MANET is a special type of ad-hoc network where nodes are expected to move, resulting in frequent changes to the network topology [23]. If a network is sparse or operating at the limits of the transmission medium, and packet delivery is not time critical, the network can be treated as a delay-tolerant-network (DTN). A common approach to DTNs is to adopt store-carry-forward (SCF) behaviour; this is where intermediate nodes will keep hold of data until either a new path appears or signal strength improves [24].

Route management is the most researched challenge when it comes to ad-hoc networks [25] with implementations typically falling into the proactive or reactive categories - though more scenario specific variations do exist (e.g. geographic). Nodes using a proactive approach maintain a routing table for the whole network, to achieve

this they rely on periodic updates from other nodes with their routing tables; these methods have low transmission delay but high ongoing overhead and adapt slowly to network changes. Nodes using a reactive approach explore the network when necessary to find a path, often by flooding route request packets; these methods have high transmission delay, but no ongoing overhead and can adapt to network changes immediately. Routing is not considered further in this paper so this is the abstracted level to which algorithms are considered. Full descriptions of many examples, including AODV (reactive), OLSR (proactive) and LAR (geographic) can be found in [26].

2.3.2 MAC Protocols

An ad-hoc network contains many transmitters, therefore a medium access control (MAC) protocol is required to regulate access to the shared transmission medium. The selected method has a considerable effect on network efficiency and reliability in terms of power usage, collision occurrence, throughput, latency and fairness. Protocols can be classed as either contention-free or contention-based. The former use transmission schedules; these can waste resources if nodes do not require equal channel access, and struggle to adapt to changing topologies, but can be completely collision-free. The latter rely on nodes competing for access, these are flexible as they can adapt to different topologies with little overhead, however they are not collision free. For critical communications it must be possible to detect these collisions and recover from them. This can be very costly, requiring acknowledgements and retransmissions [27].

IEEE 802.11 (Wi-Fi) uses a combination of carrier-sense multiple access (CSMA) and multiple access with collision avoidance (MACAW). This is where a node first senses the medium for activity, before reserving the channel by transmitting request to send (RTS) and clear to send (CTS) control messages [28]. Theoretically this will alert other nodes so they do not transmit for this duration. Although the overhead introduced is not ideal, it is acceptable for high data-rate communications and large transmissions. The reservation phase is inefficient for LoRa due to its long airtimes, however, pure carrier sensing implementations using LoRa's CAD have been shown to be effective in the presence of receivable transmissions [16].

LoRaWAN is a MAC protocol designed to be the de facto choice for point-to-multipoint LoRa applications. It is largely certified worldwide, open-source, and

is both managed and promoted by the LoRa Alliance[5]. The expectation of a star topology means the full protocol is not suited to the ad-hoc scenario, however, individual features are of interest. In principle it is implemented as P-ALOHA, a simple unchecked protocol where transmission occurs whenever a transmitter has data available to send. Duty cycle limits play a large part in keeping collisions at a minimum, Figure 2.3 explains how these are enforced. The unchecked approach reduces theoretical channel usage to just 18% [29].

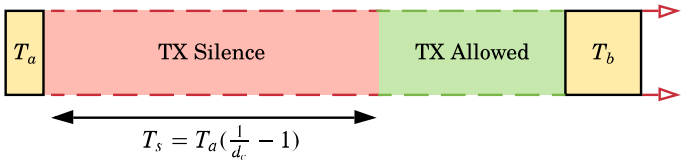


FIGURE 2.3: Demonstration of how LoRaWAN enforces duty cycle limits; that is after a transmission of airtime T_a, the transmitter must be silent for a minimum period of $T_s = T_a(\frac{1}{d_c} - 1)$ [30]. The figure is to scale for $d_c = 10\%$.

[5]Lora Alliance, https://lora-alliance.org/

Chapter 3

LoRa PHY Testing

It has been repeatedly shown that LoRa transmissions can be received at distances exceeding 10km in unobstructed environments (free-space) when antennas are highly elevated [31]. However, these ideal radio conditions are unrealistic for swarm robots operating close to the ground in high-propagation environments such as forests. Therefore the first experiment in this paper identifies LoRa's physical performance and scenario specific limitations.

3.1 Methodology

The purpose of this testing was to inform protocol decisions and assess capability of an off-the-shelf LoRa PHY. A focus was taken to get enough data across a small selection of important scenarios and parameters, such that they could be quantitively assessed to aid with qualitative protocol design decisions. Due to the expected sparsity of robots, near field scenarios (when transmitter and receiver are very close) were of little interest; this left only the far field to test. For 868MHz signals this meant the distance between radios (d_{tx}) had to exceed 34.54cm (1 wavelength).

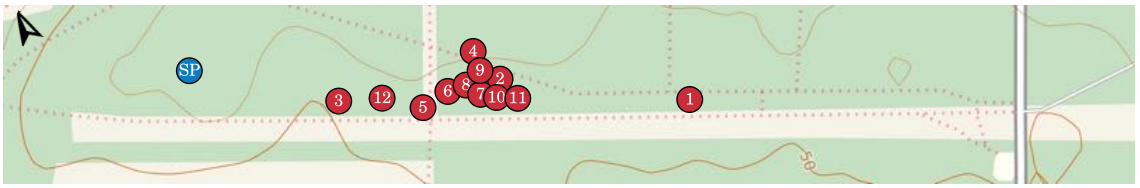
The two main transmission environments selected were free-space and in-forest; this was to give an understanding of both low-propagation and high-propagation scenarios. Data collection was mainly spread over two locations: L_A and L_B (split into L_{B1} and L_{B2}), identified in Figure 3.1 and 3.2 respectively. All locations were rural and were therefore theoretically free from strong sources of external interference. Radio placement at each location was decided by first placing the transmitting radio

(slave) at a fixed location, and then, using the furthest receivable point as the starting point for the receiving radio (master). From there the master was positioned closer towards the slave for each future test. In each scenario the main interest was ground level transmissions; however, to assess whether radio performance was actually compromised by the placement, comparative measurements were taken with an elevated antenna.

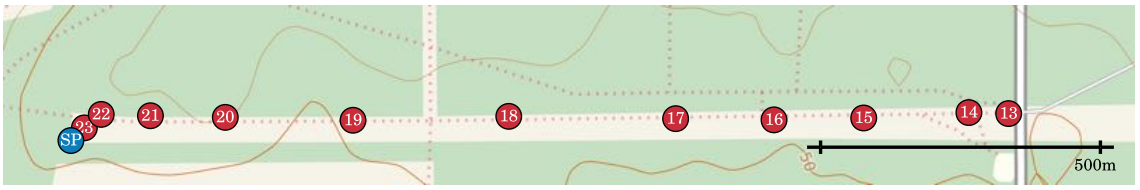
FIGURE 3.1: Test positions for L_A : The New Forest, Hampshire, UK.[1]
SP in open with LOS to other points a combination of free-space and light vegetation. Exact positions are in Appendix B.1. To the left of MP7 vegetation density increases, making MP7 the furthest position viable for free-space testing.

In terms of radio parameters, SF was the main focus due to it being an on option mostly unique to LoRa; all values were tested for this in all locations (SF = 7, 8, 9, 10, 11, 12). Variations using the lowest (4/5) and highest (4/8) CRs were collected to verify FEC performance in an environment with little or no burst interference. Additionally, as the maximum transmission unit is often defined by the protocol, and the target was to inform the protocol, the effects of varying packet length were taken (PL = 20, 128, 255 [PHY limit]). The rest of the parameters were fixed. The 868.1MHz CF was used with TP set to 14dBm so that the collected data would be relevant in regard to the ETSI regulations. The bandwidth was fixed to 125kHz so that radio sensitivity was only affected by the SF. The programmed PS were set to 8 to match LoRaWAN [10]. The number of packets (PC) transmitted for each configuration was set to 50; though not guaranteed, this gave reasonable expectation of a normal distribution, thus allowing typical statistical analysis to be performed. See Table A.1 for full test definitions.

To test the point-to-point transmissions, two identical platforms, which together could log the performance of sending and receiving LoRa transmissions, were required. The platforms had to be suitable for outdoor use, be able to test multiple radio configurations whilst on location and provide a mechanism to indicate to user when the maximum range had been reached. The hardware and corresponding software created for this purpose is detailed in Section 3.2.



(A) Test positions for in-forest testing (L_{B1}). SP in forest with LOS to other points continually obstructed by a combination of leaved and bare trees. Exact positions are in Appendix B.3. Large clump of MPs where radio reception was inconsistent.



(B) Test positions for free-space testing (L_{B2}). SP in open with LOS completely free-space. Exact positions are in Appendix B.2. No access to right of MP13.

FIGURE 3.2: Test locations for L_B : Stansted Forest, West Sussex, UK.[1]

(A) L_A : MP05 (1.0m)

(B) L_{B1} : SP (0.5m)

(C) L_{B2} : MP16 (0.0m)

FIGURE 3.3: Pictures of test environments and conditions. Although testing was split across multiple days at each location, the same dry conditions were present.

[1] Copyright © 2019 MapOSMatic/OCitySMap developers
Map Data © 2019 OpenStreetMap contributors (see http://osm.org/copyright)
British Style © MapQuest
Contour Overlay © OpenSnowMap.org

3.2 Testing Platform

3.2.1 Hardware

The basis of the designed test platform is HopeRF's[1] RFM95W - a packet radio containing a LoRa transceiver design licensed from Semtech; specifically, a broken out version from Adafruit's[2] is used. As a raw packet radio, unlike the popular Microchip RN2483, it provides direct access to the radio interface. An omni-directional 3dBi gain half-wavelength whip antenna is connected to the radio using a soldered uFL connector and a SMA to uFL connector. It is controlled by a Teensy[3] 3.6 micro-controller, which also handles all logging responsibilities. A simple breakout circuit is implemented on strip-board to connect the components in a condensed package. Each breakout board features: a JST-PH2 battery connector, a coin cell holder for the Teensy's real-time-clock (RTC), a power switch, a two-mode software switch, and three status LEDs. The schematic can be viewed in Figure C.1. This is packaged to fit in an IP67 rated container with an internal 1800mAh Li-Po battery. Switches and SMA antenna connectors are external; these are IP67 rated and sealant is added where appropriate. The Teensy is equipped with an SD card for storage but, due to cost considerations, a GPS module is not implemented. A full breakdown of materials is listed in Figure J.2. The created test platforms, seen in Figure 3.4, achieve the target of being a LoRa datalogger suitable for all-weather.

(A) (B)

FIGURE 3.4: External view of `S0` and `M0` platforms (left). Circuit view of `S0` (right); this is fixed into the assembly to avoid movement between tests.

[1]HopeRF Microelectronics Co. Ltd, China, https://www.hoperf.com/
[2]Adafruit, USA, https://www.adafruit.com/
[3]Teensy, https://www.pjrc.com/teensy/

3.2.2 Software

The system is designed such that one device (a slave), can be left unattended at a fixed location and controlled by a second device (a master); this is achieved using a command control system, as explained in Figure 3.5. Two command classes are defined for testing purposes; these are as follows:

- `HB_CMD` : Command to trigger simple heartbeat functionality. When a slave receives this command it sends a heartbeat response (`HB_RSP`) on the base configuration.
- `TD_CMD` : Command to trigger execution of a test definition (TD). A TD holds a LoRa configuration (values for CF, SF, TP, BW, CR, PL), a required PC, and packet length. Figure 3.6 explains the full control flow in detail.

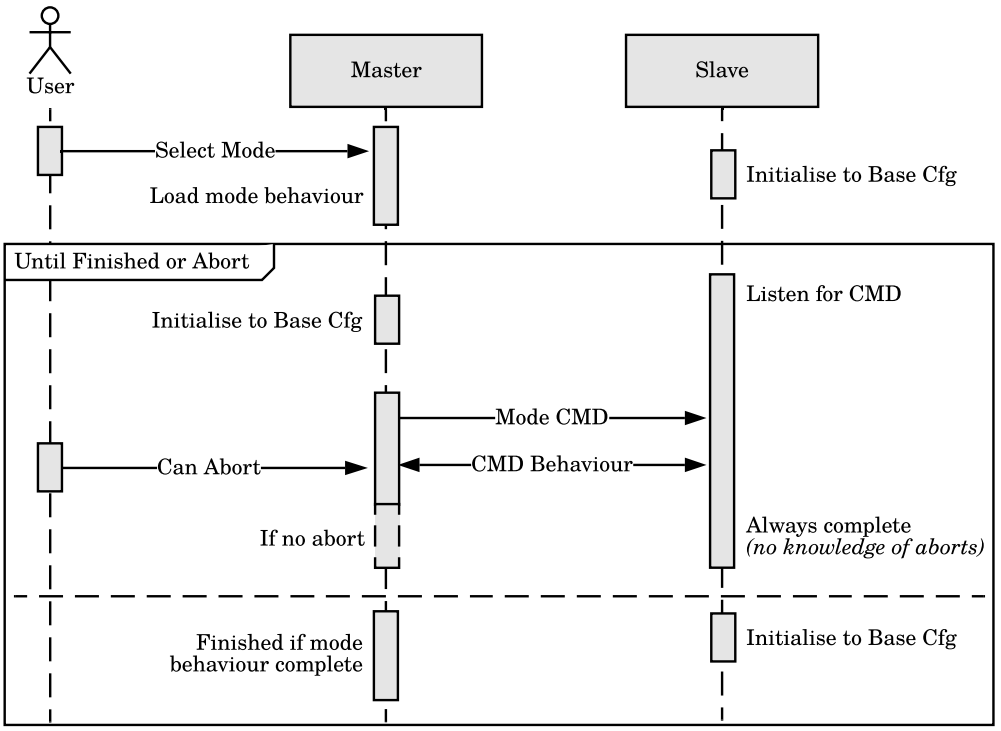


FIGURE 3.5: Diagram showing master-slave command control method. Initially, both devices default to the same hardcoded radio parameters allowing two-way communication. This base state is chosen such that the expected range exceeds or matches that of the longest test range. When a mode is selected on the master, it sends the corresponding command to the listening slave and the behaviour is carried out. At any point the user may stop the master, and unless the slave is interacting with the master, it likely has no knowledge of this and will finish its behaviour. After command behaviour has finished the base configuration is reloaded in case it has been changed. In the case a master's mode requires multiple commands, the process repeats.

Slaves always listen to handle incoming commands, whereas the master can be set into two modes (other than idle):

- **Heartbeat:** Sends periodic `HB_CMD` commands, alerts user accordingly for every received or missed `HB_RSP`.
- **Run TDs:** Loads all stored TDs and handles them sequentially using `TD_CMD`s.

Interfacing with the radio is handled by the RH_RF95 driver from the Radiohead[4] library. Operation of the software is detailed in Appendix D.

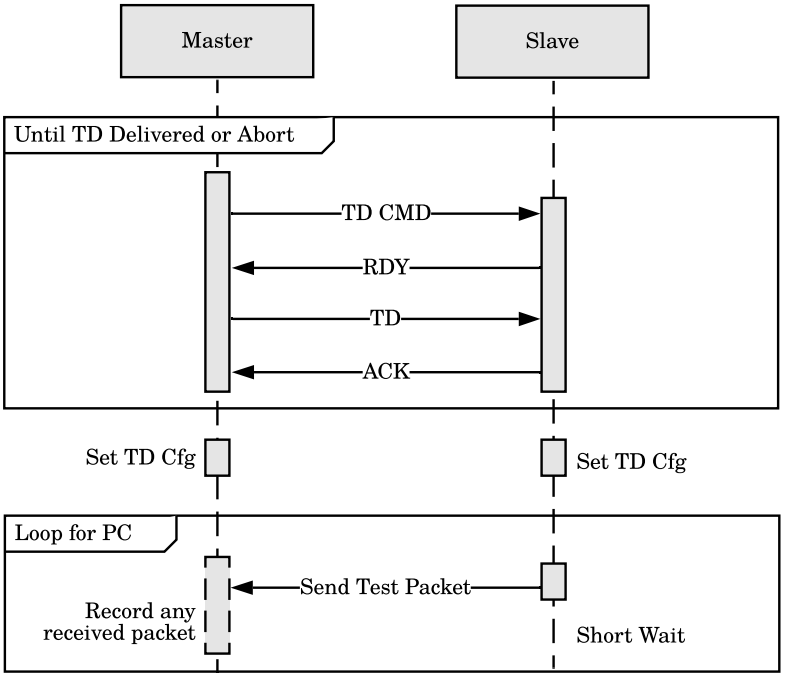


FIGURE 3.6: Diagram showing execution of a single TD. After a `TD_CMD` command is received, a short handshake takes place so that the master can share the TD to execute. After which both radios accordingly change their parameters and the slave sends the required PC. Test packets are of length defined by the TD and contain a sequence identifier with the rest of data filled by a fixed data pattern. Any received packets are recorded along with RSSI and SNR values. Failed receives that occur due to bad CRCs are also recorded. This means that only transmissions where the preamble is not received are not recorded.

[4]Radiohead, https://www.airspayce.com/mikem/arduino/RadioHead/

3.3 Results

In total 498 test cases were executed, totalling 24,900 packet transmissions. Of this total, 19,545 were successfully received (78.5%). The distribution of receive conditions for these individual points is indicated by Figure 3.7. Note that the raw RSSI values returned by the Radiohead library, and therefore the datalogger, are in fact packet strength for the SX1276 module; therefore a post-processing step has been applied to get separate packet strength and RSSI values valid for the RFM95W module. When discussing results, often received packets are considered alongside all other packets from the corresponding TD; this allows for metrics such as packet receive percentage (PRP) and average SNR to be used. Note that the logarithmic mean and standard deviation are used for decibel values.

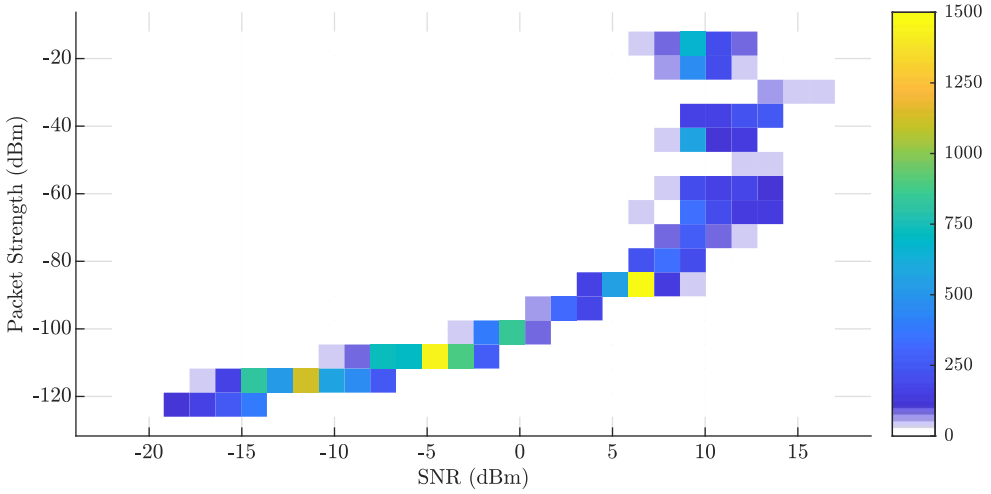


FIGURE 3.7: Density plot of received packet transmissions modelled as a bivariate histogram with colour indicating received packet count.
$Total\ \ Points = 19,545$

3.4 Discussion

Results are considered in two domains: the demodulation performance of the radio, and the environment effects. This is done because, even though actual receive performance varies with factors such as distance and attenuation, these can be abstracted into changes of the underlying SNR and RPS figures seen by the receiver. As there are no other transmission sources in this testing, theoretically, if these figures are within the required bounds for demodulation success, receives are successful.

3.4.1 Demodulation Performance

When $SNR \leq 0$ the RSSI value indicates the amount of noise seen by the receiver in the presence of no packet. When operating at 868MHz, the noise that the receiver should see is approximately the thermal noise floor ($-174 + 10log_{10}(BW)$), plus the receiver noise figure; LoRa implementations should have a noise figure of around 6dBm [4]. This indicates that for a 125kHz receive, the noise floor (n_f) should be -117dBm. The empirical noise floor calculated across all locations was -103dBm with a standard deviation of -109dBm; this is 14dBm (24×) higher than expected. As the variance is low, this result indicates that the RFM95W hardware is of much poorer quality than expected, with a noise figure of 20dBm.

Whether the radio receives a transmission is dictated by whether the received power exceeds the receiver sensitivity (R_S). For LoRa modules, $R_S = n_f - SF_{lim}$, where SF_{lim} is the minimum SNR required for the current SF. Theoretical limits are $-7.5, -10, -12.5, -15, -17.5, -20$ for $SF = 7, 8, 9, 10, 11, 12$ respectively [6]. This performance is explored in Figure 3.8 and 3.9. In short, performance is close to theoretical for $SF = 7, 8, 9, 10$ but $SF = 11$ & 12 perform similarly to $SF = 10$, just with higher reliably. For all configurations, receive success is highly variant when approaching the empirical sensitivity.

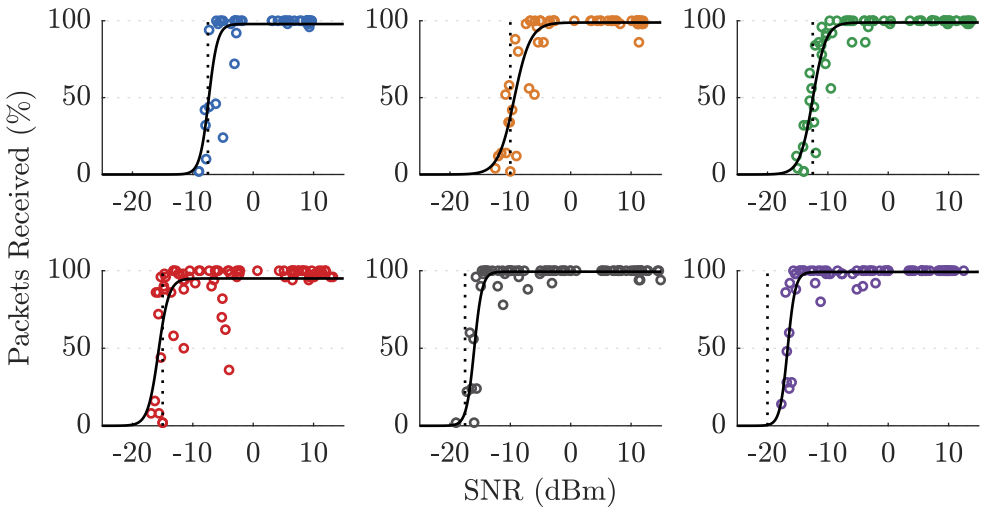


FIGURE 3.8: TD mean SNR values plotted against their PRPs, separated by SFs (Order = [[7, 8, 9], [10, 11, 12]]). For each SF plot: the theoretical demodulation limit is indicated by the dotted line and the solid line corresponds to the best-fit sigmoid function; these are repeated in Figure 3.9. Although the best-fit sigmoids give a good representation of the general data pattern, and provide empirical demodulation cut-offs, they do not capture the high-variance receive behaviour when approaching the cut-off. This is reflected by the fact that only 62%, 60%, 66%, 39%, 77% and 82% of the respective training points fall inside the corresponding 95% confidence interval.

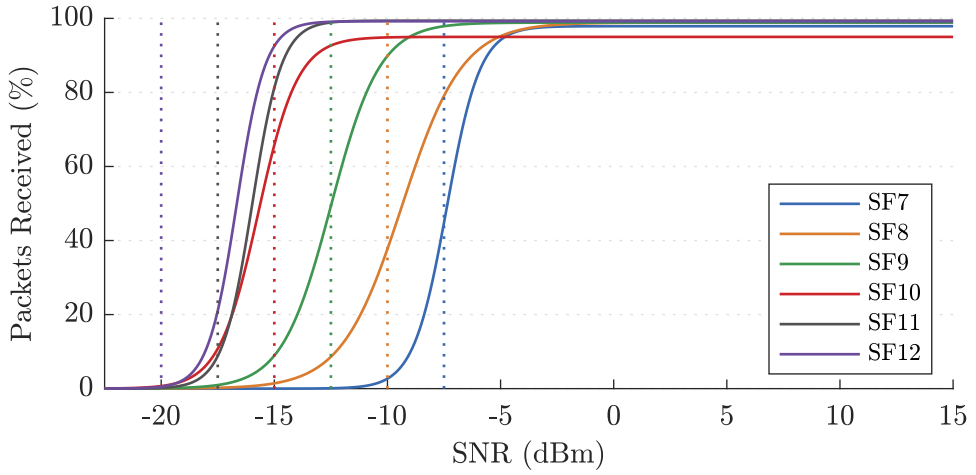


FIGURE 3.9: Plot of sigmoid best-fits generated in Figure 3.8. The plot clearly demonstrates the positive effect increasing SF has on demodulation performance of the receiver. For $SF = 7, 8, 9, 10$ demodulation success starts dropping approximately 2.5dBm before the theoretical limit (D_L), with a 50% PRP at D_L. This holds less so for SF=11, for which drop-off starts around $D_L - 5dBm$, until D_L where there is only a 10% receive success. For SF=12 drop-off starts around $D_L - 7.5dBm$, until D_L where there is a 0% receive success. Given the stable RSSI when $SNR < 0$, and that expected performance holds until a certain SNR, there is an indication that the sensitivity of the receiver is not as high as stated, possibly due to a cheap hardware implementation.

Theoretically, higher CRs will result in more data being recovered from a transmissions allowing for greater receive success. Comparative tests are plotted in Figure 3.10. As no strong visual conclusions can be made, a null hypothesis is proposed; H_0 : *The mean PRP does not increase between receives using CR 4/5 and CR 4/8* (otherwise written as $4/5_{PRP} \geq 4/8_{PRP}$). The respective means are 71.8% and 72.5%. Using a left-tailed Wilcoxon signed rank test for non-normal distributions gives $p = 41.2\%$. With a 5% significance level, H_0 cannot be rejected, indicating that CR has no effect on PRP. Given that the SNRs are not significantly different *(hypothesis testing omitted)* this indicates that receive drop-off and high variance when approaching sensitivity limits is the limiting factor for demodulation. The lack of CR effect is unsurprising given that FEC's main performance should be seen in the presence of burst interference.

When the amount of data increases in a packet, its airtime will increase for the same configuration; this can lead to more channel noise being introduced (lower SNR) and receiver clock drift (lower demodulation performance). The effect this has on comparative tests is plotted in Figure 3.11. The mean PRPs of $PL = 20, 128, 255$ are 81.7%, 78.8% and 76.8% respectively. Three null hypotheses are proposed: $H_0^1 : 20_{PRP} \leq 128_{PRP}$, $H_0^2 : 20_{PRP} \leq 255_{PRP}$ and $H_0^3 : 128_{PRP} \leq 255_{PRP}$. Using

right-tailed Wilcoxon signed rank tests with 5% significance, H_0^1 ($p = 1.1\%$) and H_0^2 ($p = 0.0\%$) are rejected but H_0^3 ($p = 13.6\%$) is not. Therefore alternative hypotheses can be accepted $H_A^1 = 20_{PRP} \geq 128_{PRP}$ and $H_A^2 = 20_{PRP} \geq 255_{PRP}$. Given that the SNRs are not significantly different *(hypothesis testing omitted)*, and that H_A^3 is narrowly rejected, a loose relationship between increased PL and lower demodulation performance of the receiver is assumed.

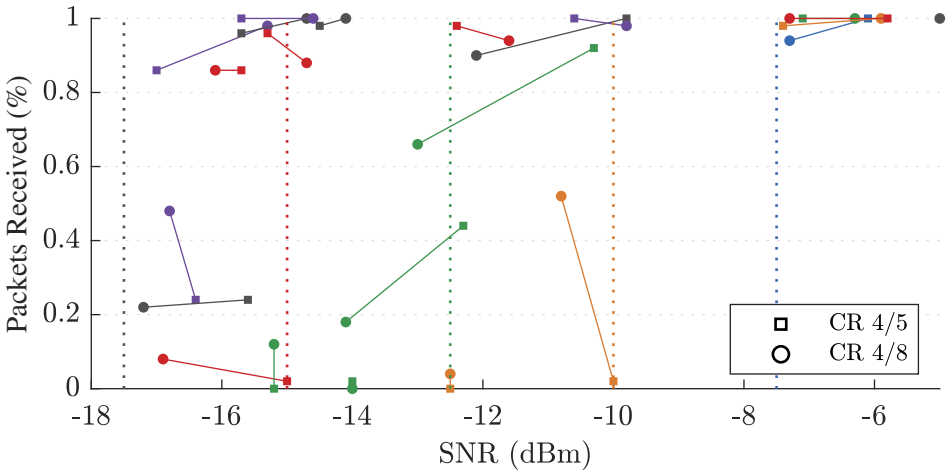


FIGURE 3.10: Plot of SNR and PRP for varying CRs. Only configurations where all other factors are identical are included (e.g. height, location, LoRa configuration). A line joins each set of points with a matching configuration. SF colouring from previous figures is applied to highlight when SNR limits start to reduce receive probability. When $SNR > -5$, the PRP is nearly always 100% and is therefore excluded.

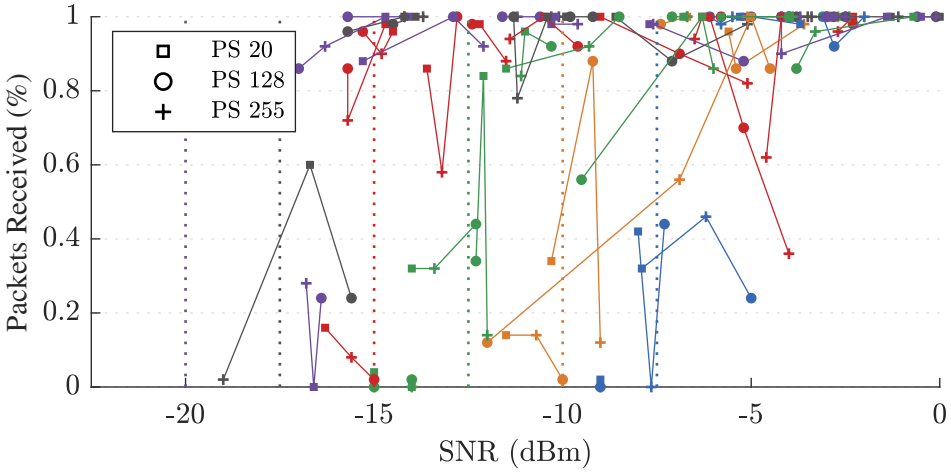


FIGURE 3.11: Plot of SNR and PRP for varying PLs. Only configurations where all other factors are identical are included. A line joins each set of points with a matching configuration. SF colouring from previous figures is applied to highlight when SNR limits start to reduce receive probability. When $SNR > 0$, the PRP is nearly always 100% and is therefore excluded.

3.4.2 Environment Effects

Free-space is the least attenuating environment possible and as such a transmission in free-space should represent the minimum path loss over a given distance; directly leading to the maximum transmission distance. The minimum free space path loss (FSPL), is calculated as $20\log_{10}(d) + 20\log_{10}(f) - 27.55$ where d is distance in meters and f is frequency in MHz [32]. A more reasonable estimate must take into account effects such as ground reflection. The plain earth (PE) model considers this and is calculated as $40\log_{10}(d) - 20\log_{10}(h_r) - 20\log_{10}(h_t)$ [33]. These are all plotted on Figure 3.12. Although transmissions occur at $0.0m$, receiver height (h_r) and transmitter height (h_t) are measured as the top of the antenna ($h_r = h_t = 0.17m$). As neither model fits the test data, with FSPL underestimating and PE overestimating path loss, an empirical log-model is calculated as $91\log_{10}(d+362) - 167$ (E-FSPL). This model fits the curve well but does not necessarily capture the variance caused by fading, as is reflected by only 83% of data points falling in the 90% confidence bound.

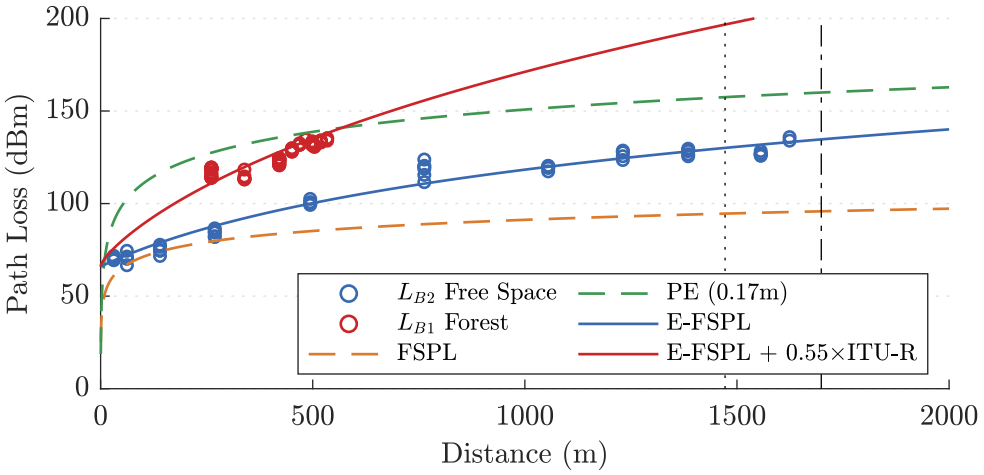


FIGURE 3.12: Plot of path loss for ground level transmissions through free-space (L_{B2}) and forest environments (L_{B1}). The LOS horizon and radio horizon are identified as the black dotted and dashed lines respectively.

Forests have high attenuation and should significantly increase path loss and reduce transmission distance. Due to the differences and complexity of vegetation, there is no de-facto propagation model, however, path loss should approximately match that of a vegetation model added to the environment's free space model [33]. Many are explained in [34], each of which can be made more flexible by applying an empirical multiplier, giving $L_{Total} = L_{FS} + \beta \times L_{Veg}$ [35]. The in-forest model demonstrated is the free-space-fit model with the ITU-R vegetation model, where $\beta = 0.55$. The model fits the in-forest test data; giving confidence in both the free-space fit model

and that in-forest behaviour follows that of generalised RF transmissions. Inter-transmission variance for a single test can be attributed to fading, however, it should be noted that test execution was inconsistent when approaching the transmission limit (480m+). This is likely a result of the bearing change between transmitter and receiver causing significant changes to the LOS obstacles. To model this, either individual objects could be modelled or a varying empirical multiplier could be used. Although both the free-space and in-forest fit models serve the purpose of describing the test data, a full assessment of their generalisation would require significantly more data.

Transmissions at 868MHz are classed as ultra-high-frequency and usually have a maximum distance somewhere between the visual-horizon and radio-horizon [32]. They are also susceptible to ground plane effects which can increase path loss. Varying radio heights for comparable measurements are plotted in Figure 3.13. The decrease in path loss is clearly seen for both $0.0m \rightarrow 1.0m$ and $0.5m \rightarrow 1.0m$. The effect is less clear for $0.0m \rightarrow 0.5m$. In free-space with $d = 10m$ on grass: path-loss for $0.0m$, $0.5m$ and $1.0m$ is 67dBm, 64dBm and 44dBm respectively. The increase between $0.0m$ and $0.5m$ is mathematically significant ($p = 0.0$) but of an insignificant degree compared to $0.5m \rightarrow 1.0m$. These results are in-line with the principle that the ground effect is insignificant once antenna height is more than a few wavelengths [32]. It is probable that transmissions are limited by the horizon in free-space, given that the furthest receivable transmissions occur between the horizons for both $0.0m$ and $0.5m$ and there is sudden increase in path loss between the horizons for $0.0m$.

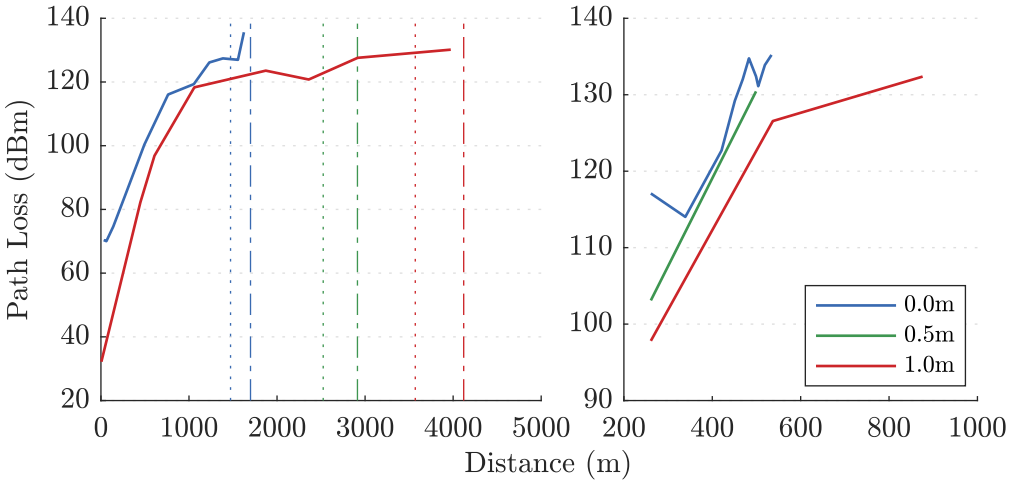


FIGURE 3.13: Plot of path loss for varying heights in free-space (left) and in-forest (right). Path loss is the mean of all tests at location. The dotted lines identify the LOS horizons. The dashed lines identify the radio horizons. Directly comparable data was not recorded for free-space 0.5m over multiple distances.

Chapter 4

LoRa Ad-Hoc Simulator

With an understanding of LoRa's performance - the next stage is to understand how LoRa performs in an ad-hoc network. This section proposes a specialised Ad-Hoc LoRa simulator, using models directly from PHY testing, as well as literature. It supports both scripted access for repeatable statistical testing and provides a GUI overlay for visually identifying system behaviour and performance.

Full source code is available in the project design archive.

4.1 Model

4.1.1 Overview

The model is considered as three main components, the environment, the radios, and time. The environment is responsible for understanding radio placement and the propagation channel between radios. Radios create new transmissions and poll the environment for existing transmissions, which they then attempt to receive. The simulator supports both the theoretical LoRa radio (modelled from the datasheet), and the empirically defined RFM95W. Time progression of the system is modelled using the activity-oriented paradigm; this is where time is considered as small sequential slices at a selectable granularity (e.g. 1ms, 5ms, 10ms). For every simulation tick, the time-slice increments and any events that have occurred or are occurring within that time slice are handled. Unlike an event-driven approach, every time-slice is simulated, allowing for continuous receive behaviour close to that of a real-world scenario; the drawback being that it is processor intensive.

4.1.2 Radio Model

4.1.2.1 Transmitting

When a packet is transmitted, it is modelled in terms of start time, airtime and emanating location. Each time-slice the environment will poll each radio to collate all ongoing transmissions in the environment; this is the list which radios attempt to receive from and interference models are calculated for. If a radio's location changes between time-slices, the emanating location will be re-evaluated each time, however, the Doppler effect is not modelled because it has little effect for on LoRa at expected movement speeds [36]. Transmission airtimes are calculated using the equations detailed in Section 2.1.3.

4.1.2.2 Receiving

Every time-slice, a radio, provided it is not transmitting, will fill its receive buffer with a partial receive of the strongest receivable transmission (highest SNR) in the environment. For LoRa transmissions, receivable implies the same SF, BW and CF values are used. If multiple receivable signals are present, and the radio is not synchronised with a signal, the strongest will be captured. A small advantage is provided to favour receiving synchronised transmissions to support Class B and C collision scenarios. Class E collisions are handled by randomly adjusting receive SNRs slightly so that no preference is made between two signals with similar strength. Simulator collision results, as verified by the `TestCollision` class, are seen in Table 4.1.

TABLE 4.1: Results for collision scenarios defined in Table 2.3, when modelled in the simulator. All scenarios are modelled correctly for all SFs.

ID	Time	Power	Status	Metadata Status
A	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} \geq B_{\text{RPS}}$	Receive A	`SUCCESS`
B	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} < B_{\text{RPS}}$	Receive A	`SUCCESS`
C	$B_{start} > A_{\text{IP}}$	$A_{\text{RPS}} \ll B_{\text{RPS}}$	CRC Fail A	`PAYLOAD_COLLISION`
D	B_{start} *inside* A_{IP}	$A_{\text{RPS}} \leq B_{\text{RPS}}$	Collision	`PREAMBLE_COLLISION`
E	$B_{\text{IP}} \approx A_{\text{IP}}$	$A_{\text{RPS}} \approx B_{\text{RPS}}$	Collision	`PAYLOAD_COLLISION`
F	$B_{\text{IP}} \approx A_{\text{IP}}$	$A_{\text{RPS}} \gg B_{\text{RPS}}$	Receive A	`SUCCESS`

Synchronisation is achieved when the important preamble of a signal (last 6.25 symbols) is in the receive buffer and demodulated successfully. Attempted demodulation occurs at the end of each time-slice and must pass at three stages to successfully receive a signal. If any stage fails the latter stages will not occur, but the signal will be unreceivable and may still interfere. The stages are:

S1 When a transmission's first important preamble symbol is seen.
Start of synchronisation.

S2 At the end of the transmission's preamble. At this point the radio is synchronised with a signal and is aware that it is receiving [6].

S3 At the end of the transmission. Check for successful receive.

S1 purely checks if the signal is strong enough using the radio's demodulation curve. Failure will result in `NO_PREAMBLE`, indicating a weak preamble. **S2** and **S3** also verify that enough of the transmission is present. This is required because it is possible that another powerful transmission has started filling the receive buffer - at this stage other signals are considered interference. For **S2**, 80% of the signal must be preamble from the correct signal. For **S3** the amount of interference must not exceed the CR's capability (Table 2.1); this resembles burst-interference disrupting a portion of the signal. Failure due to interference will result in a `PREAMBLE_COLLISION` or a `PAYLOAD_COLLISION` for **S2** and **S3** respectively. Likewise a demodulation failure will result in either `NO_PREAMBLE` or `WEAK_PAYLOAD`. These statuses are simulation metadata, the radio does not necessarily know the failure details or that it even occurred - a mapping of metadata to radio knowledge is seen in Table 4.2.

Table 4.2: Mappings of simulator receive statuses to those the radio would see.

Metadata Status	**Radio Status**	**Radio Aware**
`NO_PREAMBLE`	N/A	No
`PREAMBLE_COLLISION`	Collision	Yes
`WEAK_PAYLOAD`	CRC Fail	Yes
`PAYLOAD_COLLISION`	CRC Fail	Yes
`SUCCESS`	Receive	Yes

The mentioned demodulation curve varies depending on whether the theoretical model or the empirical RFM95W is used; only the latter is considered here. The transmission strength is calculated as the log-mean SNR of all its partial receives.

The corresponding $p = \text{PRP}_{\text{SNR}}^{\text{SF}}$ is found from the the best-fit sigmoid. If the SNR is close to D_L, p is randomly adjusted to emulate high-variance behaviour. Demodulation success is then determined as a random chance with success p. The demodulation curves for receives at various SNRs are plotted in Figure 4.1 (output generated by `FreeSpacePlot` class).

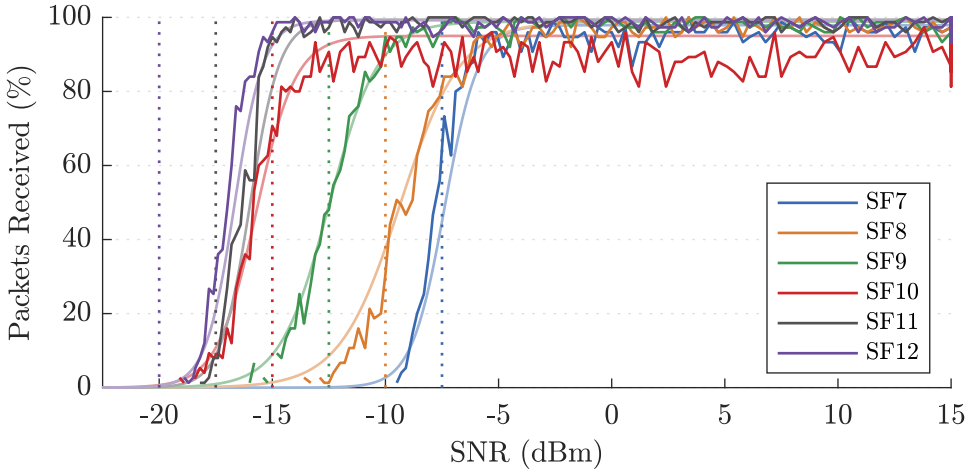


FIGURE 4.1: Simulator demodulation performance, assessed as PRP for increasing SNR. Each point is the result of 75 transmitted packets at the respective SNR. The faded lines indicate the empirical fit curves from Section 3.4.1.

4.1.2.3 CAD

LoRa's CAD process requires the ability to poll the receive buffer for a preamble symbol. When the process is invoked, the initial contents of the receive buffer are emptied as previous receive data cannot be used. A symbol's length of receive buffer is then captured (Equation 2.3a) and then the CAD search result is returned after the average processing time ($0.85 \cdot S_T$) [6]. Activity is detected if at least 80% of the receives in the buffer are preamble symbols and a standard demodulation curve check passes. When the process is complete the full receive buffer is dumped as it cannot be used for receives.

4.1.3 RPS (R_P) & SNR Model

The actual receivable signal power of a transmission is derived from the link budget equation, Equation 4.1 explicitly repeats this for the factors modelled.

$$R_P = TX_{power} + TX_{gain} + RX_{gain} - P_{loss}^{TX \to RX} \tag{4.1}$$

Gains at the receiver and transmitter end are broken down into fixed antenna gains and cable losses. The path loss (P_{loss}) is broken down into free-space loss and object loss. A global free-space model is used for the environment, selectable at creation. Although adding any new model is straightforward, built-in options include those defined in Section 3.4.2: E-FSPL, FSPL, PE.

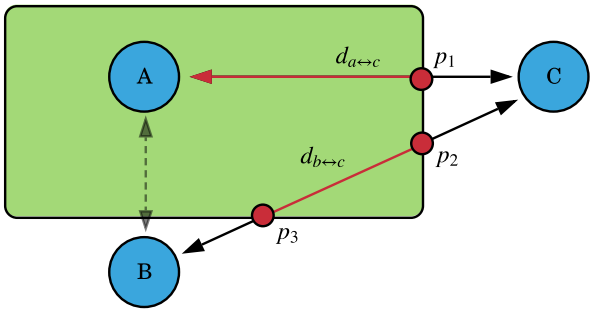


FIGURE 4.2: Scenarios for LOS path loss whilst passing through objects.

Objects are entities in the environment, which will possess their own propagation model; built-in models include: COST235, Weissberger and the ITU-R. Forests are a type of built-in object; these can take any 2D shape and use the ITU-R model with a β multiplier. All this combined allows the path loss curves from Section 3.4.2 to be recreated. Of course in a real environment, it is likely that not all radio transmissions will pass through the same obstacles. There is no perfect way to handle this, the closest would be the one woodland terminal model [37] but this sets severe limits; one radio must be in the object and the other outside with no other obstacles in the path. Therefore a more approximate approach is taken. In short the total P_{loss} is sum of the free-space loss and the segment of the propagation curve from the obstacle. Using Figure 4.2 as a scenario reference, the following are examples of extra P_{loss} due to obstacles. Equation 4.2 calculates for $A \leftrightarrow C$ where one radio is in the obstacle.

$$P^{A\to C}_{loss\ (\text{Object})} = P^{A\to p_1}_{loss} \quad \leftrightarrow \quad P^{C\to A}_{loss\ (\text{Object})} = P^{C\to A}_{loss} - P^{C\to p_1}_{loss} \tag{4.2}$$

Equation 4.3 calculates for $B \leftrightarrow C$ where the transmission completely passes through the object.

$$P^{B\to C}_{loss\ (\text{Object})} = P^{B\to p_2}_{loss} - P^{B\to p_3}_{loss} \quad \leftrightarrow \quad P^{C\to B}_{loss\ (\text{Object})} = P^{C\to p_3}_{loss} - P^{C\to p_2}_{loss} \tag{4.3}$$

As the method has large swings in path loss depending on whether obstacles are close to the receiver or transmitter, $\max(P^{TX\to RX}_{loss}, P^{RX\to TX}_{loss})$ is taken; this means

path loss will be the same regardless of the transmission direction – interference differences may still mean that SNR varies. The model scales to any number of objects in the path. Note that fast-fading is abstracted into the demodulation curve of the receiver and is not modelled separately.

The channel power as seen by a receiver (RSSI) at a location, is modelled as the sum of: all 'interfering' signal powers, the thermal noise floor (T_N), and the receiver noise figure (N_f). When attempting to receive a signal, its observed R_P can removed from the RSSI calculation to provide the channel noise (N). The SNR can then be calculated as $R_P - N$. Interfering sources, narrowband or otherwise, are considered as those where the bandwidth used overlaps with the receive channel. As per LoRa's orthogonality defined in Section 2.1.6, a LoRa transmission is only considered to interfere if the chirp rate is the same as that of the receive configuration.

4.1.4 Protocol Testing Infrastructure

Protocols are implemented the layer above the low-level radio behaviour using a listener infrastructure. When the time-slice increments, first the raw radio behaviour will occur, and then a 'tick' process will be called on any listeners. These listeners can maintain their own state between ticks allowing them to schedule transmissions and handle received packets, ultimately allowing emulation of a full protocol stack. More implementation specific behaviour like radio movement can be emulated here or on a higher level listener. This is how all the protocols defined in Section 5 are implemented. The base listener class, `ProtocolTickListener`, provides methods for directly handling events that have occurred the previous tick (synchronisations, receives, CAD results), tracks received data, and handles protocol performance tracking (for testing). Statistics can be dumped with a by-radio perspective – the number of packets received by each radio and the reason for any failures. Or with a by-transmission perspective – the number of receivers of each transmission and the reason for any failures. Outputs can be filtered to disregard failures of transmissions that are not 'wanted' by a receiver. The wanted criteria is detailed in Section **??**.

The execution of an environment (and the radios within it) is handled by an `EnvironmentRunner`; this can step time by the configured granularity a number of times or run continuously. Execution occurs on a separate thread and employs listeners to allow external interaction either through a test harness or GUI. Events can be created and scheduled for execution at fixed times to allow for scripted test

behaviour; there is built-in support for moving radios, sending transmissions and verifying the expected radio state; although, with full control over the system, any custom behaviour can be implemented.

4.2 Interface (GUI)

The Java Swing interface acts largely as a wrapper for managing an `Environment Runner` and displaying the environment being executed. The full interface can be seen in Figure 4.3 with further examples in Appendix E. User controls are provided for managing environment time, choosing presets and protocols, and accessing statistics. The graphical view, which has full support for panning and zooming, displays the environment's: objects, radios and transmissions. Objects are drawn using their obstruction shape (and specified attributes). Radios are blue circles when transmitting and grey otherwise. Transmission routes are displayed for the last received data in a radio's buffer as a line from transmitter to receiver with SNR indicated; preamble is identfied by a dotted line, payload as a solid line. The line will be grey if the receiver failed to synchronise with the preamble or red if the transmission was not the one the receiver is synchronised with. Otherwise, blue is used to indicate a data packet and orange as protocol overhead.

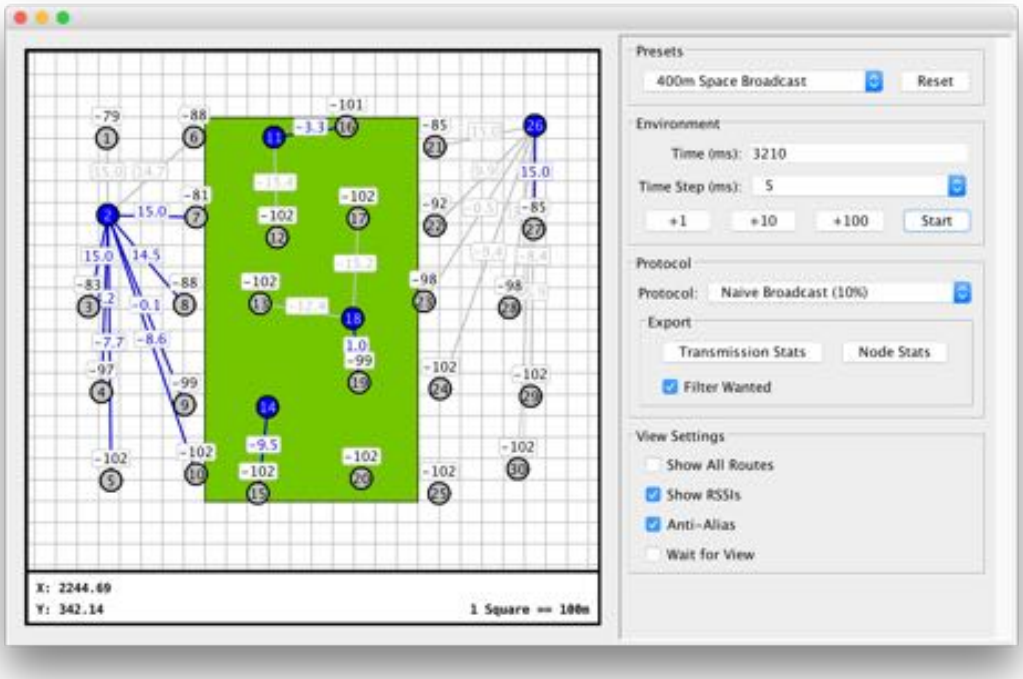


FIGURE 4.3: Simulator's GUI interface running the naive (ALOHA) protocol.

Chapter 5

MAC Protocols

5.1 Considerations

PHY testing indicated that LoRa is suitable for single-hops in a system with sparsely separated radios; ranges of 500m can be expected for SF = 11 regardless of the propagation environment. However, even with the 1600m range offered in free space, single-hop point-to-point coverage to all radios will be unlikey for most scenarios, leading to the expected dynamic mesh topology. Three contention-based MAC protocols are considered to handle the requirement of data transfers to physically close neighbours: a base case (ALOHA), a common approach (CSMA) and a scenario specific approach (LLBP).

The LoRa configurations considered for use are the abstracted DRs from LoRaWAN [10]; however, DR0 and DR2 are disregarded due to their lack of respective SF performance. Although small packets (< 20 bytes) have been shown to have slightly higher receive probability, they are not practical with transmission overhead, therefore packets are allowed to be any size (< 255 byte limit).

As opposed to the LoRaWAN duty cycle manager (see Figure 2.3), all protocols use a custom duty cycle manager proposed in Appendix F. This is more flexible because it considers airtime over the full enforcement interval (d_i), allowing multiple sequential transmissions without silence; this is a requirement for bulk transmissions or protocol handshakes. The simulator provides support for both models.

5.2 Approaches

5.2.1 ALOHA

Uses LoRaWAN's principle of just sending data periodically provided the duty cycle limit allows. No channel sensing is carried out, but transmissions will be delayed if the transmitter is synchronised at the scheduled time. For purposes of testing, the minimum spacing between transmissions is that enforced by LoRaWAN. Therefore, in the case that any backoff occurs, less than the duty cycle's maximum worth of data will be transmitted. MAC performance purely relies on each radio only using a small amount of airtime. As all radios must be on the same configuration the method can only use a single channel and DR – no mechanism is used for dynamically agreeing parameter changes. Consequently, a fixed duty cycle of up to 10% is supported on band h1.6. Test packets use random payload sizes. All packets are treated as data, acknowledgement and retransmission overhead is not considered. TP is assumed to be fixed to 14dBm for all scenarios.

5.2.2 CSMA

Uses the ALOHA approach but does carrier-sensing using LoRa's CAD process immediately before transmissions. On detection, random backoff will occur before the CAD process repeats; this is a simplified approach to that proposed in [16]. The PS are increased from 8 to 32 to increase likelihood of preamble detection with the drawback of increased airtime overhead. When all nodes are within range of one another the vulnerable collision period is very short; provided no failed synchronisations take place, collisions should only occur if two transmitter's CADs overlap, however collisions resulting from the hidden node problem are not avoided. Other mentioned ALOHA limits apply.

5.2.3 LoRa Local Broadcast Protocol (LLBP)

This is a bespoke approach that makes use of dynamic channel and SF switching with the target of constricting receivability to local listeners. The protocol uses two bands: a management-band (h1.4) and a data-band (h1.3). The former is treated as a single CSMA channel, whereas the latter allocates 15 channels at CFs

$865.1 + 0.2 \cdot C_n$ (125KHz + spacing). Optionally, an extra high-rate channel band could be allocated in band h1.6 for critical communications.

In principle, the protocol announces when a transmitter has a large chunk of data to send using a data announcement packet (DAP); this contains the LoRa configuration to switch to and a description of the data being sent (Figure **??**). When a radio receives this packet it can make its own decision on whether it wants the data (using the wanted criteria); if it does not it can ignore the message and will never even see any data packets, otherwise it can switch configurations and only see those packets.

As opposed to agreeing the configuration with a flurry of transmissions at one point, the configuration is decided by the transmitter, using its prior knowledge of surrounding transmitters. This knowledge is gained through periodical heartbeat packets (Figure **??**), sent on the management-band; these provide SNRs and locations. The fastest DR that can serve all wanting receivers is picked, along with a random data-band channel. There is no guarantee the channel is free but with the number of available channels, and a compulsory 1% duty cycle across all of them, collisions are unlikely. DR is selected such that $\min(\text{SNR}) > (D_l + 2.5)$; this is the point where PRP should be approximately 100%.

The side effect of choosing the fastest DR is less airtime, although this could be used to achieve more throughput, instead it is used to keep interference times to a minimum for the same data. The approach is high-risk, in the event a DAP is missed, or prior knowledge is incorrect, intended recipients may not receive any data. The number of DAPs sent can be increase to reduce the probability of the first case. However, increasing heartbeat frequency for the second case is problematic as it increases the likelihood in DAPs getting missed. Like with the other protocols, acknowledgements and retransmissions are not attempted. Heartbeats and DAPs are both considered as overhead.

5.3 Test Methodology

The simulator provides a number of environment presets. That most applicable to protocol testing, `LargeDataBroadcastTest`, provides automatic generation of an $[x \times y]$ radio grid where spacing (S) is the same $\pm 20\%$. It provides four obstruction options: no obstructions, full-forest coverage, half-forest left-side and half-forest central, where $\beta = 0.55$ for all; see Figure **??**.

If radios were always able to use their fastest configuration, it is highly unlikely a protocol with overhead would outperform one without. To alleviate this in testing, S was set to $400m$; ensuring that in-forest radios needed DR1 to communicate. This gave transmissions that did not pass through forests the opportunity to increase DR. Purely considering how many radios get a transmission is of little help; if local-interest data is received 1500m away, from the receivers perspective the transmission is unwanted and increasing the chance of a local collision. To this end a wanted criteria is defined, this is the distance between transmitter and receiver for which data is considered helpful; this is set to $500m$ (the transmission limits).

The defined test cases for protcols to execute were. ALOHA and CSMA were each trialled with 1% and 10% duty cycles The number of transmitted DAPs is also compared 1 v 2. This is compared to if the heartbeat packets were not required and all radios had perfect data. This demonstrates that fundamentally the protocol appears to perform less strongly, problaby because msising one packet results in many packets getting lost.

The test data that can be exported

% of intended recipients received for each message Reasoning for failed receive: Insufficient SNR (out of range), CRC fail (bad luck), Sync Collision/ CRC from interference

Missed == This implies the receiver was busy, too many transmissions going on at the same time

No Preamble == This implies the SF is not adequate

Protocol

The protocol Total helpful throughput number of bytes, number of packets

To make effectiveness comparisons fair they are

Verification of the protocol is to be gauged on - % of wanted packets received - % of wanted bytes received - Total bytes received - Total packets delivered - Number of collisions on wanted messages

5.4 Discussion

Approaches suited to DTN

Chapter 6

Conclusion

Provided a flexible research platform both physically and simulator Demonstrated how lora features can be taken advantage of

Demonstrated how hard it is to actually make use of parameters Future work: Incorporation with a routing protocol Real world testing

Incorporation with more realistic test data, have testing methodologies that carry out a combination of local broadcasts, unicasts and

When broadcasting a packet by packet basis is better, when agreeing parameters a more structured unicast system may be required.

It is highly likely that with substantial optimisations that LLBP could be made good. Only contention based Initial attempts attempted to provide slots to listeners though this didn't work Might be that it's only efficent to unicast data with different parameters (so that it's reliable)

The infrequency of possible transmissions stop this being possible. Mobile-LMAC is a time division approach that can adapt transmission slots dynamically, however there is no mechanism to adjust

Chapter 7

Retrospective

This was very much a research-led project, and from conception, was always open to change. The most major change from the brief (Appendix G) was to prioritise simulator creation over testing protocols directly on hardware. This was required as it became clear that the determined problem, managing channel access, required an infeasible number of test devices. Even if sourcing these had been possible, the protocol would need to be nearly fully defined before this stage of testing was sensible. The possibility of this was raised during the risk assessment carried out in the initial project stages. The full risk assessment, marked up with comments on actual effectiveness of risk mitigation, is in Appendix I.1. As, initial plans only considered basic simulator logic and the change meant a more feature-full simulator with radio models was required, work was re-prioritised to handle this. This is most clearly demonstrated by comparing the planned Gantt chart (Figure H.2) to the actual progress Gantt chart (Figure H.3). Unfortunately, this did lead to limited time for feeding back test results into informing the protocol design; ultimately, leading to a protocol that did not out perform simpler CSMA approaches.

The project required many disciplines. Prior to this project, I had considerable experience in Java development and embedded programming but only minor experience in electronics. Whereas, principles of ad-hoc networking and RF communications were completely new. Reference textbooks including: [27], [32] and [37] were key to building this knowledge. Also gained were skills in MATLAB and statistical analysis.