A distributed reception architecture for low-power wide-area networking

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Adwait Dongare

B.Tech., Engineering Physics, Indian Institute of Technology, Bombay

Carnegie Mellon University
Pittsburgh, PA

August 2020

Acknowledgements

First and foremost, I want to thank Anthony for being an excellent mentor and advisor. His hands-on mentoring made me feel independent, but always within reach of support should I ever want it. I am glad that he could introduce me some of his innumerable interests – biking, coffee, photography and many, many others. Some of my most memorable graduate school moments involve Anthony working alongside us late into the night while preparing for a demo or deadline. Some others involve one of the many videos he created with a tasteful choice of music in the background.

The rest of my committee – Swarun Kumar, Peter Steenkiste and Mani Srivastava – have provided invaluable help during the writing of this thesis. It has been an excellent experience collaborating with you, debating about various topics and building many interesting things along the way. It has been exciting to collaborate and discuss various ideas, both big and small, with Swarun and his research group. I want to thank Mani for leading the Roseline project; this project was how I met my earliest collaborators and helped me take my first steps in academic research. Peter provided many keen insights during the later stages of my thesis that helped me become unstuck at some critical moments. Many of the faculty at CMU have been my mentors and I would particularly like to thank Bob Iannucci, Vyas Sekar, Raj Rajkumar, Brandon Lucia and Bruno Sinopoli for helping me out during these times.

Our lab group has always been very dynamic and some of my work would have been impossible without Anh Luong, John Miller, Artur Balanuta, Nuno Periera, Craig Hesling, Nick Wilkerson, Patrick Lazik, Niranjini Rajagopal and Rahul Sharma. I would also like to thank Elahe Soltanaghaei, Max Buevick, Oliver Shih, Akarsh Prabhakara, Courtney Kowaluk, Raewyn Duvall, Eric Reibling and Luis Pinto for making my time in CIC (and now remotely) as fun as it was. All of you have been more than just coworkers, and I've enjoyed our many conference travels, stressful deadlines, happy hours, board-game nights, bike rides and cooking sessions together. I'm also glad we had Toni Fox and Chelsea Mendenhall figuring out how to support some our crazy endeavours, but also making sure we didn't get into too much trouble along the way.

I would like to thank my mother, father and grandparents for supporting me in my earlier years. Without their support, I would neither have been capable nor had the motivation to start a Ph.D. program.

Pittsburgh and CMU would not have been as memorable without all of the fantastic people I've met in the past few years. I have had many a memorable conversations with Diana Zhang and Emily Ruppel who have always been supportive, particularly when I needed it the most. Thank you Luis Oliviera, Rita Lopez and Tom Jackson for your delicious food and for humoring me during my search for croissants on some of our bike rides. Meghan Clark, you have been an excellent motivator, supporter and friend all the way from the other side of the country. Damiao Rodrigues, Soo-Jin Moon, Saurabh Shintre and Sandeep

D'Souza have been some of my earliest friends and stuck with me through my entire time at CMU. I would also like to call out some of my roomates – Eben Hoffer for introducing me to many of the creative activities happening around Pittsburgh and for introducing me to the world of plays; and to Darshan Patil for tolerating me through the Coronavirus shutdowns. I would also like to thank some of my friends from undergrad, particularly Chirag Modi, for keeping in touch despite the time and distance between us. I would also thank everyone I've encountered in CyLab, CIC, our collaborators in California, the SubT Explorer team and my CMU-Portugal friends for accepting and involving me.

The last few years of graduate school had been challenging, and volunteering at the National Aviary provided a healthy break from my routine. In addition to being a beautiful place, the volunteers and staff created a true community. The time I spent at the Aviary not only reenergized me but also let me learn about very different topics and provided an opportunity to do something for the community.

These activities helped me remain motivated and happy during challenging times, so I wish to share them: This excellent brownie baking recipe is adapted from BraveTart [65]. Sift together a cup of flour (125 g) and 1 1/3 cups of cocoa powder (115 g) in a large bowl and set aside. Heat 3 sticks of unsalted butter (340 g) in a saucepan until it browns, turn off heat and add a cup of chopped chocolate (170 g). Stir until you get a consistent, thin mixture. In your main mixing bowl, mix 2 cups of white sugar (400 g), 1/2 cup of brown sugar (110 g) and 1 3/4 tsp of kosher salt (4 g). Mix in 6 large eggs, 1 tbsp of vanilla extract and 1 tsp of instant coffee powder. Add the chocolate butter and mix until the mixture is thick and fluffy. Stir in the cocoa-flour mixture, small parts at a time, until everything is consistent. Pour into a 9x13x2 inch baking pan and bake at 350° F (180° C) until the internal temperature is 205° F (96°) – approximately 30 mins. Wait to cool before cutting. My other activity involves bicycles: If you find yourself in Pittsburgh with a bike and some time to spare, I would recommend a long ride to the Allegheny Observatory in Riverview Park and to Hartwood Acres on the North Side. Though the climbs may be challenging (and will feel like an accomplishment), the way back is downhill and extremely thrilling.

This work was made possible through the support of various programs and funding agencies. My research was supported by Roseline, a CPS Frontiers project sponsored by NSF under award CNS-1329644; the CONIX Research Center, one of six centers in the SRC JUMP program sponsored by DARPA; TerraSwarm, one of six centers in the SRC STARnet program sponsored by MARCO and DARPA; and by the Bosch Research and Technology Center.

Abstract

A large number of pervasive sensors will manage and optimize the infrastructure in future smart cities. Ideally, we will be able to sense everything from bridges, buildings and firetrucks, all the way to trash cans, bicycles and street lights. The growing field of low-power wide-area networking (LPWAN) looks at the challenge of wirelessly communicating with a large number of low-power, simple devices over long periods of time. These city-wide networks need to provide two important functions: extracting data from sensors and keeping track of where they are located. We explore the capabilities of these networks by deploying and evaluating our own OpenChirp LPWAN network around the Carnegie Mellon University campus. Though promising, these networks still suffer from problems of coverage, conservation of device battery and the lack of an ability to accurately localize devices. This thesis will explore the design space and potential of using tightly coordinated distributed gateway receivers to efficiently decode data, as well as to locate the transmitting devices. It is challenging to develop distributed radio systems that utilize physical layer signals without overwhelming the backhaul network. We first introduce a coherent combining system, called Charm, that improves network coverage, data rates and battery life of deployed devices by selectively collating receptions from multiple receiving gateways. In an indoor environment with high multipath, accurate time synchronization at nanosecond accuracies is challenging, but plays an important role in enabling both localization and coherent combining. Thus, we develop a time synchronization platform, called Pulsar, for nanosecond-scale synchronization of radios. Next, we show how we can extend our distributed reception system with time-synchronized receivers to improve device localization through the use of time-difference-of-arrival features in challenging urban environments. With multiple receivers that are accurately time synchronized, we see the potential to significantly improve LPWAN performance and effectively localize transmitting devices.

Contents

List of Figures and Tables

Chapter 1

Introduction

The ability to sense our surroundings and adapt to its changes has shaped the path of life on Earth. It led to the evolution of sophisticated senses like smell, hearing and vision; senses that keep us aware of our immediate surroundings, point us toward opportunities and keep us away from danger. Naturally, we desire to grow our awareness to beyond just our immediate surroundings, to everywhere.

As inventors, we create small devices that continuously sense the world, even when we aren't nearby and in places we've never been. These *sensors*, however, are effective at their task only if they can collect and process the data they have gathered. This thesis will explore the challenge of developing a communications system for innumerable small sensors everywhere, which enables us to gather information about our immediate surroundings and the world.

1.1 Motivation and Background

Low-power, low-cost and pervasive telemetry still remains a bottleneck in how we sense and manage our physical infrastructure. Taking Internet-of-Things (IoT) concepts outside of buildings and to massive scales will have deep implications for monitoring utilities (water, electricity, gas), sewage, roads, traffic lights, bridges, parking complexes, agriculture and waterways. Current approaches for telemetry rely on cellular infrastructure or nearby WiFi that have been optimized for high-throughput applications. In terms of energy, cost and scalability per bit of information, these existing radios will not be able to support long-term deployments of battery-operated sensing devices. Fortunately, the same radio technology that has resulted from advances in WiFi, Bluetooth and LTE has now made it possible to create new chipsets that trade off throughput for range. These low-power wide-area networking (LPWAN) radios are able to transmit over distances as long as 10 km with the same power consumption (or less) than what is used by typical WiFi radios. They also operate at lower frequencies (below 1 GHz) that are able to penetrate more deeply into

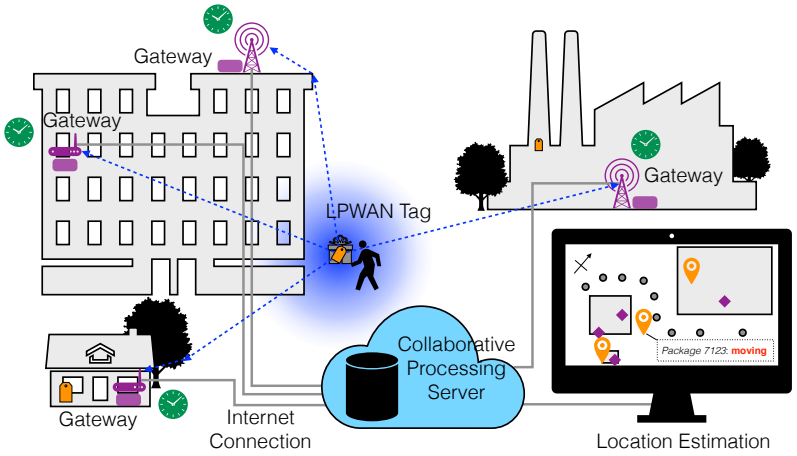


Figure 1.1: Vision for distributed reception with LPWANs

structures. These radios use the ISM bands (433/915 MHz in the U.S. and similar frequencies elsewhere), which means no licensing is required for public access.

Sensors deployed out in the world are heavily constrained devices; they rarely have access to wired power or communication, and they must be produced at low cost, operate without maintenance and be compact enough for their application. As examples, we want sensing systems to keep track of the state of our bridges, agriculture, bicycles, packages, forests and much more, for periods of time ranging from days to multiple years. Ironically, in critical applications such as sensing the state of power utilities, sensors are designed for battery operation (for reasons of redundancy and system isolation) despite being deployed right next to a power line.

The paradigm of low-power wide-area networking (LPWAN) has been making great headway towards addressing the challenge of developing the next generation of communication networks for sensing devices. LPWANs focus on providing low data-rate wireless communication for a large number of simple low-power devices that are deployed over very large areas. Startups like Sensus [84] and Samsara [78] have started providing city-scale sensing capabilities to utility and logistic companies. In contrast to the closed and proprietary nature of cellular networks that function at similar geographic scales, a number of LPWAN protocols and hardware are open and accessible. In addition, they are often designed for use in the unlicensed and shared ISM band spectrum (915 MHz in the US, 868 MHz in the EU, etc.) which enables them to cover relatively large areas many kilometers away. For this and many other reasons, we would like to see LPWANs deployments start up and grow everywhere, both indoors and outdoors, across urban, rural and uninhabited areas. However, there are some pressing questions about the scalability, performance and capabilities of these networks that must be addressed for widespread adoption. This thesis will attempt to tackle some of these major questions.

Powerful Gateways and Highly Constrained Clients: Considering the constraints on our sensing devices, communicating reliably at such large distances might seem like an impossible goal. However, the heterogeneous nature of LPWAN networks makes this concept possible; complex, powerful gateways permit our low-power devices to selectively sense, compute and transmit bursts of information while sleeping for a majority of the time. If we look closely at some of the gateway hardware designs, they closely resemble software radios and could provide almost unhindered access to the physical layer. We see a great opportunity in making these gateways more capable, either by adding auxiliary hardware or by providing additional features like time synchronization.

Collaboration Between Gateways: Many LPWAN protocols can deal with multiple receptions of the same information on different gateways by moving away from the traditional client-basestation hierarchy towards a client-gateway-server architecture. Though we observe some advantages of diversity gain with multiple gateways, these gateways do not truly share information between themselves. We see another opportunity here if the different gateways were to share richer information about the signals they observe. The modified gateways we previously discussed give us access to the physical layer of the signal. Physical layer access combined with collaboration between gateways helps us develop a distributed receiver, one whose performance and capabilities are beyond those of any individual gateway.

Synchronization: We desire for our distributed receiver networks to be ubiquitous, which implies having well synchronized gateways in indoor as well as outdoor spaces. Currently, GPS-disciplined oscillators are the default choice for any communication systems that need good synchronization at a large scale. Current market trends indicate that there are a large number of LPWAN networks planned in large indoor industrial Internet-of-Things deployments. Unfortunately, GPS-based synchronization requires access to the open sky and cannot function indoors. Thus, we will also explore developing a time-transfer platform using precise clocks and wideband radios that can bridge the synchronization gap between a gateway located indoors and an external GPS receiver.

Localization: In addition to fixed devices monitoring large infrastructure, we expect many LPWAN devices to be mobile tags attached to items of importance. The ability to localize these tags is extremely valuable for tracking packages, bicycles, equipment, valuables, etc. Though there are a number of proposals for localizing LPWAN devices using signal-strength, beacon-proximity and even time-difference-of-arrival (TDoA), they suffer from large errors of hundreds of meters due to dynamic environments and multipath propagation [54, 19]. We could augment our aforementioned distributed receiver with time synchronization to enable capturing signals on multiple gateways at the same exact instance. This enables the use of advanced methods like fingerprinting using TDoA information to localize these mobile tags in such challenging environments. The opening of unused TV UHF spectrum for use by unlicensed communications (called

whitespace [30]) also presents an opportunity to gather a variety of frequency-dependent information about the environment that can help us combat the challenges of multipath. To this end, the industry has already released a number of radio ICs capable of spanning very wide frequency ranges [83].

1.2 Problem Statement

The vision for this thesis is illustrated in Figure 1.1. We advocate for the use of well-synchronized LPWAN gateways located in a variety of environments that tightly coordinate to improve range, energy efficiency and capacity. These gateways have the capability to capture and share information about wireless LPWAN signals at the physical layer to create a *distributed receiver*.

In this thesis, we will answer the following two questions:

1. What are the requirements to create a practical distributed receiver infrastructure for LPWAN systems?
2. What additional functionality and benefits could such a distributed receiver provide?

Unlike their demodulated and decoded forms, radio signals at the physical layer directly represent the electromagnetic waves they are composed of. Hence, these representations also follow fundamental physical laws like superposition and wave propagation. In a distributed receiver, we would observe the same source of radio signals (the transmitter) at different points in space (at the receiver). If these signals could be captured at all the receivers at the same instances of time, we have at our disposal simple but powerful mathematical techniques like weighted vector addition that would enable us to build a radio receiver system with better sensitivity and better resolution than any individual receiver in the system. However, access to the physical layer and time synchronization are essential.

Access to the physical layer and GPS-like time synchronization could be provided by a modern software radio platform like a USRP or PlutoSDR. However, continuously uploading all wireless signal captures from multiple gateways to a central server for processing is impractical. Thus, a practical distributed receiver would have some components of the signal processing chain performed on each gateway independently before uploading the relevant output to a collaborative server for further processing.

The benefits of a distributed receiver for LPWANs are vast and interrelated. For example, such a receiver would be able to boost received signal power as a direct result. Such a development has a number of additional positive consequences: it reduces energy consumption due to increase in data rates, it reduces contention in the spectrum for the same reasons and it increases the coverage area of the network. Similarly, TDoA localization might seem like an obvious added capability once we have time synchronization, but it

would have hundreds of meters of location error in a multipath ridden urban environment. Thus, we have to ask if it is possible to go beyond simple time-difference-of-arrival.

In the following chapters, we will explore these questions in depth and also attempt to understand the resulting consequences.

1.3 Thesis Statement

A constellation of synchronized receivers spread out over large areas can collaborate by selectively sharing raw radio signals for the purpose of improving the coverage, data rates, energy efficiency and locatability of wireless low-power wide-area network devices.

1.4 Thesis Contributions

The contribution of the various chapters in this thesis are described below.

- **Chapter 2:** A study of the capabilities of low-power wide-area networking
  - A study of LPWAN, LoRa and LoRaWAN to understand the regulations, communication protocols, network performance and device energy consumption.
  - Through the real deployment of a campus LPWAN, demonstrate that coverage is non-uniform which affects device battery life.
  - An analytical exploration into the scalability of LPWANs, which shows the importance of collaboration, power control and strategic distribution of gateways.
- **Chapter 3:** Distributed reception on the receivers through coherent combining
  - A technique that leverages the geographical diversity of unplanned, user-deployed gateways to enable joint decoding of weak transmissions. This improves battery-life for users in the network and increases the coverage area.
  - A hardware platform and the underlying algorithms for detecting weak LoRaWAN transmissions locally at the gateway.
  - A software architecture that builds atop of LoRaWAN to enable joint-decoding of signals in a scalable manner.
- **Chapter 4:** Precise time and frequency synchronization of gateway-class devices

  - A novel hardware platform that is able to perform wireless time-of-flight propagation-aware clock synchronization at better than 5 ns resolution per communication hop that can be easily integrated with existing software-defined radio (SDR) systems
  - An end-to-end analysis and evaluation of timing uncertainty provided by the platform
  - A technique to handle the clock phase offset problem introduced by the use of PLLs.

- **Chapter 5:** Channel fingerprints to aid localization of low-power transmitters
  - A study of radio localization techniques and identifying that a receive-only time-difference-of-arrival method would be the best fit for LPWAN device localization.
  - Radio propagation simulations in an urban setting that show the limitations of physics-based localization and vouch for the use of fingerprinting methods.
  - Developing and evaluating a matched filtering-based technique that when combined with the use of multiple frequency bands can extract channel response features for fingerprinting.

In Chapter 2 we will explore the performance and capabilities of LPWANs through the deployment of our own OpenChirp network around the Carnegie Mellon University campus. We analyze the scalability of LPWANs when the spectrum starts getting congested. These help us motivate the need for coherent combining. In Chapter 3, we demonstrate distributed reception on multiple gateways using coherent combining through our system, named Charm. Next, we realize that we could extend the coherent combining system to other forms of sensing, like the localization of transmitting devices, if all the gateways were synchronized. While extremely promising outdoors, we see the lack of time synchronization indoors being a hindrance. Chapter 4 presents the Pulsar platform that enables precisely synchronizing LPWAN gateways and other communication devices in indoor and other GPS-denied scenarios. In Chapter 5, we explore the challenges of localizing low-power devices in multipath prone environments and present a technique to extract channel response fingerprints.

Chapter 2

Low-power wide-area networking

In this chapter, we will explore the paradigm of low-power wide-area wireless networking designed for the purpose of communicating small amounts of data over large distances in a scalable manner. We first look at the tradeoffs in the architecture of low-power wide-area networks which enables reliable communication with thousands of low-power devices. We then describe the OpenChirp LPWAN deployment around the Carnegie Mellon University campus to understand LPWAN's real-world performance and the challenges that must be overcome to enable wide adoption. Finally, we analytically explore the scalability limits of LPWAN networks. In the majority of this thesis, we focus on LoRaWAN, a partially open-source LPWAN implementation from Semtech. However, most other LPWANs face similar challenges and provide similar opportunities.

2.1 Contributions

The contributions of this chapter are as follows:

- A study of LPWAN, LoRa and LoRaWAN to understand the regulations, communication protocols, network performance and device energy consumption.
- Through the real deployment of the OpenChirp LPWAN, demonstrate that coverage is non-uniform which affects device battery life.
- An analytical exploration showing the benefits of LPWAN networks collaborating with other networks in the spectrum, implementing proper power control and having gateway deployments closely mirroring the distribution of devices if they are to support a large number of devices.

2.2 Related Work

A number of new radio technologies have been developed to provide robust, low-power and low-cost connectivity to a large number of devices[7]. In this section we describe other existing technologies and frameworks in this space.

2.2.1 LPWAN Technologies

Recent years have seen much interest in LPWANs, including the development of new hardware and standards. Sigfox [88] provides an ultra-narrow-band LPWAN built on IEEE 802.15.4 radios. It operates in the unlicensed 868 MHz and 915 MHz spectrum in Europe and the US respectively. Devices can communicate over long ranges (tens of kilometers) in a star topology, using low data rates (100 bits/s) and narrow bandwidth (100 Hz/channel), which enables extremely low-power communication. Sigfox deploys and operates gateways, functioning similar to cellular operators. Unlike LoRaWAN, the network layer is proprietary.

LTE Cat-M1 or enhancements for machine type communication (eMTC) is the 3GPP adaptation of LTE for LPWAN [37]. For low cost and energy reduction, devices operate on lower bandwidths (1.4 MHz/channel) with low data rates (<1 Mbps) and half-duplex communications. Regular LTE functionality like mobility and hand-off are still supported. eMTC additionally provides power-saving modes and extended discontinuous receptions that allow devices to enter extended periods of deep-sleep without losing their network registrations. Narrowband IoT (NB-IoT) [73] is similar to eMTC but operates at even lower bandwidths (180 kHz/channel) and lower data rates (20 kbps) in the licensed LTE spectrum. Mobility is sacrificed in favor of better indoor coverage and support for larger number of devices. These networks function in the licensed LTE spectrum owned by cellular operators, which will be regulated and subject to service contracts. Although they consume lower power than cellular communication, these technologies require devices to periodically wake up to synchronize with the network, significantly affecting battery life.

Several recent measurement studies have been conducted to evaluate the performance and range of LPWAN networks [66, 92]. Early pilot deployment efforts are also underway, with Sigfox deploying their hardware to connect security alarms to the cloud in Spain [79], smart blood refrigerators in the Democratic Republic of the Congo [72] and smart city applications [55]. These efforts motivate the challenge of limited range, performance and battery-drain of LPWAN clients.

Due to the extensive interest in LPWANs, a number of other technologies are also already deployed and in development [74]. These are either closed protocols or are yet to gain wide adoption.

2.2.2 LoRaWAN-Based Networks

We describe two examples out of a number of LoRaWANs currently deployed around the world. The Things Network [2] is a community-driven LPWAN initiative started in Europe. They provide gateways and end-devices along with online infrastructure for device management and communication. Symphony Link [47] is a similar network promoted by Link Labs which added listen-before-talk functionality to LoRaWAN as well as a few other non-standard enhancements to improve bandwidth utilization.

2.2.3 Scalability of LPWAN

A number of existing works have explored the question of scalability of LPWAN networks. [13] and [93] perform an exhaustive study of LoRa packet collisions, develop a LoRa MAC simulator to simulate packet collisions based on their observations and conclude that multiple gateways are necessary to provide scalable coverage. In [35], the authors develop a metric called bitflux to determine that reducing the effective range of gateways might be a strategy to deal with congestion. The authors of [63] perform exhaustive studies on inter-technology interference between LoRa and IEEE 802.15.4g to determine that LoRa communication causes significant interference to 802.15.4g, but it is not the case the other way around.

2.3 Characteristics about LPWANs

This section will explore a number of characteristic design and system architecture decisions that set LPWANs apart from other wireless communication networks.

2.3.1 Long-Range Communication

LPWAN devices are designed for communication over many kilometers. It is achieved by the use of simple narrowband modulation schemes, low-duty cycle high power transmissions and heavy encoding. Thus, LPWANs can be designed so that devices will communicate directly with a gateway, in a star topology, rather than going through multiple hops. As a consequence, we can avoid many other complexities and costs associated with multi-hop networking, e.g. maintaining neighbor lists, keeping clocks active and relaying transmission from other devices. However, in contrast, each transmission costs a significant amount of energy and can potentially lead to spectral crowding as described below.

2.3.2 Unlicensed and Shared Spectrum

Most LPWANs are meant to operate on unlicensed ISM band frequencies. As there is no licensing involved, this reduces the cost of deploying and maintaining a network. Additionally, the same ISM band frequencies are reserved across an entire country (915 MHz band in the US) or even across the world (2.4 GHz). Thus, the same radio hardware can be deployed at larger scales without having to modify it for every deployment. The advantages are uniformity in design and lower cost of devices from economies of scale.

Unfortunately, the ISM band spectrum is shared by many unlicensed communication technologies which use a variety of strategies to manage spectrum use. LPWAN communication on the ISM band will not only suffer from interference but cause a significant amount of interference to other networks (e.g. IEEE 802.15.4g) due to its higher power spectral density and longer frame times. Many have asked if LPWANs can scale; we will explore this problem further in Section 2.5.

Frequencies previously used for television transmissions, called whitespace, have been opening up for unlicensed usage. Unlike the ISM band, available whitespace spectrum varies regionally. A number of LPWAN technologies have developed variable spectrum hardware [83] and frequency agnostic protocols to leverage this additional spectrum. Although currently less congested, we expect whitespace frequencies to suffer from similar interference once it is more broadly used by other unlicensed communication systems like municipal WiFi.

2.3.3 Focus on Low-Power and Long Lifetime Operation

One of the primary use-cases of LPWANs is remote telemetry, where wired power is unavailable and even physically reaching the device might be difficult. LPWAN clients are often designed to operate on battery power or energy harvesting and function without any physical maintenance for long periods of time. Batteries hold a limited amount of charge (which degrades with time) that directly determines how long a device will function before requiring a recharge or replacement. Energy harvesting generates a limited amount of average power that must be properly managed or stored to perform more expensive operations like radio transmissions. In these scenarios, the network must be designed with low-power operation as a focus. Otherwise, devices will require maintenance very frequently or be unable to operate at all for their intended purpose.

Deployments can target 10 years of device lifetime, as new batteries with proper management can retain their charge for this period of time. In the case of energy harvesting devices which could have indefinite access to energy, we're still limited by the life of the underlying electronics.

The hardware, communication protocol and software must be developed with low-power as a focus;

otherwise, any inefficiencies will drastically reduce device lifetime or ability to operate. LPWAN devices use low duty-cycle bursts of communication, so that high power peripherals on the device are only active for a short period of time. High precision clocks consume significant amounts of power while they're active, and LPWAN devices must be designed to function without continuous access to accurate time. This currently limits us to the use of simpler MAC protocols like ALOHA or CSMA. Keeping the radio subsystem active for carrier sense, acknowledgements and retransmissions increases power consumption drastically, so many LPWAN protocols either eliminate or heavily restrict the use of these protocol-level checks.

We will explore low-power operation in more detail in Section 2.4.3 with our LoRaBug hardware platform.

2.3.4 Sophisticated Radios with Simple MAC Protocols

In the previous section, we covered LPWAN's use of simpler MAC protocols like ALOHA and CSMA to lower the energy consumption of devices. However, these simple protocols are supported by sophisticated strategies that makes them fairly robust in real deployments. Frequency hopping safeguards communication from sustained interference on some frequency bands. Adaptive data rate selection is used to account for changing environmental and radio propagation conditions. Finally, transmitting power control ensures that the radio electronics on receiving gateways are not overwhelmed by nearby transmissions.

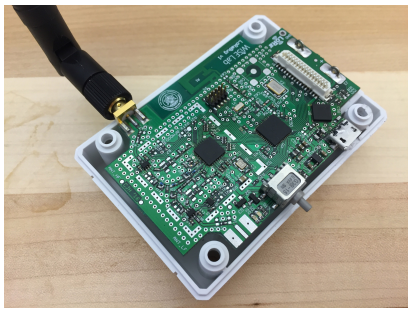
The choice to use some of these sophisticated strategies is left to the device designer, as is the sophistication of the implemented strategy. This is referred to as device-centric design. As a result, a bare-bones device might trade off the complexity of frequency hopping and adaptive data rates for less robustness in communication. Such a device can operate on the same network as a more capable device that smartly tunes various communication parameters based on existing conditions. Unlike the other strategies, some level of transmit power control must be implemented by every device in the network to maintain scalability in a congested spectrum.

2.3.5 Heterogeneous Hardware

An LPWAN network will typically be comprised of three different classes of devices: (1) very low-power client devices, (2) gateways with more sophisticated radio hardware that has the ability to handle multiple communication streams and (3) a server computer that manages the network. The designs of these different classes have very different optimizations. While the server is generally a regular computer that handles requests over the internet, the other two classes have unique design trade-offs.

Client devices are designed to be compact for deployment, be low cost and consume very little power. In contrast, gateways are designed to support a large number of devices over a wide range of frequencies

(a) Deployed gateway



(b) LoRaBug client

(c) Side-by-side comparison of gateway and devices

Figure 2.1: Photos of OpenChirp gateways and devices

simultaneously. Since most gateways will not be moved after deployment, they can have large specialized antennas, continuous access to power and network connectivity. Gateways are also not as cost constrained as client devices, which opens up the potential to add many different capabilities through additional hardware or software. A major contribution of this thesis is to identify which capabilities could be added to gateway designs to provide significant value.

On the negative side, the limited capabilities of highly constrained client devices prevents us from using many commonly used encryption techniques and MAC strategies that could drastically improve the security, privacy and reliability of the network.

2.3.6 User-Deployed Gateways

Unlike cellular networks, LPWAN gateways can be owned and managed by anyone, including the users themselves. Although, new LPWANs are sometimes seeded by an centralized deployment, the expansion and maintenance of a network can be undertaken by the users themselves through the deployment of their own gateways. This is possible because although gateway hardware is very capable it is closer in price to a WiFi access point than to a cellular base stations. The role of a gateway in the LPWAN network is to function as simple forwarders (MAC-in-the-cloud means most networking decisions are made by the server) and they do not have to worry about complicated procedures like handoffs which makes adding and removing gateways less complicated.

The obvious advantage of this paradigm is that an LPWAN can grow organically if enough users are willing to acquire and deploy gateways. Community run networks, without a large corporation or government being involved are thus a possibility. The downside is that an area may have a large number of LPWANs that do not collaborate, significantly affect performance through interference and end up competing for spectrum.

2.4 The OpenChirp Network

OpenChirp is a prototype end-to-end LPWAN architecture built using LoRa Wide-Area Network (LoRaWAN) protocol with the goal of simplifying the design and deployment of Internet-of-Things (IoT) devices across large areas like campuses and cities. A deployment of OpenChirp across the Carnegie Mellon University campus consist of a number of custom LoRa gateways deployed at various locations, a variety of low-power sensing and control devices as well as a cloud server (LoRaWAN server combined with other software) that manages this network.

To better understand the environment and network performance, the deployment uses ruggedized outdoor gateways with additional hardware for localization, synchronization and spectrum capture. At the device-level, we developed and benchmarked LoRaBug, one of the first open-source LoRa hardware platforms. It is a LoRa client that can be extended with custom transducers, and can also interact with Bluetooth Low-Energy (BLE) devices. Developing our own LoRa client gave us complete control over the hardware and firmware which enabled us to focus on low-power operations of LPWANs without interference from other unnecessary factors. The software architecture of OpenChirp is designed to allow multiple users and groups the ability to provision and manage battery-operated transducers across large facilities like campuses, manufacturing plants or cities. Many of these capabilities were missing in earlier LPWANs.

We evaluate the system in terms of end-node energy consumption, radio penetration into buildings as well as coverage provided by the network around CMU's campus.

2.4.1 A Brief Introduction to LoRa and LoRaWAN

LoRa is an LPWAN technology developed by Semtech. LoRa's physical layer is based on chirp-spread spectrum (CSS) modulation, i.e. using a chirp signal that continuously varies in frequency over a narrowband (125 KHz, 250 KHz or 500 KHz) channel in the sub-GHz ISM frequency bands. This makes it resilient to

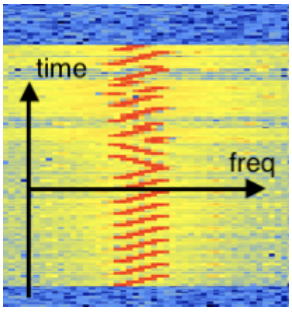


Figure 2.2: Spectrogram of a LoRa packet showing a preamble of upchirps, a synchronization header with downchirps followed by data encoded with offset upchirps. *Source: Matthew Knight, Reversing and Implementing the LoRa PHY with SDR,* https://www.youtube.com/watch?v=-YNMRZC6v1s

interference, multi-path fading and Doppler effects. A spectrogram of a LoRa packet is shown in Figure 2.2. Each data chirp encodes multiple data bits (more precisely, *chips*), with the number of bits encoded per chirp called the *spreading factor* (SF). For instance, at spreading factor of seven, each chirp encodes 7 bits with $2^7 = 128$ possible uniformly separated initial frequencies. A higher spreading factor, e.g. eight, encodes one more bit per chirp but also incurs double the transmission time, effectively halving the data rate. Increased spreading factors are used to simultaneously slow down transmissions and improve resilience to noise. Data rates can be varied between 0.3 kbps to 22 kbps by adjusting packet spreading factors (SF), bandwidth and power levels. A unique aspects of LoRa systems is that the gateway chipsets can demodulate on multiple channels and at multiple data rates simultaneously. This is possible because gateways capture a large bandwidth that consists of multiple neighboring frequency channels. Additionally, the independence of SF encoding enables reception of multiple transmissions in the same frequency band. This enables LoRa gateways to support extremely efficient star collection topologies.

In an analogy to the OSI communication stack, LoRa radios define the first and second layers (Physical and Data Link) of LPWAN. Layers three and four (Network and Transport) are analogous to the LoRa Wide-Area Network (LoRaWAN) protocol. LoRaWAN networks are designed to be simple star-topologies that have client devices directly communicating with a gateway that is connected to the internet over ethernet or cellular links. The protocol manages communication between gateways and client devices in the following ways: (1) establishing encryption keys for application payloads and network traffic, (2) device to gateway pairing assignments, and (3) channel, power and data rate selection. Part of our motivation for using LoRaWAN lies in its open nature and flexibility of implementation. LoRaWAN defines three main device classes that are broadly inspired by different MAC strategies:

1. **Class A:** Bi-directional end-devices with uplink followed by downlink.
   Similar to ALOHA and intended for ultra low-power sensors.

2. **Class B:** Bi-directional end-devices with scheduled transmission slots.
   Similar to TDMA and intended for sensors with actuators.

3. **Class C:** Always-on bi-directional devices.
   Intended for powered devices that require low-latency actuation

Each LoRa device in the system has a network communication and application encryption key. All packets are transparently sent from gateways to a LoRaWAN server without any local decryption to limit the potential risk of compromised clients and gateways. Similarly, a gateway only acknowledges a reception once commanded by the server. Packet decoding, managing acknowledgments and MAC parameters like

data-rate are decided at a LoRaWAN server. The LoRa community often refers to the system as having a "MAC-in-the-Cloud" design. An interesting consequence of this design is that multiple gateways could receive the same transmissions and only one acknowledgement would be send back to the device. This allows LoRaWAN to move away from strict client-gateway association. Additionally, as the gateway is not maintaining any credentials or state information about client devices, a single gateway can "support" a large number of devices as long as their transmissions do not contend in the air.

2.4.2 LPWAN Gateways

LoRaWAN Gateways are responsible for converting raw LoRa signals into digital data that is sent to a server. In LoRaWAN and may other LPWANs, gateways can be owned and deployed by anyone. In stark contrast to the low-power power devices they communicate with, gateways are expected to have access to continuous power, internet connectivity (through cellular, WiFi or Ethernet) and very sophisticated radio frontends. Even the oldest LoRa concentrator ICs [20] can support eight 125 KHz frequency channels with a variety of coding rates simultaneously. These gateways do not perform any heavy computations on the decoded data, store any device-specific encryption keys or device state, which allows simple embedded computers to be used inside gateways.

Our custom gateway is powered by a Raspberry Pi 4 (previously Raspberry Pi 3) connected to a custom LoRaWAN concentrator over SPI[1] and connects to the internet over WiFi or Ethernet. Power-over-Ethernet (PoE) simplifies deployment of the gateways. The first version of the gateway had a RTL software-defined radio for spectrum sensing and access to raw radio signals. This has since be upgraded to a PlutoSDR for better performance and access to wider bandwidths. The gateways are also equipped with a GPS radio for localization and time-synchronization. For extended use in rough outdoor environments, we hardened the deployed hardware (weather-resistant design) and software (watchdog resets). The gateway communicates with the OpenChirp network over a secure MQTT connection.

2.4.3 Low-Power Client Devices and the LoRaBug

A typical LoRaWAN client device will have a low-power MCU and a LoRa radio. It will also have a number of sensing and control peripherals for the device's intended purpose. There have been a number of recent products that either integrate the MCU and radio in a module (for regulatory compliance and simplicity) or attempt to implement the radio on the MCU silicon. Some devices will have a secondary radio e.g. BLE that can help with configuration, firmware updates and deployment. Most LPWAN devices will operate at

[1] Early LoRa reference designs had a USB-serial interface that dropped data

very low duty cycles, and thus the hardware and software on a device must be optimized so as to not waste any energy when the device is in it's sleep states.

We developed an open-source, low-cost, low-power, and extensible LPWAN end-node hardware platform named LoRaBug [1] shown in Figure 2.1b. The main motivations for the development of this platform are (1) ease-of-use in terms of registration, (2) expandability and (3) a well profiled reference firmware stack that can maintain low-power consumption. We envision OpenChirp simplifying deployment through a combination of BLE configuration of end-devices and a simple web-portal for registrations.

The LoRaBug hardware is housed in a small plastic enclosure that accommodates two AA batteries. The LoRaBug itself provides processing and communication while expansion modules provide the sensing and actuation functionality. These are attachable daughter boards containing application specific sensors and actuators. For example, we use an expansion module that has passive-infrared, temperature, humidity, sound, acceleration, and light sensors to monitor rooms in campus buildings.

The LoRaBug is powered by a Texas Instruments CC2650 microcontroller (MCU) with integrated 2.4 GHz IEEE 802.15.4 and Bluetooth Low-Energy (BLE) radios. It communicates to LoRa networks through a Semtech SX1276 LoRa radio. The node can be augmented with expansion modules for a variety of applications (e.g. environmental sensing, GPS localization and actuation). In addition to typical sleep states, the MCU has an ultra low-power sensor co-processor for sensor sampling and data aggregation, and a cryptographic accelerator that enhances the performance of security functions and reduces code-size.

The LoRaBUG firmware is built on top of the open-source TI-RTOS. Based on the application, the firmware can be configured from a minimal multitasking kernel to a complete network-enabled environment supporting low-energy operation. The LoRaBUG connects to networks such as OpenChirp (as class A or B LoRaWAN device) using the IBM LMiC library [42]. LMiC supports both over-the-air activation (network parameters shared by joining the network) and activation by personalization (network parameters directly stored in device). Mobile devices can interact with the LoRaBug using the MCU's integrated BLE radio that allows for easy configurations and communication. The BLE stack is a TI-RTOS extension and the firmware can be created with (using 13.9 KB Flash and 6.2 KB free SRAM) or without it (using 75.4 KB Flash and 10.6 KB free SRAM). Over-the-Air (OTA) updates may be performed if the flash space is partitioned into two parts for the running firmware and its update. Due to the limited size of available flash, we recommend adding external storage for OTA updates on the sensor expansion modules.

Energy Consumption

Figure 2.3 shows the energy profile of a LoRaBug functioning as a typical battery-operated LoRaWAN client. The device performs some local computation, sends a LoRa message with 8 bytes of data, waits for

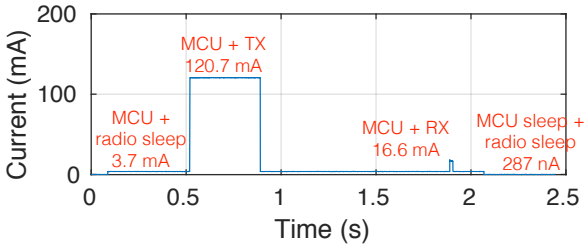


Figure 2.3: LoRaBug current consumption over time for transmitting a packet and then checking for an ACK from the gateway.

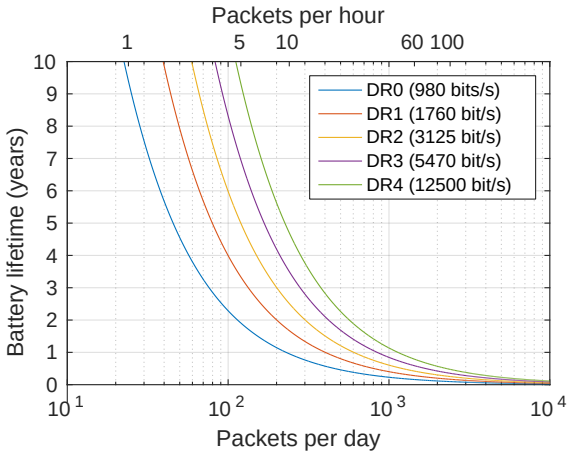


Figure 2.4: Lifetime of a LoRaBug powered by two AA batteries at various operating points based on a measured energy profile.

an acknowledgment and then goes to low-power sleep mode. Both our MCU and LoRa radio have multiple sleep and function states that consume varying amounts of power. The radio transmission consumes the highest amount of energy (= area under the curve × voltage) by a large margin. Thus, any optimization to battery life must focus on reducing the energy of transmissions. Radio reception is the second largest power sink and indicates that any operations requiring long receive windows (e.g. channel activity estimation) will significantly affect lifetime. Finally, any unnecessarily active peripherals and hardware flaws will increase quiescent power that could adversely affect battery life. We discovered and fixed such a hardware design flaw in the Semtech reference designs for the SX1272.

Two parameters affect the energy consumed by transmissions: (1) transmit power and (2) transmit time. Using the currently available LoRa radio chipsets (Semtech SX1272 and SX1276), we've observed that the transmit power does not significantly change the power drawn from the battery during transmission. However power control is still essential to improve network capacity. Any optimization will thus have to focus on reducing the transmit time. The transmit time is determined by the data rate and the amount of data to send. We do not control the amount of data generated by client devices and thus, improving the

data rate would provide the largest improvements.

Figure 2.4 shows the estimated battery life of a client device equipped with two 2000 mAH AA batteries (assuming 10 year charge life and 60% usable energy) if it were to communicate with different data rates. Wireless systems try to communicate at the highest data rate that does not cause too many errors. In the case of LoRa devices, switching to a slower data rate increases the spreading factor, which have better sensitivity on the receiver. Thus, LoRa devices communicating at the highest spreading factors (and correspondingly using the lowest data rates) can communicate at much longer range and with higher reliability. The downside is a significant increase in their transmission time which severely affects battery life.

This analysis shows that with proper duty-cycling, a LoRa device can function and communicate for multiple years on simple batteries. However, this does not uniformly apply to all LoRa devices in different conditions, particularly if they are located further and must use slower data rates.

2.4.4 Software Architecture

Most LPWAN systems will provide streaming of data to and from deployed LPWAN devices, but a very limited ability to manage these devices. Being an open community-driven network, we want gateways and end-devices on OpenChirp to be independently manageable by their owners. We present a software architecture that exposes an application layer allowing users to register devices and gateways, describe transducer properties, transfer data and retrieve historical values. We also want users to be able to add more functionality to OpenChirp for evolving use-cases. To this end, our architecture allows user-created "services" access to raw data and the permission to convert it to other usable outputs. These services could run on external servers or on OpenChirp as javascript functions.

OpenChirp builds upon LoRaWAN by adding a user management framework, application interface and a set of core services for performing data serialization (converting over-the-air binary data into a typed form with a schema), meta-data management and time series data storage. This is analogous to layers five through seven of the OSI stack (session, presentation and application) with the addition of an Infrastructure-as-a-Service layer on top that gives developers management and monitoring tools. In the purest sense, a LoRaWAN server is responsible for delivering binary blobs to an application while OpenChirp provides structured data with supporting meta-information and services like a web interface and storage. All system configurations are carried out using a REST interface while devices that require more direct access to data like gateways or processing agents communicate using a Publish-Subscribe layer.

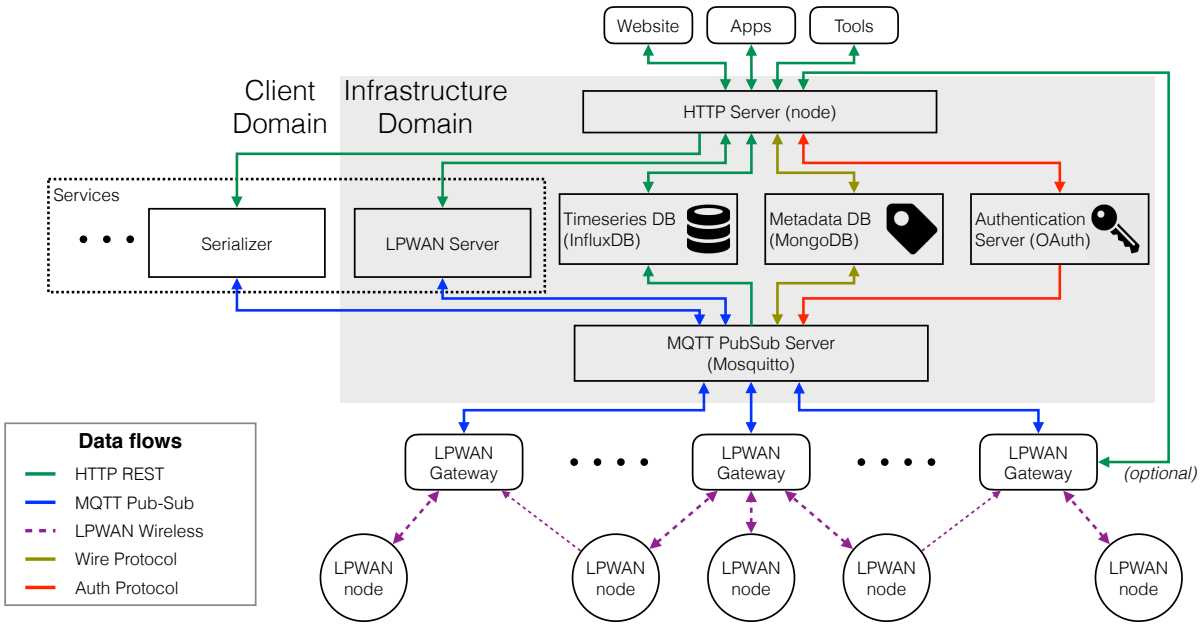


Figure 2.5: System architecture for the OpenChirp network

Application Programming Interface (API)

External devices communicate with the OpenChirp network through two interfaces: (1) HTTP REST and (2) Publish-Subscribe (Pub-Sub). Client websites, mobile applications, management tools, etc. interface through an HTTP REST interface. The REST interface provides easy management of devices and their properties (location, metadata, functionality etc) as well as access to device time-series data. HTTP operations are managed by a server implemented in *node*. Separating the OpenChirp API from the internal implementation of various services helps us create a modular architecture that also allows us to experiment with various components of the infrastructure.

Publish-Subscribe Dataflows

Heavily-constrained end-nodes with sensors are the primary producers of information in most IoT deployments. Publish-Subscribe (Pub-Sub) architectures help decouple the producers and consumers of information in terms of timing and availability. We often see multiple consumers subscribe to the same produced data. Finally, relatively resource-heavy operations like access control are managed by more capable machines in the infrastructure rather than in the end-nodes. LoRaWAN gateways communicate with OpenChirp using MQTT Pub-Sub flows. Various internal dataflows (e.g. between databases and services) are also implemented using Pub-Sub. The MQTT protocol is implemented using Mosquitto [49].

Built-in and Extensible Services

In the OpenChirp infrastructure, *services* provide additional features through server hosted software modules. We provide a framework that services can use if they wish to be notified about new devices, subscribe to and process their data feeds. Following are some examples of provided services.

- **Data Serialization and Deserialization:** Given a schema, this service converts binary blobs of serialized data into structured data and vice versa. This service is helpful to get usable outputs from uplinked data or to send downlink messages with multiple fields.
- **LoRaWAN Server:** The LoRaWAN server [14] is responsible for processing, decrypting and managing LoRa communications in the OpenChirp network. It is responsible for MAC decisions like selection of the best downlink gateway, data rates and power levels for messages. It is structured as a service for easy testing and upgradability.
- **Timeseries and Meta-Data Storage:** The timeseries database (implemented using InfluxDB) stores all data to and from end-devices so that users can access it at a later time without having to keep their own storage server active. Some end-node and gateway properties like location, device type, capabilities, sensor sampling rates, closest gateway, etc. provide meaning and context to their data are are stored in the a MongoDB database.

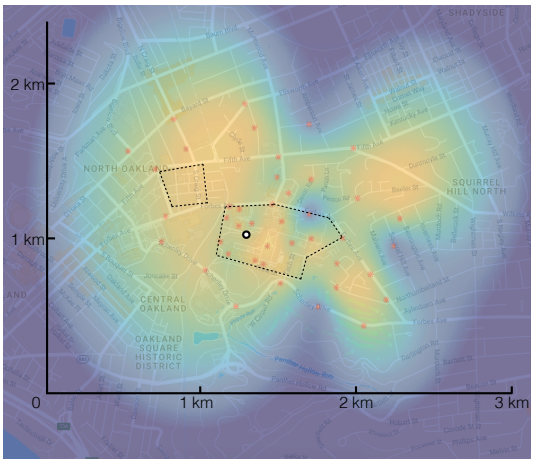
2.4.5 Network Performance

In this section, we evaluate the network performance of the OpenChirp network around the Carnegie Mellon University campus. We look at network coverage both outsides and inside buildings.

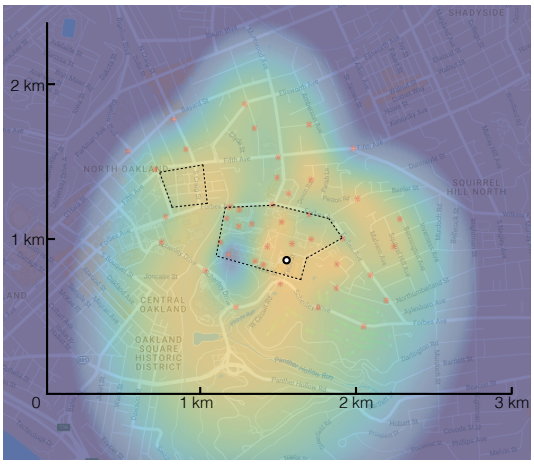
A major objective of the OpenChirp network is to be able to serve CMU's campus with a small constellation of gateways. This requires coverage of the complete geographical area as well as signal penetration inside buildings. After installing only four gateways on the roof of campus buildings, we perform a set of coverage tests. The tests are performed with a LoRa end-node configured for uplink communications with 125 kHz channel bandwidth, data rate of 980 bits/s, spreading factor of 10 and coding rate of 4/5. Downlink communications from the gateway used 500 kHz channels at 3900 bits/s.

Figure 2.6 shows the coverage heat-map based on the average received signal strength indicator (RSSI) of ~ 12 messages sent from each location. Though some regions may not be covered using one gateway (due to shadowing, attenuation, etc.), a combination of four gateways can successfully cover most campus areas.

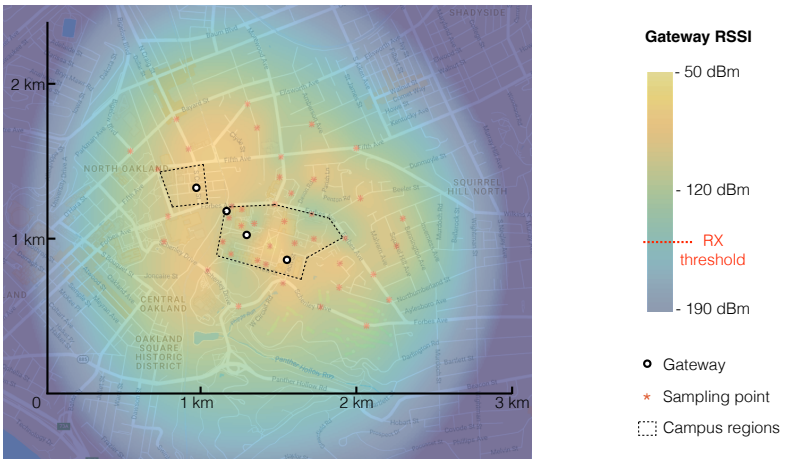
Figure 2.7 shows the signal penetration across multiple floors of Wean Hall, a large 250,000 sq.ft. 9-story poured concrete building with a single gateway located on the roof. The packet success rate on the left



(a) Wean Hall gateway



(b) GSIA gateway



(c) All gateways combined

Figure 2.6: Network coverage in and around CMU campus. (left) Wean Hall gateway, (middle) GSIA gateway and (right) combination of all gateways.

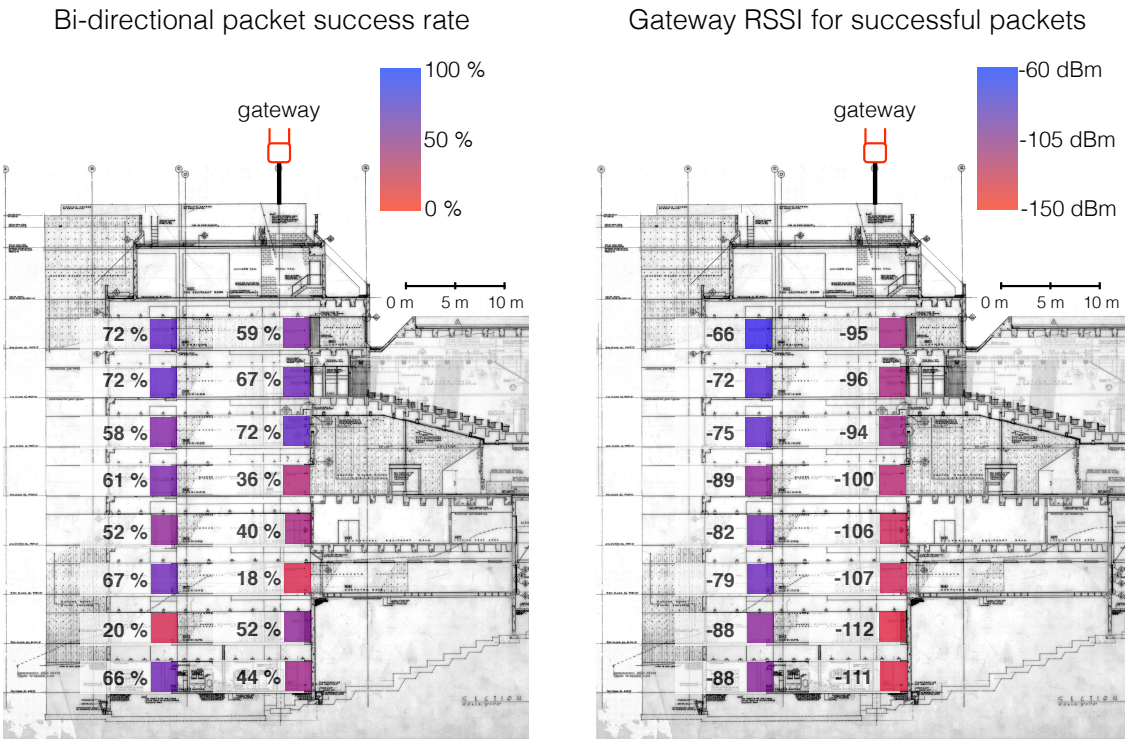


Figure 2.7: RF signal penetration experiments performed in a large poured-concrete building on campus. (left) shows the success rate for bi-directional packet exchange between end-node and gateway and (right) shows the RSSI at the gateway for successful transfers.

is computed based on the number of complete bi-directional transfers ($\sim$ 60 points in each left corridor, $\sim$ 15 points in each right corridor). The image on the right shows the gateway RSSI of successful transfers. Though we get fair coverage inside buildings, it is not reliable everywhere, particularly in deeper sections. We also noticed that network performance was better in areas near the elevator shaft compared to other areas at the same level. This hints at multipath reflection helping improve coverage rather than transmission through construction materials.

We have used the data collected during these coverage tests to calibrate a log-distance path-loss model with parameters $PL_0 = 105.5729$ dBm for $d_0 = 40.0$ m, $\gamma = 2.1495$ and flat fading $\sigma^2 = 100.0724$.

$$PL = PL_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + \mathcal{N}\left(0, \sigma^2\right)$$

2.5 Scalability of LoRaWAN

In this section, we theoretically analyze the scalability of LoRaWAN. We will attempt to find the ideal geographic distribution of devices around a gateway, the number of devices a single gateway can support and the effect of LoRaWAN transmissions on other ISM-band communications.

2.5.1 Distribution of Devices around a Single Gateway

A typical LoRaWAN gateway can simultaneously receive from multiple neighboring frequency channels and multiple coding schemes (represented by spreading factors (SF)). Transmit power control must be properly implemented on all devices on the network to successfully receive multiple transmissions simultaneously. To better understand the scalability of such a network supported by many such gateways, let us first analytically derive the distribution of devices a single such gateway could support.

We will focus on LoRaWAN Class-A (pure ALOHA) as it is the simplest protocol supported by most LoRaWANs and has the least requirements from devices. Let us assume a log-distance path loss model for radio propagation and limit our deployment to a flat region. Let us also assume that all devices are sending frames with equal sized payloads and they are sending these at the same average rate. Though these assumptions are simplistic, they will provide a good intuition for practical scenarios.

Every increase in SF increases the frame time (T_{sf}) for a given payload size. Though the exact scaling varies for each SF, due to the discretized distribution of bits across symbols, we will make a close approximation and say that increasing the SF by 1 increases the frame time by a factor of α. We can make a pessimistic estimate by considering $\alpha = \min_{sf}(T_{sf+1}/T_{sf}) \approx 1.8$ for a 20 byte LoRa packet.

A LoRaWAN gateway can receive multiple transmission in the same frequency band as long as they each have different SFs. Let us say that a single frequency band at a given SF can support n_{sf} devices (n_{sf} may be the optimal number for maximum throughput on ALOHA or the number that can saturate a planned TDMA system). Since the SF affects frame size,

$$n_{sf+1} = \frac{n_{sf}}{\alpha}$$

If a gateway can support k different SFs ranging from sf to $sf + k - 1$, the total number of supported devices will be

$$\begin{aligned} n_{\text{total}} &= n_{sf} + n_{sf+1} + \ldots + n_{sf+k-1} \\ &= n_{sf} + \frac{n_{sf}}{\alpha} + \ldots + \frac{n_{sf}}{\alpha^{(k-1)}} \\ &= n_{sf}\left(1 + \frac{1}{\alpha} + \ldots + \frac{1}{\alpha^{(k-1)}}\right) \\ &= n_{sf}\left(\frac{1-(1/\alpha)^k}{1-1/\alpha}\right) \\ &= \left(\frac{n_{sf}}{1-1/\alpha}\right)\left(1-\alpha^{-k}\right) \end{aligned} \tag{2.1}$$

Next let's try to express this the total number of devices as function of max range. Changing the SF by 1 improves the sensitivity of the receiver (S_{sf}) by 2.5 dB [18] (we will call this ΔS), which results in the signal being received successfully at a longer distance. Say d_{sf} is the furthest mean distance at which a packet using sf could be received.

$$PL = P_{TX} - S_{sf} = PL_0 + 10\gamma \log_{10}\left(\frac{d_{sf}}{d_0}\right) + \mathcal{N}$$
$$\therefore P_{TX} - S_{sf+i} = P_{TX} - S_{sf} + i\Delta S = PL_0 + 10\gamma \log_{10}\left(\frac{d_{sf+i}}{d_0}\right) + \mathcal{N}$$

If we ignore the noise variance of the log-distance model we get the following expression on subtracting the above equations.

$$i\Delta S = 10\gamma \log_{10}\left(\frac{d_{sf+i}}{d_{sf}}\right) \tag{2.2}$$

k represents the count of different SFs in the system, while i is the separation between the largest and smallest SFs in the system. Thus, $k = i + 1$. We can now combine equations 2.1 and 2.2.

$$\begin{aligned}
n_{\text{total}} &= \left(\frac{n_{sf}}{1 - 1/\alpha}\right)\left(1 - \alpha^{-(i+1)}\right) \\
&= \left(\frac{n_{sf}}{1 - 1/\alpha}\right)\left(1 - \alpha^{-1 - \frac{10\gamma}{\Delta S}\log_{10}\left(\frac{d_{sf+i}}{d_{sf}}\right)}\right) \\
&= \left(\frac{n_{sf}}{1 - 1/\alpha}\right)\left(1 - \alpha^{-\frac{10\gamma}{\Delta S}\log_{10}\left(10^{\Delta S/10\gamma} \times \frac{d_{sf+i}}{d_{sf}}\right)}\right) \\
&= \left(\frac{n_{sf}}{1 - 1/\alpha}\right)\left(1 - \left(10^{-\Delta S/10\gamma} \times \frac{d_{sf}}{d_{sf+i}}\right)^{\frac{10\gamma}{\Delta S}\log_{10}\alpha}\right)
\end{aligned}$$

This expression is only valid at integer values of i. However we note that a continuous Pareto distribution with scale $x_m = 10^{-\Delta S/10\gamma} \times d_{sf}$ and shape $a = (10\gamma/\Delta S)\log_{10}\alpha$ has a CDF that matches the above expression when $x = d_{sf+i}$. We can thus model the distribution of our devices at any distance d as a Pareto distribution with the above parameters. Based on the Pareto principle, an individual LPWAN gateway can support a large number of devices that are close by, but it can only support a few devices further away. Figure 2.8 illustrates this concept by estimating the area density of devices that would saturate a LoRaWAN network.

We can extend this analysis to multiple gateways by first partitioning all our regions into areas closest to each gateway and then performing a similar calculation for each gateway individually.

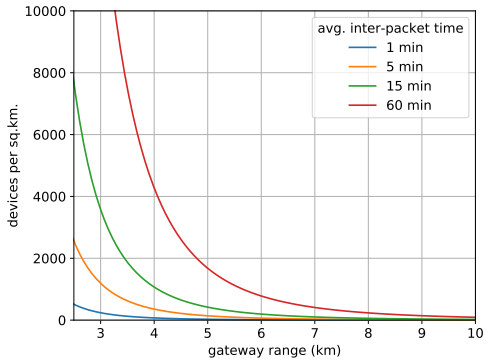


Figure 2.8: Geographic distribution of devices estimated using the Pareto distribution.

This analysis indicates that in a realistic setting capacity issues may arise if the geographic distribution of gateways does not approximately mirror the geographic distribution of devices, particularly for congested networks. Areas with more devices must have a larger number of gateways in close proximity that can communicate at faster data rates.

2.5.2 The Number of Supported Devices

In this section, we will analytically find n_{sf} which is the number of devices that can be "supported" at a given spreading factor. The meaning of support can be vague so we will define it as the number that achieves maximum throughput in an ALOHA access scheme. As before, we'll assume all devices implement satisfactory power control to not overwhelm the radio frontend on the gateway. A gateway can support many independent frequency channels (c) and these channels don't have any overlap so they don't interfere with each other. Let T_{sf} be the frame time of messages using a spreading factor sf.

We know that the throughput of ALOHA is maximized when there are $G = 0.5$ frame transmissions per frame time. Thus, if we are supporting n_c devices on a given frequency channel, each transmitting at a rate of r transmissions per second. The total number of transmissions per second is given by

$$\begin{aligned} \sum_{i=1}^{n_c} r_i &= \frac{G}{T_{sf}} \\ \implies n_c &= \frac{G}{T_{sf} r} \\ \therefore n_{sf} &= c \times n_c = \frac{Gc}{T_{sf} r} \end{aligned}$$

For maximum throughput, we would set $G = 0.5$. We can also find other operating points either by setting a success probability $P_{success} = \exp(-2G)$ or by fixing the desired throughput in units of successful frames per frame time, given by $S = G\exp(-2G)$.

Inter-packet time	SF7	SF8	SF9	SF10	SF11	SF12	Total
1 sec	71	39	22	11	6	3	152
10 sec	707	389	216	108	61	30	1511
1 min	4242	2332	1295	647	364	182	9062
5 min	21210	11660	6474	3237	1820	910	45311
15 min	63631	34981	19423	9712	5459	2730	135936
60 min	254525	139925	77693	38847	21836	10918	543744

Table 2.9: Estimated number of devices that maximize the throughput on LoRaWAN under ideal conditions

Table 2.9 shows an estimate of the number of devices that would saturate a LoRaWAN network (i.e. reach optimal ALOHA throughput). This assumes an ideal scenario for Class-A operation: no acknowledgements or retransmission, no other interfering devices, 125 kHz bandwidth per channel and 8 channels being used with devices frequency hopping between them. Note that each packet only has a $1/e = 36.8\%$ chance of being successful at this operating point.

Transmit power control is an important requirement for the scalability of LoRaWAN networks. If a signal is received at a power that is much higher than the other frames ($\approx 8dB$ for LoRaWAN devices [13]), it results into the failure of receptions with other spreading factors at the same frequency. A typical gateway services many frequency bands using a single radio frontend. If the power is even higher, it will saturate the radio frontend completely and we lose all receptions at all frequencies serviced by that frontend. Effectively, the gateways will lose almost all concurrent receptions during the presence of a strong transmission.

Effective power control could become problematic if we have a scenario with multiple LoRaWAN networks in the same area. Say a device is communicating with gateway A, but is located far away and hence, needs to use high transmit power and slow data rates (i.e. it's frames are very long). If it were to be located close to gateway B that is servicing a different network, the device transmissions would overwhelm gateway B with its high transmit power for the (relatively long) period of its transmission. To avoid these pitfalls, different networks located in the same space must collaborate in some way to either divide up the frequency spectrum or agree to share access to gateways.

2.5.3 Interference to Other Wireless Communication

In [63], the authors investigate the interference between LoRa and IEEE 802.15.4g, another commonly used ISM band communication protocol. They determine that LoRa transmissions, even at higher data rates are coded heavily enough that inter-technology interference of similar power levels does not affect LoRa packet reception. On the other hand, LoRa transmission, being much longer, adversely affect 802.15.4g communications. In this section we will try to understand how the load on a LoRa network can affect the communication of a different technology also relying on ALOHA-like random access.

Since we will be working with communication protocols using different frame times, we will make a simplifying assumption that a long transmission lasting T_{long} is on an average equivalent to $k = T_{long}/T_{short}$ transmissions of time T_{short}. To determine the probability of success, we must determine the the equivalent number of transmission attempts made by the LoRa network. Let T be the frame time of the shorter 802.15.4g packet and T_{sf} be the frame time of the longer LoRa packets with spreading factor sf. Thus, $k_{sf} = T_{sf}/T$. Say there are n_{sf} devices using sf each sending r_{sf} frames per second. Additionally, say that c LoRa frequency channels overlap with a given 802.15.4 channel.

For each sf, there are a total of $cn_{sf}r_{sf}$ transmissions per second that can interfere with our 802.15.4g frame. However, these transmissions last for a much longer time of T_{sf}. This can instead be interpreted as $ck_{sf}n_{sf}r_{sf}$ transmissions per second of time T each. A simpler modulation scheme like the one used by 802.15.4g will suffer from interference due to transmissions in all the used SFs. The number of transmissions per frame time T is given by

$$G = T \sum_{sf} ck_{sf}n_{sf}r_{sf}$$

The probability of a 802.15.4g frame being received successfully is thus given by

$$P_{success} = \exp(-2G)$$

Once we populate the above expression with parameters for a busy LoRaWAN network, we find that the probability of success is very low. This is because LoRa transmissions occupy significant air time. As an example, interference from a LoRaWAN network running at only one tenth of it's saturated capacity would result into a 99.2% failure probability for the much shorter 802.15.4g packets. We can safely argue that the ALOHA random access used by LoRaWAN can be troublesome for other ISM band communication, particularly if the LoRa network is heavily loaded. It would be beneficial to switch towards alternative MAC strategies (like LoRaWAN Class-B which is inspired by TDMA) to avoid wasted transmissions due to collisions. However, we must be mindful of the additional cost, particularly in terms of energy, introduced by these MAC protocols.

2.6 Summary

In this chapter, we explore the paradigm of low-power wide-area networking. We believe open LPWAN networks have the potential to unlock a plethora of creative ideas that are currently either power or cost

limited by existing wireless technology. We also presented the design and performance of OpenChirp, a LoRaWAN network that is deployed around the Carnegie Mellon University campus. OpenChirp enables collaborative and communal wide-area networking for telemetry. This deployment demonstrated that a few well positioned gateways can a large region and that low-cost nodes can be designed and deployed to run on batteries for many years. Finally, an analytic analysis of the scalability of LoRaWAN shows that gateway deployments must mirror the distribution of devices and that interference from class A LoRa transmissions could pose interference challenges to other ISM band communications.

Chapter 3

Distributed reception of LPWAN transmissions

Despite the expected rise in density of LPWAN gateways, not all LPWAN devices across the same network will have the same long 10-year lifetimes. Devices located in urban spaces deep inside buildings or in remote neighborhoods will experience severe drain in battery, as their signals are highly attenuated even at the closest base station. Some of these devices, such as those in basements or tunnels, may not be in communication range of any gateway at all. Unlike cellular networks, LPWANs are largely user-deployed and unplanned, meaning that these devices may remain battery deprived or simply out of network reach in perpetuity, even as thousands of gateways proliferate city-wide.

Most wireless systems, such as cellular networks and WiFi, use a single device to basestation/access point link to communicate. Some LPWANs, like LoRaWAN, support a simple form of diversity by allowing multiple gateways to independently receive the same uplink transmissions, which are then forwarded to a server that deals with duplicates. However, this method does not involve any sharing of information between the gateways. Distributed reception would mean gateways share information that is richer than the just the final decoded data to enable a functionality they could not achieve independently.

This chapter presents Charm, a system that enhances the coverage of LPWANs and the battery life of client devices in large urban deployments through distributed reception of LPWAN transmissions. Charm exploits the observation that while signals from certain clients may attenuate significantly, they are still likely to be received by multiple gateways in a dense network. Charm introduces a hardware and software design at the gateways that identifies and transports weak received signals to the cloud. We then develop a joint decoding system at the cloud that coherently combines weak signals received across multiple city gateways to decode the underlying data. As a result, Charm both expands the decoding range of the

LPWAN network and improves data rates for nodes already in range, allowing client devices to spend less energy per transmitted bit. Charm is built on the LoRaWAN platform [53], described previously in Section 2.4.1. Charm is implemented in a first-of-its-kind pilot deployment for coherent diversity combining and demonstrates increased network coverage and improved data rates across client devices.

While coherent diversity combining and PHY-layer processing in the cloud has received much attention in the WiFi [91, 96] and cellular [16, 95] context, designing such a system for low-power WANs offers radically new challenges. At the gateways, we would have to decode very weak signals, weaker than 30 dB below the noise floor. Simply uploading all received data to the cloud would overwhelm the back-end link, which is often a simple home LAN. Both the LPWAN gateways and clients are designed to be economical and deployed at scale, and without the time synchronization required for coherent combining. At the cloud, collating receptions from a large number of gateways at city-scale to identify which of them contain packets from the same client is a challenge.

3.1 Contributions

We make the following novel contributions through the development of Charm for distributed reception:

- A technique that leverages the geographical diversity of unplanned, user-deployed gateways to enable joint decoding of weak transmissions. This improves battery-life for users in the network and increases the coverage area.
- A hardware platform and the underlying algorithms for detecting weak LoRaWAN transmissions locally at the gateway.
- A software architecture that builds on top of LoRaWAN to enable joint-decoding of signals in a scalable manner.

3.2 Charm's Approach to Distributed Reception

We provide an overview of our approach to address each of the above challenges.

3.2.1 Noise-Resilience at the Gateway

The key challenge at the gateway is identifying packets that are significantly below the noise floor and, therefore, virtually undetectable. A straw-man approach to this problem would be to correlate the received signal with a known preamble in any valid packet. For instance, LoRaWAN uses a sequence of identical

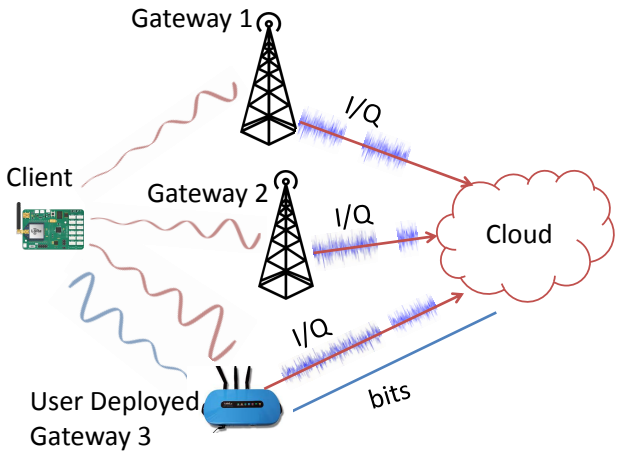


Figure 3.1: Charm performs joint decoding for LPWANs in the cloud

chirps – signals whose frequency increases linearly in time – as a signature that is prefixed in every packet. In principle, sending an extremely long preamble could provide high resilience to noise. In practice, doing so goes against the spirit of LPWANs, where energy for transmission is a valuable resource for the client.

Charm's approach to resolving this challenge is a hardware and software gateway design that leverages the structure of the LoRaWAN LPWAN protocol. Specifically, we develop a transform that converts the data symbols containing *a priori* unknown bits into a repeated and known sequence of signals, much like the preamble. Charm can therefore now use both the preamble and the modified data sequence to detect any packet.

To understand our approach at a high-level, we present an illustrative example that dives into the details of the LoRaWAN PHY-layer. LoRaWAN transmits data symbols as chirps whose initial frequency is a function of the data. For instance, over a bandwidth of 100 Hz, LoRa could represent the bit "0" as a chirp starting at 2 Hz and bit "1" as a chirp starting at 52 Hz. Charm's filter aliases the received LoRa signal so that frequencies modulo 50 Hz fold into each other. This means that both bit "0" and bit "1" now map to an identical chirp starting at 2 Hz. We apply this filter through the received packet to obtain a repeated sequence of chirps as long as the entire packet itself. This technique allows us to detect the packet with a much higher resilience to noise compared to using the preamble alone, without incurring additional overhead.

We develop a custom gateway hardware platform integrating a Semtech LoRaWAN radio front-end, a low-power FPGA and a Raspberry PI that can filter and detect weak signals by processing received raw I/Q samples in real-time. Our hardware platform, a hybrid between a full SDR and a dedicated high-performance radio, is designed to be open and highly programmable – a novel tool to experiment with alternative LPWAN PHY-layer designs in the 900 MHz ISM band, without compromising on signal quality or real-time performance.

3.2.2 Scalability at the Cloud

At the cloud, Charm must deal with a large number of receptions from various gateways in a city, pruning for weak signals and identifying common signals between gateways. Charm proposes multiple optimizations to run its algorithms seamlessly at city-scale. For instance, it is often the case that gateways transmit weak signals to the cloud for packets that have already been decoded perfectly at other gateways. However, realizing that the weak signal has already been decoded elsewhere is impossible without decoding it in the cloud in the first place. Charm resolves this chicken-or-egg dilemma by exploiting the timing and geographical location of the received signal. Prior to sending any signal data to the cloud, a Charm gateway sends the location, frequency, accurate timing and signal-to-noise ratio (SNR) of the received weak packet. The cloud collates such information across multiple gateways and requests for signals only from the gateways that receive these signals the best. In doing so, Charm saves valuable uplink bandwidth at the gateways and computation at the cloud. We describe how Charm mitigates range of other important challenges at the cloud such as imperfect timing, frequency offsets and overlapping transmissions.

3.3 Related Work

3.3.1 Distributed MIMO and Coherent Combining

A large body of work has proposed the use of multiple-antennas (MIMO) to improve SNR and reduce interference [96, 51, 46]. In the WiFi context, past systems have used multi-user MIMO to improve performance on the uplink [85, 91, 96]. In the cellular context, massive MIMO proposals have demonstrated scaling gains of towers with a large number of antennas [86, 48]. There has been much theoretical work on distributed MIMO overall in both the sensor networking context [24] and wireless LANs [26] and cellular networks [80]. A recent system, Choir [27], has demonstrated improving range and scalability of LPWANs through collaborations of weak client radios. In contrast, this work seeks to use collaboration between gateways without any modifications to client behavior whatsoever, to improve the battery life of even a single client. More recently, practical distributed MIMO systems, primarily in the LAN-context, have demonstrated both multiplexing and diversity gains [40, 97, 70]. Instead, our approach brings the diversity gains of distributed MIMO on the uplink to LPWANs. In doing so we overcome multiple challenges owing to the fact that signals at any individual tower are well below the noise floor and are captured by low-cost hardware that lacks the precise time synchronization required for coherent combining.

3.3.2 Cloud Radio Access Networks (Cloud-RAN)

Multiple research efforts from industry and academia have advocated the use of PHY layer processing at the cloud as opposed to the base stations [77, 39]. In the cellular context, CloudRAN aims to perform baseband processing at the cloud, allowing base stations to be simple and easy to deploy [16, 95]. The key challenge, however, is the need for a reliable fiber optic backhaul to the cloud to collate data streams in a low latency manner, motivating the need for cost-effective high-performance backhauls [52, 17]. Our approach aims to bring PHY processing in the cloud to LPWANs that operate at significantly lower bandwidth, with loose latency bounds, and can therefore afford Ethernet backhauls. We perform a wide variety of optimizations to minimize the use of uplink bandwidth, including local packet detection and data compression using an FPGA accelerator. These are helpful when the gateways are user-deployed with residential internet backbones.

3.4 Coherent Combining

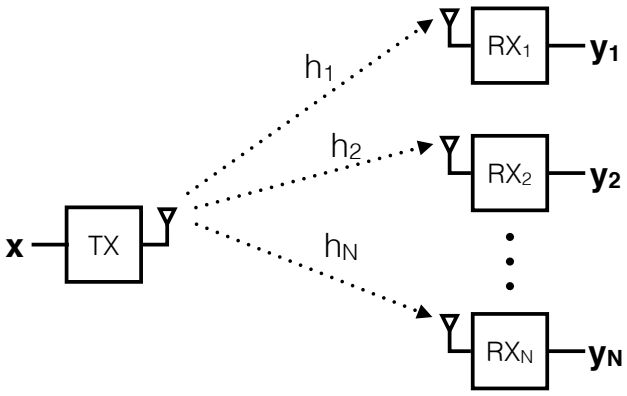


Figure 3.2: Coherent combining helps receivers collaboratively improve signal-to-noise ratio

Wireless radios leverage multiple antennas (MIMO or multiple-input multiple-output) to improve throughput. This work considers coherent combining where transmissions from a single-antenna transmitter (e.g. an LPWAN client) are heard by multiple receiver antennas (e.g. LPWAN gateways). These gateways can then coherently combine the received signals to improve signal decoding.

Mathematically, let the transmitted signal be x and each of the gateways receive a signal y_i through wireless channel h_i, introducing an independent noise n_i at the receivers. For a narrow-band system (as are LoRaWAN and most LPWAN technologies), we can write the received signal as: $y_i = h_i x_i + n_i$.

The receivers can now coherently combine their received signals by using the known wireless channels h_i:

$$y_{\text{combined}} = \sum_{i=1}^{N} h_i^* y_i = \sum_{i=1}^{N} |h_i|^2 x + \sum_{i=1}^{N} h_i^* n_i$$

The first term is the combined signal, while the second term is the combined noise. However, while the signals add up coherently, the noise, being independent, adds up incoherently. This results in an overall increase in the combined SNR, which allows us to jointly decode a packet that may otherwise not be decodable by any individual receiver.

$$SNR_{\text{combined}} = \frac{\left|\sum_{i=1}^{N} |h_i|^2 x\right|^2}{\sum_{i=1}^{N} \left|h_i^* n_i\right|^2} \geq \frac{\left||h_i|^2 x\right|^2}{\left|h_i^* n_i\right|^2} = SNR_i$$

In practice, performing coherent combining as shown above makes two important assumptions: (1) the packets can be detected at individual receivers above some SNR threshold, and (2) receivers share a common clock reference for time and frequency. This chapter describes the challenges in implementing coherent combining in the low-power wide-area context where neither assumption holds.

3.4.1 Working with the LoRa Physical and MAC Layer

LoRa and LoRaWAN were already introduced in Section 2.4.1. However, let's look further into some of their properties which further motivate and enable the development of Charm.

LoRaWAN packets consist of a preamble of upchirps, followed by a synchronization (SYNC) header with downchirps and finally data. All LoRa symbols are created using linear frequency modulated waveforms offset by known frequencies. A number of parameters, like the sensitivity of a receiver and the data rate, are determined by the spreading factor. Changing the spreading factor lets us trade off data rate (and thus, transmit time) against resilience to noise. LoRaWAN radios are therefore designed to transmit at the lowest possible spreading factor that can be received at existing noise levels for minimizing transmission time and the resulting battery drain. This work therefore strives to reduce spreading factor (thereby improving data rate) for weak transmitters.

LoRaWAN networks are designed to be simple star-topologies that have client devices directly communicating with a gateway that is connected to the internet over ethernet or cellular links. Thus, we only have to make changes on the gateways to enable distributed reception without touching the deployed client devices. Due to the MAC-in-the-cloud design, all major network decisions are made by the LPWAN server and not the gateways themselves. A distributed reception system using multiple gateways can then simply masquerade as another gateway without having to change anything else in the LoRaWAN protocol. LoRaWAN allows and encourages its users to deploy their own gateways. These gateways are completely

unplanned and on low-bandwidth, unreliable internet connections (compared to cellular base-stations that are extensively planned and have dedicated optic fiber connections). We refer to these as user-deployed gateways. The penultimate goal of this work is to make individual unreliable user-deployed gateways more valuable by pooling together PHY-layer processing at the cloud.

3.5 Charm's Architecture

The goal of Charm is to decode weak transmissions, which cannot be decoded by any individual gateway, by collating receptions from multiple gateways at the cloud. At one level, this enables us to expand network coverage area, reaching clients deep inside buildings, underground or in outer reaches of the city. More fundamentally, it saves energy on the vast majority of client devices, even if they are within range of some gateways, by allowing them to increase their data rate without experiencing any loss in performance. Our results in Section 2.4.3 demonstrate that lowering transmit time results in a direct and significant impact on battery life.

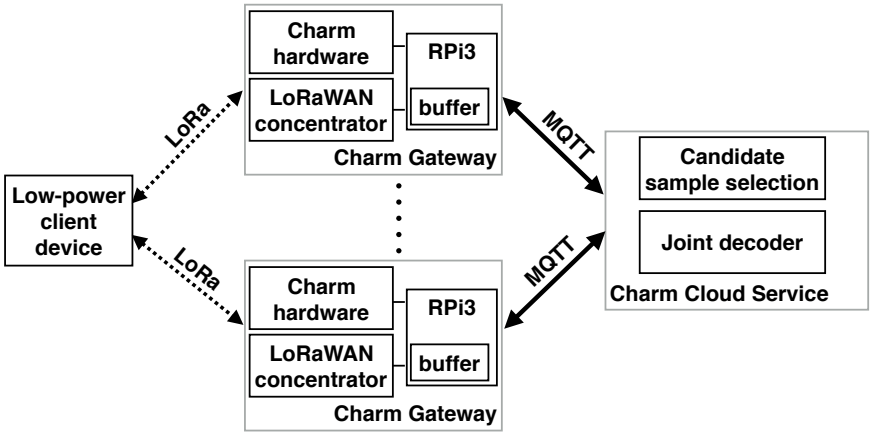


Figure 3.3: Hardware and software architecture to enable Charm

Figure 3.3 depicts Charm's architecture where we assume the gateways can be user-deployed both indoors and outdoors, at a cost of a few hundred dollars. These base stations have an Ethernet backhaul to the cloud that accommodates a maximum uplink bandwidth of a few megabits per seconds. Much like the standard LoRaWAN architecture, MAC-layer scheduling is performed at the cloud with gateways relaying their received data to the cloud. However, to accommodate decoding weak received signals, we also allow gateways to ship raw received I/Q signals from feeble low-power clients to the cloud. The cloud aggregates such weak signals and coherently combines them to decode the underlying data bits from feeble receptions across multiple gateways. In other words, Charm performs a joint optimization of the PHY-layer at the

cloud, simultaneously improving battery life and range of low-power clients at the expense of increased computation at the cloud.

Realizing a scalable and real-time system based on the above architecture is challenging both at the gateways and the cloud:

- **At the Gateway:** Given that signals from weak LPWAN clients are often well below the noise floor, gateways are unaware of these packets in the received signal. This means that base stations must effectively send all their received raw signal data to the cloud to detect and decode weak signals, stressing their limited uplink bandwidth.
- **At the Cloud:** The cloud must identify signals from which gateways need to be combined to recover transmitted data from multiple clients. At city-scale, it is conceivable that overlapping weak transmissions from different clients are received at the same time by gateways, making data recovery challenging at the cloud. Additionally, due to the use of low-cost hardware that lacks precise time synchronization, each of the gateways adds clock and frequency errors to the captured signals. These must be resolved before the signals can be combined.

The rest of this chapter describes Charm's solutions to each of these challenges. Specifically, Charm makes two key contributions: (1) A software interface at the gateway to identify weak transmissions to ship to the cloud, and a hardware design that facilitates these decisions in real-time; (2) A scalable cloud-based PHY-layer processing system at the cloud that can operate at city-scale. Next we elaborate on each of these components.

3.6 The Charm Gateway

We first describe Charm's design at the gateway to enable accurate decoding of weak clients, by relaying suspected weak signals to the cloud. Charm achieves this first through a software algorithm at the gateway that identifies weak transmissions that may be significantly below the noise floor. We further implement this approach in hardware by building a custom programmable radio platform for the gateway, which streams and processes raw I/Q samples using an FPGA. We show how a Charm-gateway can detect weak signals in real-time through this design, while simultaneously being programmable and responding to policy changes from the cloud.

3.6.1 Locally Detecting Weak Signals

To reap the benefits of coherent diversity combining across multiple gateways, Charm must relay weak signals to the cloud. Yet, uploading all received signals to overcome this problem is unfeasible given that gateways have limited uplink bandwidth to the cloud. To put this in perspective, streaming all received I/Q samples to the cloud requires an uplink bandwidth of 72 Mbps. However, the vast majority of LPWAN gateways are likely to be user-deployed hardware such as set-top boxes that cannot afford this bandwidth. Indeed, this creates trade-off between detecting weak transmitters and conserving uplink bandwidth.

Charm breaks this trade-off by detecting weak signals well below the noise floor at a single LoRaWAN gateway. At a high level, our solution relies on the structure of the LoRa protocol. Specifically, LoRa transmits signals in the form of chirps, i.e. signals whose frequencies increase linearly in time. In addition, several of these chirps are identical. For instance, consider the initial preamble in LoRaWAN with as many as 16 identical and consecutive chirps. This means one can design a receiver that coherently sums up adjacent symbols of any received signal over a sliding window. If the summing-up operation is truly coherent, the underlying signal (i.e. the chirp) will add up constructively, while noise will add up incoherently. In effect, this boosts the signal-to-noise ratio of the received signal significantly, allowing us to detect at least the preamble of a LoRaWAN packet. One can then deliver a long chunk of packets surrounding this preamble to the cloud.

However, the resolution of the above approach is a function of preamble length – the longer the preamble sequence is, the greater will be the extent of the noise that Charm can tolerate. Transmitting extremely long preambles increases the overhead of the communication system, and in the long term, impacts battery life. Charm therefore develops an approach that can detect weak signals by leveraging data symbols in addition to the preamble – even though the transmitted data sequence is unknown a priori at the gateway. We detail our approach below.

Leveraging the Structure of LoRaWAN Data: Charm seeks to use the structure of the data symbols in LoRaWAN to improve detection of the packet in the presence of noise. Indeed, much akin to the preamble, the data symbols of a LoRaWAN packet are also composed of a sequence of chirps. Unlike the preamble, though, LoRaWAN data is composed of a sequence of chirps with different frequency-shifts based on the bits they represent. Assuming that the underlying data in a message is completely unknown and arbitrary, this makes looking for structure within the data challenging.

Charm relies on the fact that while the data does cause shifts in frequencies of chirps within the packet, these shifts are not completely random. In particular, chirps can undergo a discrete number of possible shifts based on the number of bits per chirp. For a spreading factor of SF (i.e. a transmission data rate of SF bits

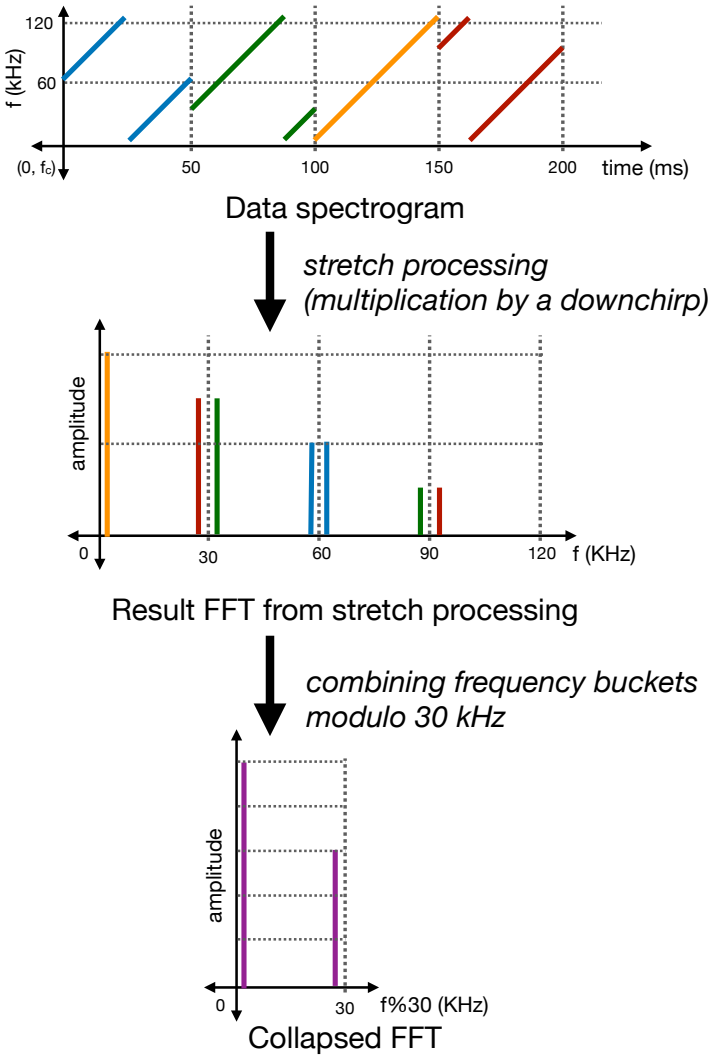


Figure 3.4: The enhanced Charm packet detection process. The chirp signal first is multiplied by a downchirp in the Fourier domain (stretch processing). Windows of the resulting signal are then combined together for threshold detection. This illustration also shows that accurately correcting for frequency offsets will significantly increase the strength of the combined signal.

per chirp), the frequency shift is one of 2^{SF} values. Charm therefore implements a solution that coherently reinforces adjacent chirps, modulo the minimum possible frequency shift between them. This ensures that regardless of their underlying data, adjacent chirps always add up to reinforce each other while noise adds up destructively as before. Given that there is a significantly larger number of data symbols when compared to preamble symbols in any transmission, this provides an additional mechanism to detect packets below the noise.

Mathematically, let $y_1, y_2, \ldots, y_m$ denote the m received data symbols and $x_1, x_2, \ldots, x_m$ denote the transmitted data bits encoded as frequency shifts, each a number between 0 and 2^{SF-1} where SF is the spreading factor. Let $\delta f = \text{Bandwidth}/2^{SF}$ denote the minimum possible frequency separation between two encoded data chirps. Then, we can write the received signal at any time t of the i^{th} symbol as:

$$y_i(t) = he^{j2\pi(f(t)-x_i\delta f)t} + n_1 \tag{3.1}$$

Where $f(t)$ denotes the time varying frequency of the chirp, j is the square root of -1, h represents the wireless channel and n_i represents noise.

When multiplied by $e^{-j2\pi f(t)t}$ and viewed in the Fourier domain, this results in a single tone at frequency $x_i\delta f$ subject to noise. Clearly the location of the tone is a function of the underlying data – a different quantity for different data symbols.

In contrast, let us sub-sample the above equation at times t that are multiples of $1/\delta f$ (let's say $t = \frac{k}{\delta f}$ for integer values of k).

$$y_i\left(t\right) = he^{j2\pi(f(t)-x_i\delta f)\frac{k}{\delta f}} + n_1 = he^{j2\pi f(t)t} + n_1 \tag{3.2}$$

This time, when multiplied by $e^{-j2\pi f(t)t}$ and viewed in the Fourier domain, this results in a single tone at frequency 0 (subject to noise) regardless of the underlying data in each symbol. In other words, sub-sampling in the time domain led to aliasing of all the data peaks in the frequency domain into one frequency bin (in this case, the DC bin), while noise is smeared uniformly across all bins. Indeed, Charm repeats the sub-sampling across multiple time steps separated by $\frac{1}{\delta f}$ and averages the results. The resulting average reinforces peaks corresponding to all the data symbols coherently in one Fourier frequency bin, while noise adds up incoherently among all remaining bins. This leads us to a very natural LoRaWAN packet-detection mechanism that applies this operation across different sliding windows of the received signals. We signal the presence of a packet once our algorithm detects a significant peak in the Fourier domain that dominates other peaks (subject to a threshold). Given that our approach averages results over a large number of data symbols, it remains resilient to noise without making assumptions about the contents of the packet itself.

Algorithm 1: Charm's enhanced detection algorithm

```
for bits in instream do
    [C=I+jQ]=downsample(bits);
    for chirp_length in C do
        F=chirp_length*down_chirp;
        FCollect.collect(F);                                   // Data Collection
    end
    C=FCollect.modulo(δf);                                     // Modulus Bucketing
    if max(abs(fft(C)))/mean(abs(fft(C))) > τ then
        SEND C to CLOUD ;                                      // Packet Forwarding
    end
end
```

Mitigating Frequency Offsets: To add up signals from adjacent symbols coherently, Charm must assume that the received symbols in these signals are identical – subject to noise and discrete shifts in frequency due to the data (as described above). In practice, however, wireless signals from the LPWAN client to the gateway experiences an additional arbitrary shift in frequency due to Carrier Frequency Offset (CFO). CFO stems from the subtle variation in frequency between the clocks on the transmitter and receiver. Given that the client is inexpensive, its clock often exhibits large and arbitrary frequency differences relative to the gateway. Additionally, the CFO for a given transmission received at different gateways is also different and must be resolved individually.

Two properties of CFO make its impact on Charm's algorithm above particularly damaging: (1) CFO, unlike data, introduces a frequency shift that is not discrete, but continuous. As a result, it is not simply eliminated by looking at the chirp in the Fourier domain "modulo δf" akin to the data as described above. (2) CFO introduces a continuous phase shift $2\pi\Delta f_{CFO}t$ onto the received signal that accumulates over time. This means that even otherwise identical received symbols may add up incoherently owing to a time-varying phase shift.

The straw-man approach to eliminating CFO would be an attempt to directly estimate it. For instance, one could rely on the repeated symbols of the preamble where any phase variation is purely a function of CFO. In particular, the phase shift between two otherwise identical preamble symbols separated by t is simply $2\pi\Delta f_{CFO}t$, which one can solve for to estimate Δf_{CFO} and eliminate its effect. However, this solution fails if the number of preamble symbols in the transmitted signal is insufficient to overcome noise. Further, this approach cannot exploit data symbols to estimate CFO, which, as explained earlier, are greater in number and would greatly enhance resilience to noise.

Charm overcomes this problem by realizing that while estimating Δf_{CFO} from the data symbols alone is challenging, it is sufficient to estimate Δf_{CFO} modulo δf to detect the LoRa packet. To see why, recall that

the frequency offset over a packet Δf_{CFO} can be decoupled into two components: $[\frac{\Delta f_{CFO}}{\delta f}]\delta f + \{\frac{\Delta f_{CFO}}{\delta f}\}\delta f$, an integer multiple of δf and the remaining fractional component respectively. When looking at the data chirps in the frequency domain modulo δf, all the data symbols appear identical given that all frequency shifts of the data are multiples of δf. Similarly, the first term of the CFO, $[\frac{\Delta f_{CFO}}{\delta f}]\delta f$, is also an integer multiple of δf and therefore disappears under the modulo. Only the fractional part of the CFO, $\{\frac{\Delta f_{CFO}}{\delta f}\}\delta f$, persists and introduces a time varying phase shift of $2\pi\{\frac{\Delta f_{CFO}}{\delta f}\}\delta f t$ across symbols. This means that we can simply solve for the fractional component of CFO and eliminate its effect akin to the straw-man approach, but using the data symbols in the frequency domain modulo δf. In other words, Charm's solution remains resilient to frequency offset, both in detecting the preamble as well as data symbols of a LoRaWAN packet.

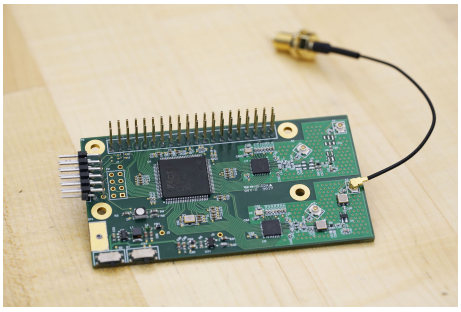
3.6.2 Programmable Hardware Design

Charm must process raw I/Q samples from the gateway and selectively relay this information to the cloud in real-time. However, existing LoRaWAN gateway hardware cannot provide the raw I/Q streams necessary for joint decoding. In contrast, deploying a full software-defined radio (SDR) at the gateway allows packet decoding although it comes with high cost in term of power, sensitivity, unit price and software development. We therefore develop a custom Charm hardware platform shown in Figure 3.5 as an auxiliary peripheral to a gateway and can provide the necessary quadrature streams. Key to our performance is a light-weight, low-cost and easy-to-reprogram hardware accelerator for data reduction enabling further local processing (e.g. on the accelerator or by a Raspberry Pi). In effect, we allow for a system that simultaneously allows some SDR-like programmability of the PHY while maintaining high performance and low cost.

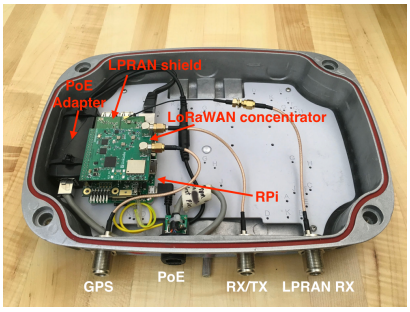
Compressing the Data Stream: The raw IQ stream would be too much for a low-power microprocessor and would also contain too much redundant information for our purpose. In particular, we use the SX1257 RF front-end that provides 1-bit delta-sigma modulated signals at a whopping 36 MSps each for the I and Q streams. In order to keep the data stream to a more microprocessor-friendly load, the design would require some lossless compression. Through careful choice of parameters, we chose to compress the IQ stream by summing consecutive samples in windows of size 64 and convert it into a single 7-bit sample:

$$x_i = \sum_{j=64*i}^{64*i+63} s_j \tag{3.3}$$

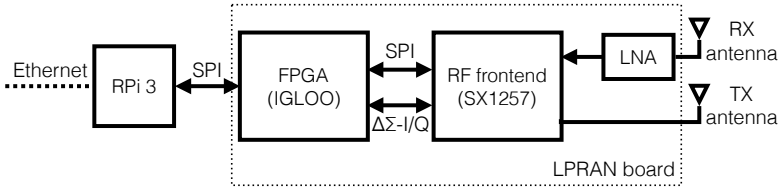
where (x_i) is the analyzable samples and (s_j) is the I/Q sample rates. A window size of 64 is selected since we are only interested in a final bandwidth of approximately 500 kHz that the RF front-end is capable of capturing. Upon applying the above technique, the compressed I/Q streams generate data at a rate of 9 Mbps, down from the original 72 Mbps.



(a) Charm programmable gateway



(b) Gateway with Charm hardware and LoRaWAN concentrator



(c) Block diagram of the Charm programmable radio hardware platform

Figure 3.5: Charm Hardware Platform

Programmability: The delta-sigma I/Q samples are processed locally on a Microsemi IGLOO AGL250 FPGA, which performs the necessary compression for data reduction. The stream of data is transferred using a high-speed serial interface (SPI) to the microprocessor (Raspberry Pi), and forwarded when requested by the joint-decoder for additional processing. Each block of samples is double buffered to ensure the validity of the data during transfers. The microprocessor can then perform additional local processing, time-stamping and temporary local storage until a stream is requested by the joint-decoder. While our hardware platform is not a full-scale SDR, the FPGA allows programmers to implement advanced real-time algorithms for packet decoding and/or full duplex transmission across multiple channels. In addition, the Raspberry Pi allows for ease of programmability when gathering low-rate statistics about the received signals at the gateway. Overall, we believe the Charm hardware platform will reduce the barrier for LPWAN PHY-layer innovation for programmers and researchers across the board.

3.7 Charm in the Cloud

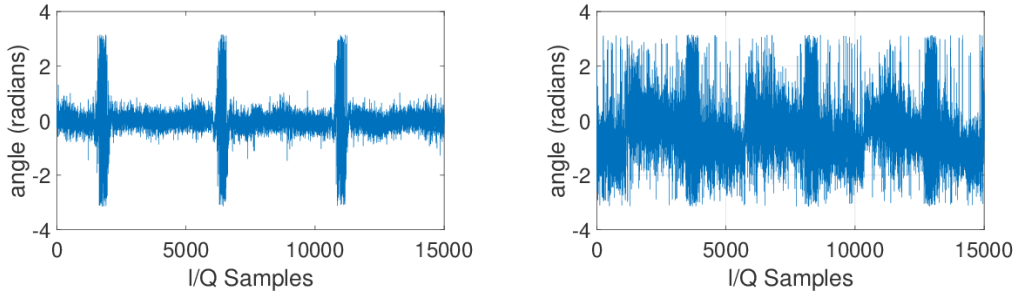
At the cloud, Charm seeks to coherently combine received signals from multiple gateways to recover weak received signals. At a high level, Charm collates I/Q samples from multiple gateways and estimates their packet start time and wireless channel. It then uses standard coherent SIMO combining (see Section 3.4) of the same weak transmission across multiple gateways to ensure that the data can be accurately recovered. Charm repeats this cloud-based PHY-layer processing at city scale across clients and gateways.

The rest of this section describes the key challenges and opportunities in making the above design scalable and practical. First, we describe Charm's approach to ensure accurate time-synchronization between gateways, showing how even an offset of one or two samples can be severely detrimental to coherent combining. Second, we present our solution to dynamically infer signals from which gateways need to be combined over time to best recover a weak signal. Finally, we present opportunities to improve bandwidth and system performance at the cloud by avoiding wasted transmissions of I/Q data to the cloud, as well as wasted computation.

3.7.1 Time Synchronization in the Cloud

Charm relies on the accurate timing of received weak signals at the gateways for two important reasons: First, any offset in timing between signals corresponding to the same packet across gateways will prevent the signals from coherently combining. Second, the precise start time of packets across gateways is valuable information to identify the packet, allowing Charm to infer which received signals across gateways correspond to the same packet.

A naive approach to synchronizing base stations would be to synchronize them through highly accurate clocks (GPS-synced) or through time-synchronization protocols in software over the backbone network (e.g. NTP). In practice, for indoor gateways (e.g. set top boxes) connected to an Ethernet backhaul, these can provide time synchronization of up to a few milliseconds. In practical terms, this means that the received signals at the gateways can be time synchronized to within a small number of time samples.



(a) Without timing offset

(b) With timing offset

Figure 3.6: Effect of timing offset on phase angle of the received signal

Unfortunately, even a small offset in the timing between two gateways can severely deteriorate coherent combining. Figure 3.6 depicts a simple example of the phase difference between two gateways whose signals are offset by zero and one sample respectively. We note that even an offset of one frequency bin causes a significant time-varying error in phase between the gateways. As a result, summing up these signals would

cause some symbols in-phase to reinforce, while others that are out-of-phase cancel each other out.

Phase-Based Time-Sync Below the Noise Floor: Charm overcomes this challenge by recognizing that small time-errors between two gateways results in a phase difference over time that is predictable. As shown in Figure 3.6, this phase difference is a linear function of time, given by $2\pi f(t)\delta t$, where $f(t)$ is the instantaneous frequency of the chirp (linear in time) and δt is the required timing offset. In principle, one can therefore estimate the slope in phase over time to recover the timing offset. In practice, however, doing so is challenging, particularly when each received signal at each gateway is completely buried below the noise. The phase of such signals at any such gateway simply appears to be random – making any form of linear regression of the slope highly error-prone.

Charm overcomes this challenge using two key properties: First, owing to coarse time synchronization of the gateways (via NTP), any residual timing error between them is limited to a few samples. This allows Charm to iteratively optimize over a small number of time-shifts to infer the offset that leads to the best fit. Second, Charm can extract timing offsets both from the preamble and the data symbols. To see how, notice that our approach only considers the *difference* in phase between the same packet heard at two different gateways. Given that, in the absence of timing offsets, both gateways perceive the same underlying message bits over time, the resulting phase difference would be independent of the transmitted data bits – whether they belong to the preamble or data.

Charm's approach therefore considers the range of possible small offsets between any two received signal sequences. For each candidate offset, it computes the phase difference between the signals as a function of time. It then identifies the true offset between the gateways as the one whose phase difference varies minimally across the entire packet. Given that our approach averages measurements through the entire packet (both preamble and data), it remains highly resilient to noise.

Maintaining Synchronization Across Packets: Finally, Charm can learn the time-offsets between gateways, particularly in busy urban deployments, by using information from past packets. Recall that Charm's coherence is only affected by timing errors between pairs of gateways – not the gateway and any particular client. While these errors may change over time, over small intervals (e.g. hundreds of milliseconds), they are unlikely to change. As a result, one can use the measured time offset from a previous packet to infer the offset at the next packet that follows soon after. This allows us to maintain a history of the time-offsets, smoothed by algorithms such as Kalman filtering with outlier rejection, that helps us better predict time offsets between gateways even when signals from certain clients are too weak to measure reliably.

3.7.2 Joint Decoding at the Cloud

This section answers an important question: How does Charm decide which weak signals received from a set of gateways need to be combined coherently? In other words, Charm must identify which signals at the gateway correspond to the same packet from the same transmitter. It must do so even in the presence of overlapping transmissions from multiple clients at geographically different locations.

Signal Selection: Charm addresses this challenge by using the timing information of packets to infer transmissions that correspond to the same user. It further uses the perceived signal-to-noise ratios and geographic location of the gateways and measures the likelihood that far-away gateways can listen to transmissions from the users at the observed signal-to-noise ratios. It then calculates a feature vector for each received signal that contains two tuple: (1) The time instance at which the packet was received; and (2) The geographic location of the gateway. We apply the OPTICS clustering algorithm [6] to then cluster received signals from multiple clients at any time instance.

Past-clustering, we combine received signals from a subset of clients in each cluster. Specifically, we only choose to combine signals with a sufficiently high signal-to-noise ratio. This is because transmissions that are highly noisy tend to add little additional value yet cost uplink bandwidth.

An important consequence of our clustering approach based on geographic location of the gateway is that it facilitates spatial re-use. Specifically, it is quite possible that weak transmissions from two different neighborhoods occur at the same time but are heard at distinct subsets of gateways. Charm allows us to decode these transmissions simultaneously without mixing up their signals. Indeed, gateways that are geographically in-between and hear interfering signals from both clients can be simply weeded out from clustering due to their poor signal-to-noise ratio.

Joint Decoding Algorithm: Algorithm 2 below describes Charm's joint-decoding algorithm end-to-end. At a high level, our approach retrieves the wireless channels of the signals to be combined at any instance, their timing offsets and frequency offsets computed as described in the above sections. We then eliminate any phase errors owing to time and frequency offsets in the received signals. We then coherently sum up the resulting signals multiplied by the conjugate channels as described in Section 3.4.

3.7.3 Opportunistic Fetching of Information

Our design thus far assumes Charm gateways relay raw I/Q received signals to the cloud, only if their signals are too weak to be decoded yet can be detected. However, this approach can be ineffective for two reasons: (1) On the one hand, the cloud may have already received the decoded data bits from another gateway, meaning that Charm simply wasted uplink bandwidth unnecessarily; (2) On the other hand, some received

Algorithm 2: Joint decoding algorithm

```
1  packets = receive_data(candidates);
2  for p in packets do
3  |  p = e^{j2π(Δ_f)t} p ;                          // Freq Offset Correction
4  |  p = e^{j2πf(Δ_t)} p ;                         // Timing Offset Correction
5  |  h(p)= p / reference ;                          // Channel Estimation
6  end
7  combined_packet=zeros(p);
8  for p in packets do
9  |  combined_packet = combined_packet + h*p ;
10 end
11 decode(combined_packet);
12 SEND ACK;
```

signals may be significantly below the noise floor even to be detected, yet be valuable enough to be relayed to the cloud to be jointly decoded with other such weak receptions.

Two-Phase Data Fetch: Charm overcomes these challenges through a pull-based approach where gateways relay raw I/Q samples to the cloud, only when explicitly asked for by the cloud. Each gateway keeps a circular buffer of I/Q streams as well as any recent snapshots containing a potential packet. For each potential reception, a gateway first reports its signature (center frequency and spreading factor), the time of the reception packet, the perceived wireless channel and signal-to-noise ratio. Charm then performs clustering as described above and requests the raw I/Q samples *only* from clients whose signals were chosen to be combined. Given that latency to the cloud is on the order of few milliseconds, smaller than a typical LoRaWAN packet size (tens, often hundreds of milliseconds), our system can perform decoding virtually in real-time at LPWAN timescales, despite incurring multiple round-trip times in fetching information.

Opportunistic Data Buffering: In some instances, Charm's clustering algorithm may fail to have enough signals to successfully combine and decode a packet using the gateways that detected the packet alone. However, Charm may be able to opportunistically fetch information from other gateways in the same geographical region of the cluster and tuned to the same frequency, who may have received the same signal, yet at a signal-to-noise ratio too weak to detect locally. Charm therefore requires all gateways to store past signals for up to 1.6 seconds (maximum LoRaWAN packet length) in the past in a 5 MB circular buffer. This allows Charm to query and fetch signals from gateways, even in scenarios where only one gateway in the entire network was able to locally detect a signal from a given transmitting client.

3.8 Integration with LoRaWAN

Charm is implemented as a service running on a campus-wide OpenChirp network at Carnegie Mellon University which we covered in Section 2.4. We currently have four gateways mounted on rooftops providing wide-area coverage and eight auxiliary indoor gateways extending coverage into remote parts of campus. The only modifications required to make a gateway Charm-enabled is the additional hardware platform for receiving raw I/Q streams and a modified LoRaWAN packet forwarder that runs the packet reception event detector, maintains a circular buffer of I/Q streams and brokers interactions with the Charm cloud. Communication between gateways and the cloud is managed using the OpenChirp's MQTT publish subscriber messaging layer where compressed Charm packets can be easily grouped and organized based on location. The Charm service can instruct clients to switch to faster data rates (instead of the normal data rate negotiation process) by spoofing improved SNR values during the join process and subsequent receptions. In this way, Charm can seamlessly operate with existing LoRaWAN devices with no modification.

3.9 Evaluation

We evaluate Charm in both indoor and outdoor environments through proof-of-concept experiments and large-scale trace-driven simulations. We use two testbeds on the Carnegie Mellon University campus and around the city of Pittsburgh. Eight user-deployed gateways built using our custom hardware platform support a testbed covering a 0.6 sq.km. area around campus, which is used to study Charm's performance with regard to local packet detection, range and data-rates. Four rooftop gateways support the OpenChirp LPWAN network described in Section 2.4 that we use to acquire traces for large-scale simulations.

3.9.1 Local Detection Algorithm

We perform trace-driven simulations to demonstrate an improvement in the local packet detection of LoRa packets in a noisy environment. To perform this experiment, we collect data at different spreading factors at high SNRs. We then measure the signal power and progressively increase additive white Gaussian noise in the signal. At every dB of decrease in SNR, we test the state-of-the-art LoRaWAN decoding algorithm against Charm's local and enhanced detection algorithm, where the former uses the preamble alone and the latter uses both preamble and data in its optimization.

The results in Figure 3.7 show that Charm's local detection algorithm outperforms the LoRaWAN detection algorithm. Further, Charm's enhanced detection algorithm outperforms Charm's local detection algorithm by up to 10 dB, since it uses data symbols in addition to the preamble. Our results also reveal

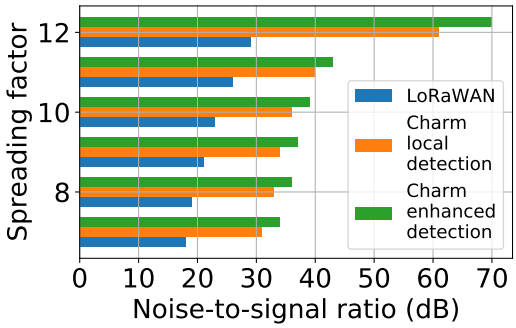


Figure 3.7: Local packet detection capability for low SNR receptions. Methods using data and preamble symbols outperform preamble-only methods.

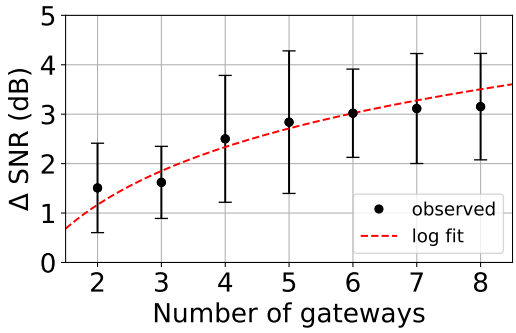


Figure 3.8: Charm's diversity gain: Combined SNR increases logarithmically as more gateways receive a transmission

a 33% increase in the negative SNR under which a packet can be detected, when compared to LoRaWAN – a gain of between 16-30 dB. To put this in perspective, this is equivalent to a boost in SNR by coherent combining of more than 40 gateways. Thus, under identical transmission and noise conditions, Charm's packet detection is comparable to a detection requiring at least 40 gateways performing coherent combining.

3.9.2 Diversity Gain

Next, we evaluate Charm's improvements to combined SNR, after coherently combining multiple transmissions across geographically diverse receivers. These benchmarks are completed on a testbed covering 0.6 sq.km. using an ensemble of 8 user-deployed gateways equipped with our custom LPRAN hardware. This testbed spans multiple buildings and open spaces between them, and is supposed to emulate a dense urban deployment. We measure the mean and standard deviation in SNR improvement as a function of the number of gateways, for clients at different locations using multiple spreading factors.

Our results, shown in Figure 3.8, reveal remarkable SNR improvements, which logarithmically increase

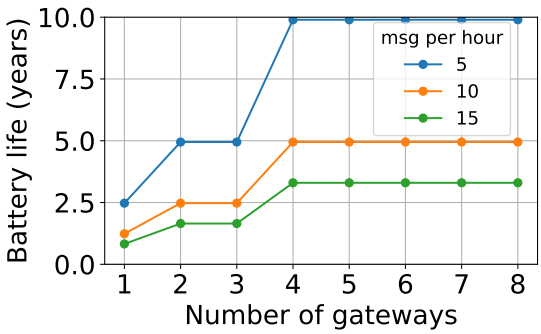


Figure 3.9: Client battery life improves as a larger number of receiving gateways permits switching to higher data rates

	SF7	SF10
LoRaWAN	<60 m	<60 m
Charm with 4 gateways	<60 m	<100 m
Charm with 8 gateways	<200 m	<200 m

Table 3.10: Range in congested indoor urban settings

with the number of gateways. Across experiments, Charm gave an average observable improvement of 1 dB with the addition of each new receiver. This improvement is valuable, given that every 3 dB of gain allows us to use the next spreading factor. Any increase in spreading factor halves the transmission air time and the resulting energy expenditure. Figure 3.9 depicts the improvement in battery life of an indoor LoRaWAN client with an increasing number of gateways collaborating to decode its signal. We observe that the battery life for a device transmitting 5 messages per hour at SF11 improves from 2.5 years to 10 years (SF9) with 4 or more collaborating gateways.

3.9.3 Range Improvement for Indoor User-Deployed Gateways

In typical urban settings, users would deploy a large number of gateways. Indoor settings reduce the range of a LoRaWAN device and the data rate it can support, even for short distances of tens of meters. We deploy Charm in a congested urban building and demonstrate that collaboration can improve the maximum range the LoRa device can use at any given data rate.

In this test, we compare a group of regular LoRaWAN gateways that independently decode transmissions against Charm coherently combining signals from an ensemble of 4 and 8 gateways. The distances reported in each case are between the transmitter and the closest gateway. Our results are shown in Table 3.10. Note that the ranges we observe here are smaller than outdoor gateways, owing to attenuation inside buildings and transmission power limits on small portable battery-powered devices. In this context, a regular LoRaWAN

gateway can service a client up to approximately 60 m away. In contrast, Charm consistently supports a higher maximum range at each spreading factor. Four collaborating Charm gateways can communicate up to 100 m away, while 8 Charm gateways go as far as 200 m.

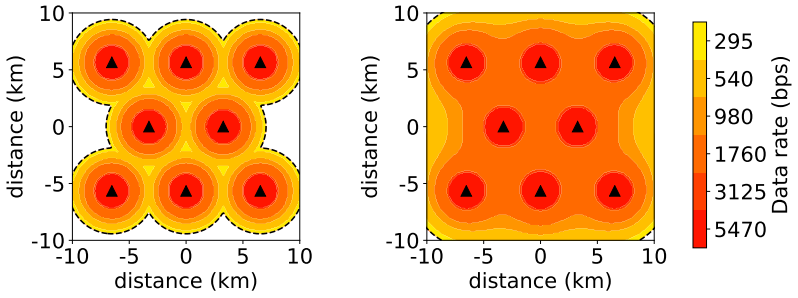
3.9.4 Effect on Coverage and Device Data Rates

In this section, we use trace-driven simulations to show the advantages of Charm in improving coverage area and client energy consumption in both planned and unplanned gateway deployments. The signal power at any given receiver is estimated using the log-distance path loss model. The model is calibrated using 4850 points collected during our coverage tests in Section 2.4.5. The log-distance parameters are $L_0 = 98.0729dB$ for $d_0 = 40.0m$, $\gamma = 2.1495$ and flat fading $\sigma^2 = 100.0724$. Sensitivity values for the gateway are taken from [13] to determine the SNR threshold required to decode a transmission. In an urban environment with many obstacles and reflectors, we observe a maximum range of 3.77 km with a transmit power of 15 dBm as opposed to the marketed range of 10 km with line-of-sight. As we are interested in the trend of changes, we provide an optimistic estimate and ignore the effects of fading in the simulation (assume $\sigma^2 = 0$).

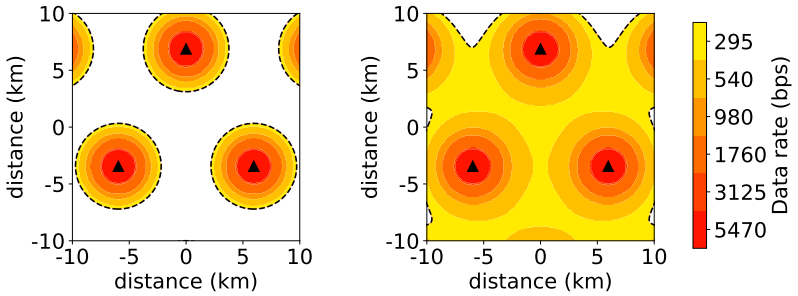
We perform simulations with three sample deployment scenarios. Figure 3.11a is an ideal dense planned deployment, where gateways are placed in a hexagonal grid 6.53 km apart from each other ($= 2 * 3.77 * \cos(\pi/6)$ km). Such an arrangement, popular in cellular deployments, provides optimal coverage with no gaps when using an independent decoding scheme, like in LoRaWAN. Figure 3.11b shows a planned sparse cellular arrangement with gateways 10.05 km apart from each other, and can provide gap-free coverage with coherent combining. Figure 3.11c is a randomly-generated unplanned deployment, a consequence of user-deployed gateways.

With a fixed transmit power of 15 dBm on the client device, Figure 3.11 shows the region where Charm's local detection followed by joint decoding shows an improvement in either coverage, client data rates or both compared to independent decoding on gateways. Areas are colored if they have any coverage, and the shade of the color determines the data rate (red is a better data rate than yellow). As seen in each of the figures, Charm shows an improvement in the coverage area, an increase in client data rates as well as both simultaneously.

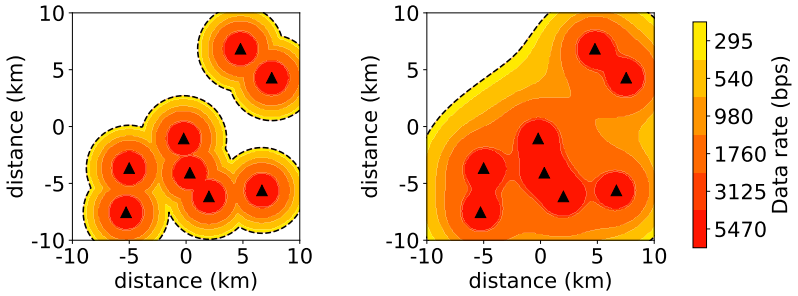
Some specific examples of Charm's improvements are as follows: In the planned dense deployment of Figure 3.11a, Charm improves coverage area by 46% and substantially boosts the data rate around the centroid areas. For the planned sparse deployment of Figure 3.11b, Charm allows us to increase the inter-gateway distance to 11.92 km and still maintain gap-free coverage (a decrease in gateway density by a factor of $(11.92/6.53)^2 = 3.33$). With an unplanned deployment such as in Figure 3.11c, Charm not only improves



(a) Dense cells



(b) Sparse cells



(c) Random placement

	dense cells	sparse cells	random
increase in coverage	46.60%	97.85%	74.59%
data rate +1 (energy/2)	35.33%	38.82%	33.70%
data rate +2 (energy/4)	22.30%	0%	25.82%
data rate +3 (energy/8)	2.26%	0%	3.48%

(d) Summary of trace-driven simulations

Figure 3.11: Improvement in coverage area and data rates due to Charm (right column) compared to LoRaWAN (left column)

coverage and data rates but also manages to fill in islands and orphaned regions with coverage. This is particularly relevant to urban regions where areas of bad coverage are formed in building basements and other indoor regions as seen in Figure 2.7. These examples provide an insight to Charm's substantial benefits to existing and future LPWANs. A detailed summary of these results is shown in Table 3.11d. Improvements are reported as percentages with reference to the area covered by LoRaWAN in each deployment. Every increase in the data rate doubles the battery life of a client device. Some regions in the simulation show up to 8 × energy savings.

3.10 Summary

This chapter presents Charm, a novel system that improves battery life and range of LPWAN clients. Charm achieves this through a mechanism that pools together weak received signals across multiple gateways at the cloud in order to jointly decode them. Charm introduces a hardware-software design that detects weak signals at the gateway, to provide scalability at the cloud. A pilot evaluation of Charm on a network of twelve LoRaWAN gateways serving a large neighborhood of a major U.S. city demonstrates a large improvement in coverage and client battery-life. Our results reveal the following:

- **Battery-Life:** By coherently combining across 8 base stations, Charm improves the SNR of a typical LoRaWAN transmission by 3.16 dB, extending battery life by up to 4×.
- **Range:** We improve the maximum communication range of 8 indoor user-deployed gateways in urban settings from 60m in LoRaWAN to 200 meters using Charm, an overall increase in coverage area by 10×.
- **Coverage:** Our trace-driven simulation, based on city-wide drive tests, estimates an overall increase in coverage area by up to 2× due to Charm over LoRaWAN.

An interesting side-benefit of Charm is its impact on scalability of the network overall. Given that Charm improves coverage, one might expect a large number of collisions from transmitters who newly gain coverage with existing ones. Counter-intuitively, this is not the case, because Charm allows devices across the board to transmit at faster data rates, increasing available air time in the network.

Chapter 4

Precise synchronization

Access to accurate time is essential to almost all modern systems, though the level of precision varies with the application. Wireless systems like WiFi and cellular require microsecond-accurate timing to efficiently use time-slots. LPWAN networks like LoRaWAN, which are more forgiving, still require their gateways to be synchronized with their servers to within a few tens of milliseconds. If LPWAN gateways could be accurately synchronized to the order of a few nanoseconds, we could enable valuable functionality like multilateration to find the location of LPWAN devices to within a few meters. Charm, which enables distributed reception using multiple gateways, had to introduce complex mathematical techniques in Section 3.7.1 to deal with the problem of time and clock frequency offsets between different gateways. This could have been avoided, were it easy to synchronize LPWAN gateways precisely. Stable synchronization at the scale of nanoseconds is challenging because commonly used oscillators and electronics introduce phase noise at this same scale. Additionally, if wireless synchronization were to be achieved wirelessly, we have to compensate for propagation path delays which can be difficult to estimate in dynamic environments.

The most common form of time synchronization is the Global Positioning System (GPS). This is an excellent option for LPWAN gateways with access to the open sky. Unfortunately, GPS signals cannot easily penetrate buildings, and this limits its use in indoor or other GPS-denied scenarios. The best wired time synchronization solutions, like the Precision Time Protocol (PTP) [43], can achieve accuracies as low as 25 ns but require updating all intermediate network switches. WiFi fine time measurement (FTM) systems designed to operate on existing WiFi channel bandwidth cannot provide the necessary range resolution, due to their limited bandwidth and changing multipath conditions. Also note that in many wireless systems that currently provide synchronization (e.g. WiFi FTM), though the system itself may be able to synchronize very accurately with itself, it is very difficult to transfer this synchronization to another device, like an LPWAN gateway. In this chapter, we discuss the design and evaluation of Pulsar, which provides clock

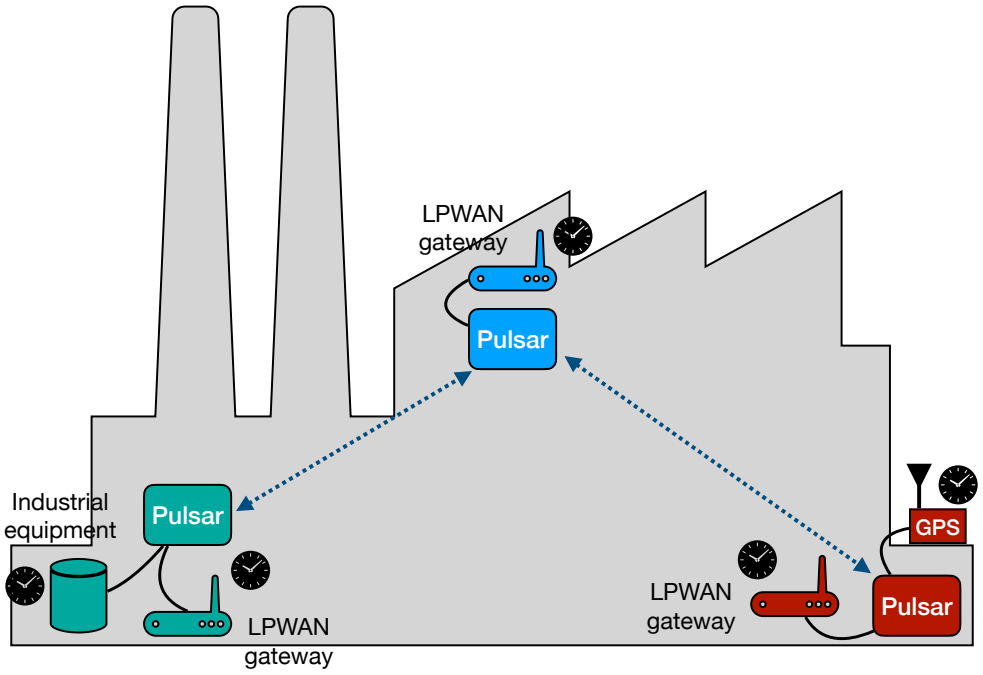


Figure 4.1: Illustration of Pulsar network providing precise time to LPWAN gateways and other time-sensitive equipment.

synchronization of better than five nanoseconds per communication hop across a multi-hop network. Pulsar is developed as an independent platform that interfaces with other devices using standard time and frequency signals so as to decouple time synchronization from wireless protocol implementation.

4.1 Contributions

The contributions of this work are:

- A novel hardware platform that is able to perform wireless time-of-flight propagation-aware clock synchronization at better than 5 ns resolution per communication hop that can be easily integrated with existing SDR systems
- An end-to-end analysis and evaluation of timing uncertainty provided by the platform
- The design of a propagation-aware clock synchronization algorithm

4.2 Pulsar's Approach to Clock Synchronization

Understanding Timing Errors: Time synchronization protocols suffer from five main sources of error that are associated with: (1) transmit time, (2) propagation time, (3) receive time, (4) residency delay and (5) clock instability. Transmit and receive timing errors result from jitter and offset when timestamping packets. Residency delay results from messages being in buffers after a packet has been constructed. In message passing protocols like NTP [58], the majority of error is associated with asymmetry in round-trip message passing times. Propagation time is the delay resulting from the time it takes a signal to travel over the air or through a medium like wire or fiber. One nanosecond corresponds to the time it takes light to travel approximately 30 cm. An offset of 100 ns could simply be a 30-meter difference in distance. PTP uses hardware-level timestamps to estimate propagation time of network signals. This is extremely difficult in wireless systems because of the error in timing associated with locking onto preambles in a noisy channel. Finally, clock instability is the result of error in the frequency of local oscillators that can change depending on physical properties like temperature or crystal aging. A quartz crystal found in a typical electronic device could drift as much as 1 μS per second. Even the best oven controlled oscillators begin to drift on the order of nanoseconds per second over periods longer than a few seconds.

Given these error sources, an ideal wireless time transfer system would benefit from two main technologies: (1) a stable clock source to minimize drift and message passing overhead and (2) a radio that operates across a wide bandwidth to improve the theoretical range resolution, and one that can perform accurate timestamping of packets. Pulsar leverages innovations on both fronts by using a chip-scale atomic clock (CSAC) as its primary clock source and a commercial-off-the-shelf UWB radio capable of sub-nanosecond packet time stamping.

Precise Message Timestamps: For the radio, we use the DW1000 UWB chipset from Decawave[23] that provides a nominal 15.6 ps timestamp precision of packet transmit and reception through use of equivalent time sampling on a repetitive pulse train. The combination of stroboscopic sampling and the fact that UWB uses short pulse durations makes these chips ideal for precise timestamping and ranging applications. UWB radios provide good time and range resolution in multipath environments due to their large bandwidths. It is important to note that the presence of multipath itself is not the problem, as a signal inside a wire also propagates because of multipath reflections. However, in a wireless environment, the nature of this multipath changes rapidly and unpredictably, which is when a wide-band signal can help us distinguish the different paths and choose the correct one for timestamps. The DW1000 is designed primarily for TOF ranging applications and can provide centimeter-level distance corrections when given line-of-sight. These distances can be used to estimate speed-of-light propagation delays.

Using Precise Clocks: The Pulsar includes a Quantum SA.45s Chip Scale Atomic Clock (CSAC) that provides a short-term stability (Allan Deviation) of $2.5 * 10^{-10}$ with an averaging period (τ) of 1 second. The CSAC is connected directly to the UWB radio and an ARM processor using a programmable low-jitter PLL. The high stability and low drift of the CSAC not only improves the DW1000 in terms of frequency locking performance, but it enables synchronization and ranging over longer intervals which improves multi-hop performance. It is important to note here that the accuracy of a CSAC may not be completely necessary, but developing a system with a very good clock will help us identify which clock properties are important.

Synchronization Between Hardware Subsystems: One of the main challenges in our system is utilizing DW1000 timestamps in a manner that allows for precise clock conditioning. The digital subsystem of the DW1000 runs at 38.4 MHz, which means that all I/O is discretized to 26 ns. A significant contribution of this work is that we provide a hardware mechanism for pushing synchronization accuracy below the I/O discretization level. We utilize the PLL to synchronously clock the radio and processor subsystems while using the CASC PPS signal as a common event for time stamping. Since the PLL provides frequency locking but cannot be phase aligned to the input clock from the CSAC, the radio and clock would still have an unknown phase offset up to 26 ns. We are able to improve error by using a phase measurement sub-system to then measure the error between the PPS signal and the outputs from the PLL. We can then compensate for the phase error in software to achieve below 5 ns of accuracy. Finally, the system provides a synchronous PPS signal along with a phase locked 10 MHz output that can be used to synchronize our LPWAN gateways and other communication equipment.

Multihop Synchronization: To expand the coverage of any synchronization system, we have to explore methods to use multihop communication that go beyond a simple star topology. In propagation-aware time transfer systems, the device location and timing accuracy are tightly coupled. In protocols like NTP and PTP, time is distributed along the edges of a tree. Related work has shown that not all links and clocks should be treated equally [81]. One of the benefits of the broadcast nature of wireless communication is that multiple nodes within a network can perform pair-wise ranging with each other to capture information about the topology with more ranging options as compared to wired systems. As part of the Pulsar's synchronization protocol, we have a graph realization and a low-jitter link selection step where the system collects range measurements between nodes to capture the topology of the network. Graph realization can be solved in an attempt to find routes that minimize jitter caused by non-line-of-sight (NLOS) communication. Similar to localization systems like those used in optical motion capture, it is possible to augment the graph structure with a mobile UWB device that adds additional ranging measurements to improve accuracy. This graph also provides the physical location of nodes, a critical component to many wireless applications.

4.3 Related Work

There is an abundance of related work in both clock synchronization as well as ranging and localization systems. We first discuss clock synchronization and then look at mechanisms for accurate ranging, which can be used to remove propagation delay errors. We also discuss related work from the wireless MIMO community.

4.3.1 Clock Synchronization Approaches

Significant effort has addressed establishing a common notion of wall-clock time [58, 45]. The Network Time Protocol (NTP) uses round-trip message delay averaging to set times. We adopt many similar concepts to NTP, like clock discipline and network-delay estimation.

Various message passing approaches have looked at minimizing access, transmission and reception time in wireless systems. The reference broadcast synchronization [28] (RBS) scheme uses timestamps exchanged between multiple receivers to eliminate all transmission delays with the exception of propagation delays. This approach targets the sources of timing jitter associated with wireless devices and averages over multiple transmissions to achieve tight pairwise clock synchronization. The Pulsar platform adopts a similar approach using beacon messages, except that it adjusts for propagation delays. The flooding time synchronization protocol [56] and the time-sync protocol for sensor networks [34] (TPSN) use hardware timestamping to eliminate these similar sources of timing jitter. Messages are flooded across the network forming a spanning tree that periodically compensates for drift. Local clock rates are adjusted to help reduce drift, which could also be achieved using our module. Both approaches could be applied to the Pulsar platform and would improve performance compared to their original implementations, due to Pulsar's fine-grained timestamping capabilities. In [31], the authors propose a scheme called Glossy, which utilizes constructive interference at the symbol level to boost the probability of successful time synchronization messages. Glossy currently provides the best-in-class WSN synchronization, which is on the order of $500ns$ and does not compensate for propagation delay.

Multiple synchronization approaches leverage external hardware to receive global time broadcasts. The WWVB atomic clock broadcast from NIST uses a 50 kW radio tower located in Boulder, Colorado to transmit a 60 Khz time beacon. This is ideal for outdoor applications within the tower's broadcast range, but the radio transmission does not penetrate far into buildings. The signal also suffers from high levels of jitter with offsets due to the long transmit distances. The Global Positioning System (GPS) [64] uses precise clock synchronization derived from satellite transmissions for localization and timing. This is achieved using the Time-Difference-of-Arrival (TDOA) of radio messages to estimate location as well as synchronize

receiver clocks with an atomic clock-driven infrastructure. Unfortunately, GPS does not penetrate buildings and requires at least three satellites in order to compute precise time (a single satellite gives crude time on the order of micro-seconds since it cannot determine distances). GPS time receivers have commonly been used as sources to discipline NTP servers and often use temperature-controlled oscillators to improve timing stability. These have even been implemented in software for wireless sensor networks [82].

In [5], the authors study using nearly simultaneous receptions of various sources, both natural and man-made, for synchronization. For example, optical pulsars (from which we adopt our name) can broadcast flashes of light simultaneously visible to large regions of the earth. These reception times are known to be nearly simultaneous to all viewers and hence can be used as synchronization points.

4.3.2 Synchronization for Software-Defined Radios

There have been multiple proposed approaches for tight clock synchronization from the wireless community. [97] proposed using power-line communication (PLC) as a back channel for wireless synchronization. The system is able to achieve an average synchronization accuracy of 225 ns with as high as 400 ns. While ideal for small area clock distribution, PLC requires repeaters to go across circuits and can be susceptible to noise that is difficult to eliminate. SourceSync [69] presents a system that is able to harness sender diversity through tight time synchronization using an SDR. The system is able to achieve better than 20 ns time synchronization, but is limited to a single collision domain, and like many SDR-based approaches modifies the underlying MAC protocol to include synchronization capabilities. AirSync [11] and MegaMIMO [70] use similar approaches with SDRs that modify the underlying MAC. In contrast, our approach provides an external input at 5 ns per hop across a network with the sole purpose of providing synchronization. This decouples the time synchronization from the underlying wireless MAC.

4.3.3 Localization and Ranging for Time Transfer

There have been some efforts in performing propagation-aware time synchronization without GPS [94, 50, 25]. [94] presents an approach for doing propagation-aware synchronization that uses known locations of beacons to back compute propagation times. It assumes known locations from GPS markers, but this approach could utilize the Pulsar platform, and given their required time scales, would likely see a significant improvement in terms of the number of supported nodes. [50] takes a similar approach to what we propose, using a sub-GHz CC1101 radio for internal synchronization. As is the case with most existing platforms in the WSN community, they operate in the 200-1500 μS range. It is unfortunately difficult to transfer the time synchronization attained on these systems to another external device like an LPWAN gateway.

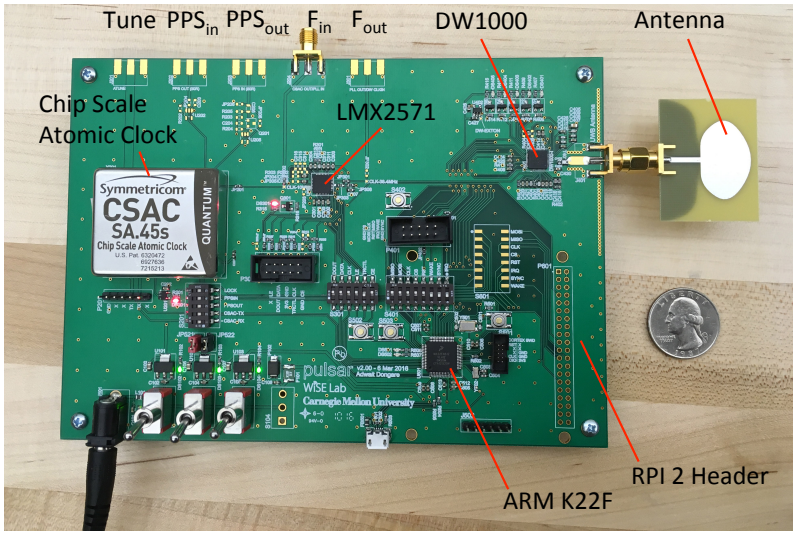


Figure 4.2: Pulsar hardware photograph

4.4 Platform Design

In this section, we discuss our hardware design and then address our specific sources of synchronization error. We look at how these sources of error can be reduced given our proposed architecture. Our platform is open-source, with all hardware designs available on Upverter and code to be released on GitHub.

4.4.1 Pulsar Hardware

The Pulsar platform, shown in Figure 4.2, is 18 cm by 12.5 cm. The block diagram of the Pulsar board in Figure 4.3 shows four main components: (1) the CSAC, (2) a frequency synthesizer, (3) a UWB radio and (4) an ARM processor. The entire Pulsar board consumes a peak of 200 mA at 3.3V, most of which is consumed by the radio and the CSAC heating element. The main output of our system is a 1PPS signal along with a phase locked 10 MHz clock which will be synchronized across the entire network of nodes. These outputs represent the standard time-transfer interface for most SDR and communications hardware.

The CSAC is a Microsemi SA.45s module that outputs a 10 Mhz signal with a short-term stability (Allan Deviation) of $2.5 * 10^{10}$ over a 1 second averaging period with a long-term aging of $< 9 * 10^{-10}$ per month, and a maximum frequency change of $5 * 10^{-10}$ across a temperature range of -10 to 35 degrees Celsius. The CSAC has the ability to be disciplined from an external high-precision PPS source (PPS in synchronization), improving its phase and frequency performance to within 1 ns and $1.0 * 10^{-12}$ respectively. In our experiments, we pre-calibrated the clocks from a single GPS source. The CSAC has a variety of I/O including PPS in, PPS out, an analog tuning input for phase adjustment and a digital interface over serial. The CSAC can be digitally servoed at up to a maximum frequency steer of 4 parts in 10^8 through a serial

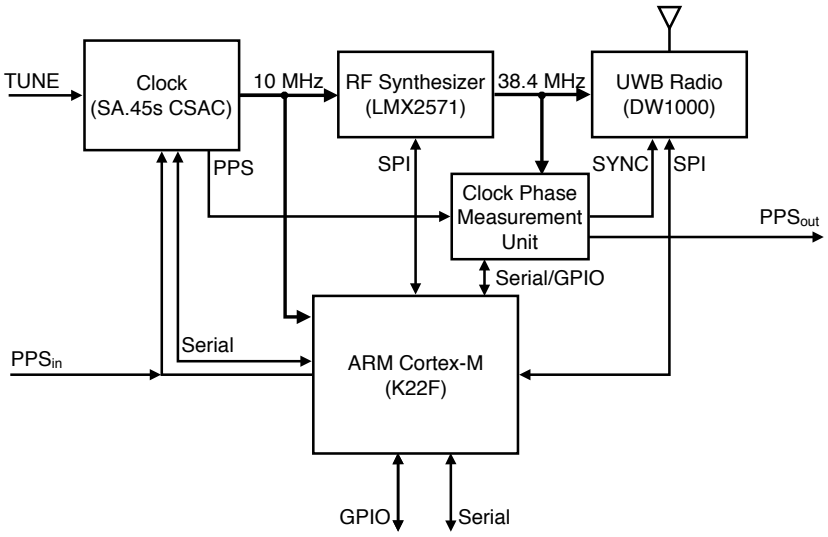


Figure 4.3: Pulsar Block diagram with interconnects.

interface or the analog tuning input. Since it would take an extremely long time to servo PPS outputs into alignment, we feed a GPIO pin from our main processor into the PPS input of the CSAC as part of an initialization process (manual PPS in synchronization). The CSAC uses an oven controlled quartz oscillator (OCXO) that is disciplined at 1 Hz by a resonance cell containing rubidium 87 that is heated into a vapor. The vapor is illuminated with light from a semiconductor laser diode which is naturally modulated at 6.834 GHz. Once the laser drives the atoms into an oscillating state, they absorb less light, which allows the system to determine if the light is modulated at the same frequency as the atomic source. An inner control loop servos (based on light intensity) to condition the OCXO. Using the excitation of rubidium atoms as a reference is what provides the long-term frequency stability.

The 10 MHz output from the CSAC is connected to a LMX2561 low-jitter frequency synthesizer from Texas Instruments and to a hardware counter on the main ARM processor. The frequency synthesizer is used to convert the $10MHz$ signal into a 38.4 MHz signal that can drive the UWB radio and other related subsystems. The LMX2561 contains a fractional PLL that can be programmed to generate any frequency from 10 MHz to 1344 MHz with low phase noise (-145 dBc/Hz at 1 MHz) with a PLL noise floor of -231 dBc/Hz and better than -75 dBc/Hz in terms of spurs. This further improves phase noise entering the radio. In cases where you do not have a tunable clock source (like a CSAC), the PLL can also be used to tune an incoming clock signal. Introduction of a PLL into a clock system results in the loss of absolute phase information regarding the output signal with respect to other signals (e.g. 1PPS and 38.4 MHz). However, stable PLLs will introduce a phase offset which is held constant during lock, and we can use this insight to measure and compensate for the error.

At startup, the PLL and CSAC are configured by an NXP Kinetis ARM K22F Cortex-M processor running at 120 MHz. The ARM processor has a variety of connections to control all of the Pulsar's subcomponents as well as interconnect with external devices using a RPI2 compatible header. The main processor has 512 KB of Flash, 128 KB of SRAM, an FPU and on-board DSP.

We use a Decawave DW1000 UWB radio for communication and timestamping. It has the ability to timestamp packet arrival with a resolution of 15.6 ps through equivalent time sampling of a pulse stream that is part of the message preamble. UWB is an excellent communication source for ranging applications since the pulses can be made to be extremely narrow in time and hence wide in frequency. From radar literature, we know that range resolution in multipath environments is related to the time bandwidth product.

The DW1000 has a synchronization line (SYNC) that can be used to reset an internal 40-bit counter that increments at 64 GHz or deterministically trigger a radio transmission. This SYNC input can be used to reset the system time-base for the radio messages (known as *time-base reset*). As is the case with most digital radio platforms, the SYNC pin will only be read on the next rising edge of the 38.4 MHz clock driving the I/O subsystem of the radio. This introduces up to 26 ns of error unless the source driving the SYNC line is phase aligned with the 38.4 MHz. To achieve synchronization accuracies below the I/O discretization level of the radio's digital system, we feed the raw CSAC PPS signal into the SYNC line (but only resetting the time-base when required) and then use a phase measurement unit (PMU) to determine the phase error between the PPS input and the next 38.4 MHz clock edge. Knowing this phase error allows us to correct the Decawave time stamps to within a few nanoseconds. This phase error only needs to be computed once at startup and can then be removed as a static offset from the received time stamps in software. In our current hardware implementation, we perform the PMU measurement externally and feed the phase error back into the main ARM processor using its serial port. Alternatively, it is possible to remove the phase error between the PPS and Decawave input by generating a PPS signal directly from the 38.4 MHz clock with a divider. Note that this must be a counter/divider and not a PLL, which would again introduce random phase error. A counter would introduce a deterministic delay that could be removed through calibration. However, this configuration would also make it difficult to perform time synchronization (external as opposed to internal) since the clocks would be internally consistent but unable to servo towards an input like a GPS PPS signal.

4.4.2 Sources of Error

Nanosecond scale clock synchronization is difficult due to the variety of errors that can accumulate in the timing system. Some of these errors can be identified, some statistically filtered out while others are completely unobservable and dependent on the architecture of the system. A critical part of this work is to

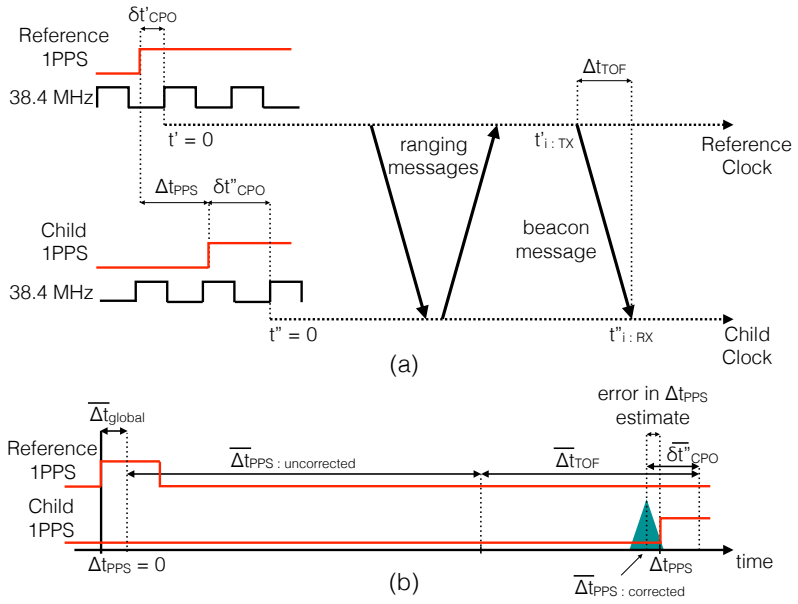


Figure 4.4: Timing in the Pulsar platform for phase offset estimation

identify and mitigate various sources of timing errors encountered during synchronization.

Frequency offset and stability errors are the easiest to identify and correct. Receiver-only systems (like the one we describe in Section 4.5.4) can be used to calibrate for frequency offsets by locking on signals with known time differences. Frequency stability is a fundamental error source modeled using Allan deviation described later in Section 4.4.2.

Phase errors are more complex to estimate and correct, stemming from the difficulty in establishing a common reference point across the multiple clock domains found in typical electronic systems. Phase is also highly susceptible to various types of propagation delays in the signal path that do not affect frequency estimates. Figure 4.4 describes various phase error sources in our system.

The main quantity of interest is the time offset (Δt_{PPS}) between two 1PPS signal lines on different nodes. If the radio clocks on the nodes are to be started perfectly in synchronization with the 1PPS line ($\delta t_{\mathrm{CPO}} = 0$ for both nodes), then timestamps for a message i provide an estimate of the 1PPS offset:

$$\Delta t_{\mathrm{PPS:uncorrected}} = t''_{\mathrm{i:TX}} - t'_{\mathrm{i:RX}}$$

This estimate does not consider propagation delays due to time-of-flight (Δt_{TOF}) that can be computed and compensated for through message passing as described later in Section 4.5.1.

As shown in Figure 4.4, the clock radio does not start at the same instance as the start of the 1PPS line for two reasons: (a) electronic signals take finite time to rise before the radio's CMOS logic can detect them and (b) the digital I/O on the UWB radio only samples on the positive edges of its 38.4 MHz I/O clock.

We call the combined error due to these effects the *clock phase offset (CPO)* which is represented by δt_{CPO} for each node. The PLL used to bridge our 10 MHz CSAC clock domain and 38.4 MHz UWB radio clock domain locks the relative phase between them in a 25:96 ratio, but we lose information about the absolute phase difference between them which was previously provided by the 1PPS line in the 10 MHz domain.

Assuming ideal timestamping on both nodes, Figure 4.4 (a) gives us:

$$\Delta t_{\text{PPS}} = t'_{\text{i:TX}} - t''_{\text{i:RX}} + \Delta t_{\text{TOF}} + \delta t'_{\text{CPO}} - \delta t"_{\text{CPO}}$$

We design our synchronization protocol to operate on a spanning tree across the network to simplify distribution to intermediate nodes. We combine the 1PPS offsets computed from upper layers of the tree with $\delta t'_{\text{CPO}}$ into a single variable Δt_{global} that can be passed to the lower layers. This results in the final offset estimation expression as

$$\begin{aligned} \Delta t_{\text{PPS}} &= \Delta \bar{t}_{\text{PPS:uncorrected}} + \Delta \bar{t}_{\text{TOF}} + \Delta \bar{t}_{\text{global}} - \delta \bar{t}_{\text{CPO}} + \epsilon_t \\ &= \Delta \bar{t}_{\text{PPS:corrected}} + \epsilon_t \end{aligned} \tag{4.1}$$

where ϵ_t is the error in estimation.

Allan Deviation

The traditional characterization of oscillator stability is a plot of Allan deviation, defined using a series of relative frequency estimates between a clock and a reference [4] [57]. Each point on the Allan deviation (σ_y) graph denotes the expected standard deviation in the relative clock frequency (y) for a given sampling interval (τ).

$$\sigma_y^2(\tau) = \frac{1}{2} \left\langle (\bar{y}_i - \bar{y}_{i-1})^2 \right\rangle_i \tag{4.2}$$

An Allan deviation plot helps understand the limits of an oscillator, with respect to frequency and phase stability. This can be used to select an optimal message passing rate (τ_{update}).

We measure Allan variance using two nodes placed in close proximity (approximately 1.5 m) with line-of-sight of each other. The radio is configured with the default bandwidth that improves timestamping performance as described in Section 4.6. A GPS-calibrated Pulsar board is used as a transmit-only reference node with various receiver nodes. In Figure 4.5, we compare receiver nodes clocked by a regular Quartz

crystal, a TCXO and another CSAC. Message transmit and receive timestamps are collected over a period of 10 hours and used to estimate fractional frequencies as described in Section 4.5.1. Allan deviation for *good* clocks (exponential phase noise and Gaussian frequency noise [57]) does not vary much over short intervals. This can be leveraged for estimation if messages are not equally spaced in time. Since the measurements are performed with the complete Pulsar system, they also include any additional errors added by the RF synthesizer, UWB radio and processor.

Allan deviation plots for *good* oscillators are smooth with two distinct parts: the negative slope phase line at lower intervals and the positive slope frequency line at higher intervals. Phase noise is added by PLLs, time-discretization, interrupts (in case of software timestamps), timestamping algorithms, etc., and shifts the phase line upwards. This can be observed since we see that the CSAC-clocked Pulsar's phase line is higher than the datasheet value, due to phase noise added by other components and the reference CSAC. Frequency noise may be added by factors like temperature variation, motion, errors in frequency locking, etc. The intersection point of these two lines is an oscillator characteristic called the *Allan Intercept*. The effective update time for a clock synchronization protocol must be less than or close to the interval period of the Allan intercept for the best performance synchronization.

Our results in Figure 4.5 show that crystal oscillators are not suitable at all, and even TCXOs have such a short coherence time that they would have to resynchronize continuously. CSACs are extremely stable as their Allan intercept is not even visible on the time scale of the curve. However, if we can afford to resynchronize occasionally – say, every few seconds – we could use another oscillator with similar short-term

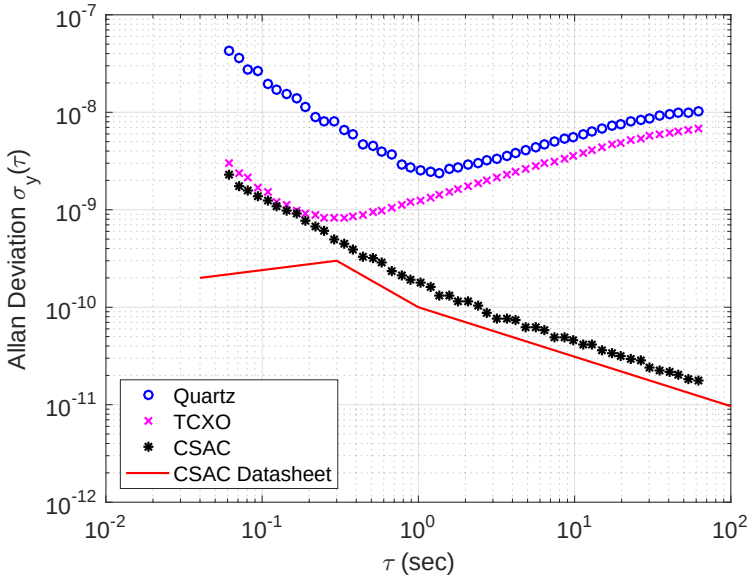


Figure 4.5: Allan deviation between nodes given different clocks

stability but worse long-term stability, e.g. an oven-controlled oscillator.

4.4.3 Pulsar Firmware

The firmware on the Pulsar platform is responsible for configuring the hardware peripherals, tracking peripheral failures as well as arbitrating the message passing and synchronization protocol. It is developed as a set of FreeRTOS v8.2.3 task routines and driver functions for component setup, message passing, USB communication and time conversion written in C. Debugging functionality is available through SWD and Serial and can be accessed through an Eclipse ARM IDE environment or through the command line and GCC Embedded.

A set of watcher tasks are responsible for initializing each of the CSAC, RF Synthesizer and UWB radio. They keep track of events like peripheral lock, reset and halting, and informing dependent tasks of these events. The radio watcher task is also responsible for synchronizing the radio clock with the CSAC PPS so that they share a synchronized time-base. The CSAC watcher task is additionally responsible for bootstrapping PPS alignment corrections in the current implementation of our protocol. A messaging task waits for all required peripherals to lock before starting message passing between nodes as required by the protocol. A disciplining task and some synchronization tasks are responsible for computing and applying all necessary phase and frequency corrections (except the PPS bootstrap, which is delegated to the CSAC watcher task). Phase and frequency offset estimates are provided to higher-level applications through a Serial interface to internally correct for them. Finally, a command task accepts inputs from the user over Serial to change mode of operations.

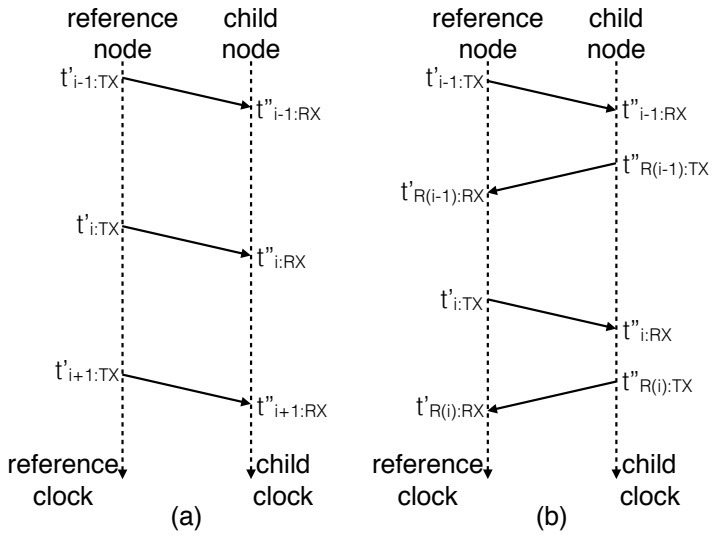


Figure 4.6: (a) One-way and (b) two-way message passing with timestamps.

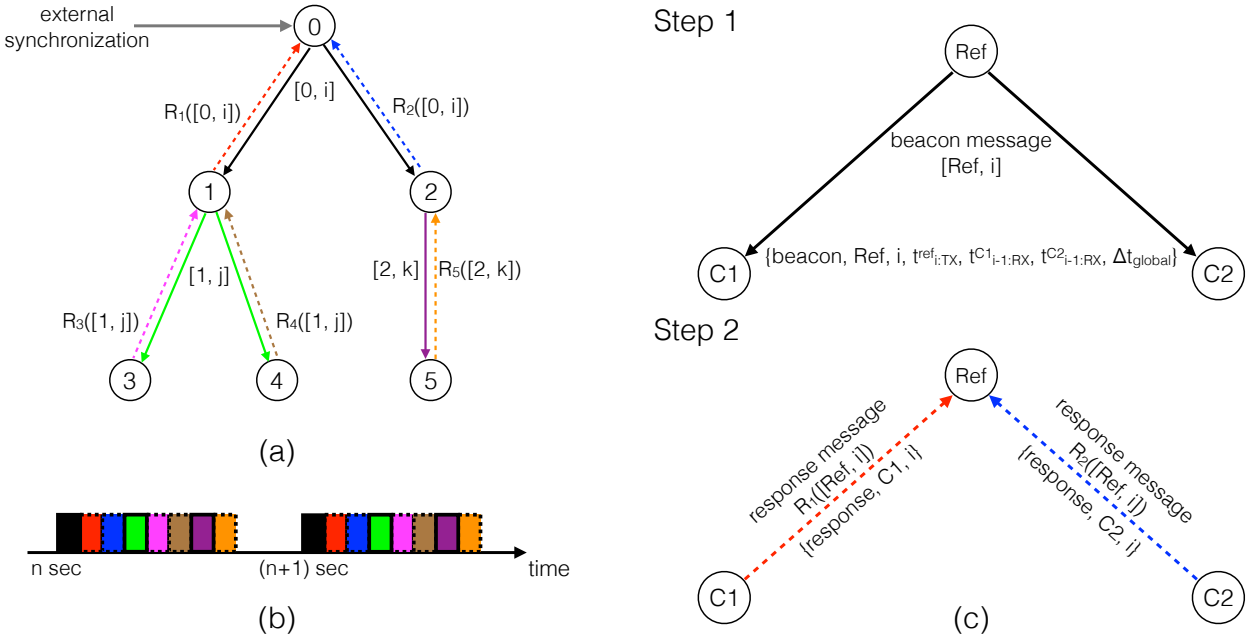


Figure 4.7: Proof-of-concept protocol for clock synchronization. Nodes are arranged in a tree topology (a), a TDMA schedule is generated for communication (b) with the per-hop messaging scheme shown in (c).

4.5 Propagation-Aware Time Synchronization

One of the most challenging aspects of nanosecond synchronization is the need to estimate propagation delay. The protocol described below both estimates range to subtract TOF delay as well as disciplines local clocks.

4.5.1 Messaging with Timestamps

The DW1000 UWB radio on the Pulsar platform provides three time-sensitive messaging functions: (a) transmit as soon as possible and record timestamp, (b) transmit at a deterministic future time and (c) receive and timestamp message. Figure 4.6 describes combinations of these messaging primitives for estimating (and thus allowing for the corrections of) various metrics used in clock synchronization. In our notation, a message i is a beacon message sent by a reference node while $R(i)$ is the response sent by its child.

One-Way Messaging

Described in Figure 4.6 (a), a *reference node* sends messages to a *child node*. These messages could be sent with predetermined transmit times or as soon as possible. One-way messaging is sufficient for propagation-agnostic time synchronization like RBS.

Multiple one-way message timestamps can compute fractional frequencies with respect to a reference node (y_i), which can then be used for frequency locking stationary devices.

$$y_i = \frac{f_i^{\text{child}}}{f_i^{\text{reference}}} = \frac{t''_{\text{i:RX}} - t''_{\text{i-1:RX}}}{t'_{\text{i:TX}} - t'_{\text{i-1:TX}}} \tag{4.3}$$

If the radio clocks were started on a known 1PPS signal edge (using the time-base reset functionality of the radio described in Section 4.4.1), one-way messaging can also be used for propagation-agnostic estimation of the 1PPS line offset between a pair of communicating nodes.

$$\Delta t^i_{\text{PPS:uncorrected}} = \left(t'_{\text{i:TX}} - t''_{\text{i:RX}}\right) \%N \tag{4.4}$$

where N is the number of clock ticks between two 1PPS edges (nominally the clock frequency).

Two-Way Messaging

Two-way messaging as shown in Figure 4.6 (b) requires both *reference nodes* and *child nodes* to transmit and receive messages. The transmit and receive timestamps from frequency-locked nodes are sufficient for stationary inter-node time-of-flight (and hence, range) estimation.

$$\Delta t^i_{\text{TOF}} = \frac{\left(t'_{\text{R(i):RX}} - t'_{\text{i:TX}}\right) - \left(t''_{\text{R(i):TX}} - t''_{\text{i:RX}}\right)}{2} \tag{4.5}$$

The error analysis in two-way messaging is well studied in literature [22]. More messages can be exchanged between the two nodes in a generalized N-way messaging scheme, which then attempts to estimate and compensate for higher moments of clock error.

4.5.2 Timing Tree Construction

Previous work has shown that particular links or certain clocks exhibit abnormally high levels of variance [81]. As shown in Section 4.6, UWB radios exhibit increased and often non-Gaussian error in NLOS configurations. For this reason, it is critical to select links within the network that have low levels of jitter. As part of our network setup, we have a mode that exchanges pair-wise messages between each node in order to capture the link graph and an initial gauge on link variance. We process the graph data using a Sparse version of Full SemiDefinite Programming (SFSDP) relaxation for the Sensor Network Localization Problems package in Matlab which generates a graph structure of the network. Each link on the graph is weighted based on its variance over 100 messages. There are any number of ways to then select a spanning tree across the graph that minimizes cumulative variance. Though not the focus of this work, we show in Section 4.6 that in practice there are cases where minimal hop count, which is often used in PTP, leads to comparatively

poor synchronization. It is worth noting that unlike most wireless link assessment problems, timing variance is easy to compute over a series of message exchanges.

4.5.3 Protocol

For clock synchronization, we propose a proof-of-concept protocol based on PTP that utilizes the messaging schemes described in Section 4.5.1. Our algorithm has the following prerequisites:

1. The clock parameters and relevant regions of the Allan deviation curve for the nodes are used to determine the update rate of the algorithm (τ_{update}).
2. A tree-like time distribution network is created with multiple reference-child relationships as shown in Figure 4.7 (a). The importance of generating a good tree is described in Section 4.5.2.
3. A feasible TDMA schedule has been generated for inter-node communication as shown in Figure 4.7 (b) using the approach similar to [67]. A set of beacon message and its responses are clubbed together to reduce potential errors due to motion, temperature variance, etc.

For simplicity, we assume that each node has a list of relevant communication links, update rates and TDMA slots, and that these do not change during the process of synchronization. The content of each message is shown in Figure 4.7 (c).

The reference node algorithm is as follows:

1. Perform a radio time-base reset on the 1PPS edge.
2. Update $\Delta t_{\text{global}} = \Delta t_{\text{external}} + \Delta t^{\text{ref}}_{\text{global}} + \delta t_{\text{CPO}}$. ($\Delta t_{\text{external}}$ and $\Delta t^{\text{ref}}_{\text{global}}$ are updates from external synchronization and upper-levels of the tree respectively, if applicable)
3. Send a beacon message at time $t^{\text{Ref}}_{\text{i:TX}}$ using the deterministic future send function in the allotted TDMA slot (Step 1 in Figure 4.7 (c)).
4. Listen for responses from child nodes until start of the next update cycle. Record their receive timestamp ($t^{\text{C}j}_{\text{R(i):RX}}$) to be sent in the next beacon message.
5. Go to step 2.

The child node algorithm operates in two stages. The first stage is listen-only and handles frequency estimation and locking. The second stage sends response messages for phase estimation and locking to the reference.

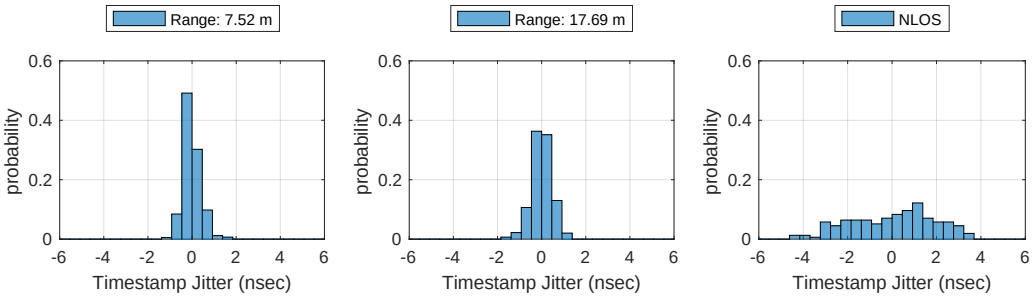


Figure 4.8: Timestamping jitter over various distances and with obstacles.

S1 Frequency estimation and lock

1. Perform a radio time-base reset on the 1PPS edge.
2. Listen for beacon messages from reference node, record the receive timestamp ($t^{\text{Cj}}_{\text{i:RX}}$), extract relevant information from beacon message ($t^{\text{Ref}}_{\text{i:TX}}$, $t^{\text{Ref}}_{\text{R(i):RX}}$), compute y_i using timestamps and pass information to discipline task.
3. If $\bar{\sigma}_y \leq \alpha\sigma_y(\tau_{\text{update}})$, go to step 4, otherwise go to step 2. $\bar{\sigma}_y$ is fractional frequency estimate, $\sigma_y(\tau_{\text{update}})$ is from the Allan deviation curve and α is a predetermined constant.

S2 Phase estimation and lock

5. Send a response message in the allotted TDMA slot and record the transmit timestamp ($t^{\text{Cj}}_{\text{R(i):TX}}$).
6. Estimate $\Delta\bar{t}_{\text{TOF}}$, $\delta\bar{t}_{\text{CPO}}$, $\Delta\bar{t}_{\text{PPS}}$, y_i as described in Section 4.4.2 & Section 4.5.1 and forward these to the clock discipline task if required.
7. Execute phase bootstrap step, if applicable, and go to step 1. Otherwise go to step 2.

One advantage of this approach compared to more sophisticated protocols is that the disciplining only occurs on the child node. The primary drawback of this approach is that clock synchronization errors accumulate per hop, and timing quality degrades with increasing depth. If a node is to function as a relay (both reference and child simultaneously), it would start in child mode and wait for frequency lock ($\bar{\sigma}_y \leq \alpha\sigma_y(\tau_{\text{update}})$) to release its reference task. If phase alignment is also enabled, then we wait until both frequency and phase are within predetermined bounds ($\Delta t_{\text{PPS}} \leq \Delta t_{\text{PPS:threshold}}$, a predetermined value, in addition to the frequency lock condition) before releasing the reference task.

4.5.4 Clock Disciplining

Controlling clock frequency is an essential requirement for phase estimation (and correction) in our protocol. Our current implementation uses a PID feedback loop designed around fractional frequency estimates and

corrections. This is sufficient for frequency correction but not necessarily for phase correction. The discipline task on a child node waits for new y_i estimates and applies the following correction.

$$y_{\text{steer}}^{\text{Cj}} = F_{\text{PID}}\left(y_{err} = y_i, y_{set} = 1, [K_p, K_i, K_d]\right) \tag{4.6}$$

Based on implementations in NTPv4 [58], this could be modified into a hybrid PLL + FLL controller that can correct for both phase and frequency offsets. Since the maximum frequency steer in the CSAC is limited to 2 parts in 10^8, a worst case PPS offset of 1 *sec* would take more than 3 years to correct. Thus, we add a bootstrap step for phase correction.

- If $\Delta t_{\text{PPS}} > \Delta t_{\text{PPS:threshold}}$, (a) set the clock in manual PPS synchronization mode and (b) start a timer clocked off the 10MHz input and set to trigger the PPS_{in} line at the closest earlier clock edge ($\text{count} = \text{floor}\left(\Delta t_{PPS}/\Delta t_{\text{10MHz}}\right)$).

4.6 Evaluation

In this section, we benchmark the timestamping accuracy of our UWB radio, evaluate link quality for spanning tree generation in our testbed and perform a single-hop evaluation of our clock synchronization and phase estimation protocol. All evaluations for Pulsar are carried out on channel 2 (3774 to 4243.2 MHz) of the DW1000 UWB radio with the slowest data rate of 110 $Kbps$, a preamble length of 1024 symbols and a pulse repetition frequency of 64 MHz. These parameters are chosen to focus on known good timestamping performance at the expense of low data rates.

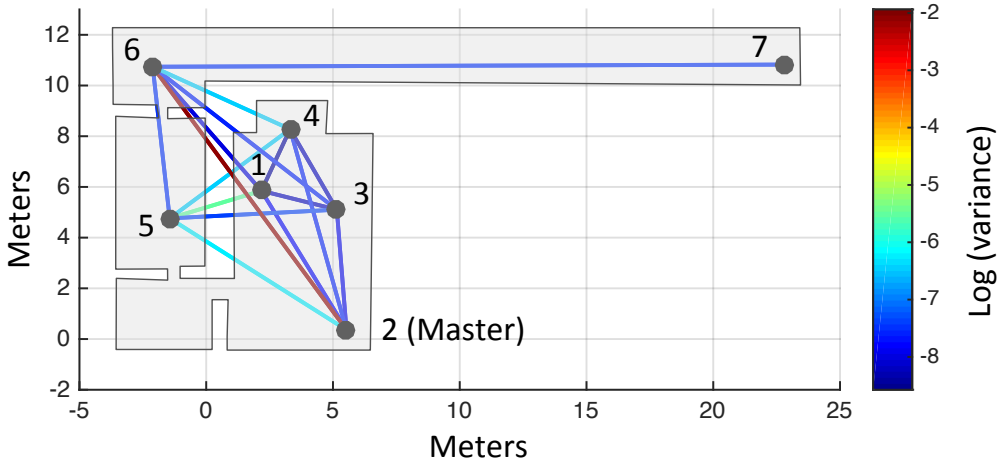


Figure 4.9: Variance in range measurements along edges of testbed network

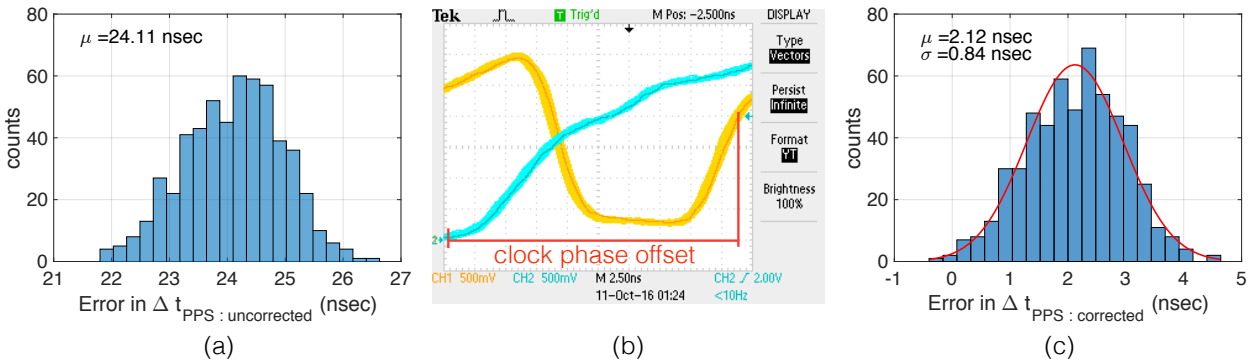


Figure 4.10: Synchronization performance: (a) uncorrected PPS error, (b) phase error between PPS and $38.4MHz$ clock and (c) corrected PPS error.

4.6.1 Timestamping Jitter

Wireless clock synchronization functionality in the Pulsar platform like (frequency + phase) estimation and locking is based on hardware timestamps for message transmission and reception generated by the UWB radio using Decawave's internal algorithm. In order to model our system performance, we first evaluate the quality and consistency of message timestamps to determine their error contribution to our subsequent clock synchronization.

The reference Pulsar node is used in transmit-only mode with a previously frequency-calibrated child node in always-listen mode. Beacon messages are periodically sent by the reference node containing embedded transmit timestamps ($t'_{\text{i:TX}}$) using the deterministic delayed transmission functionality of the UWB radio. Receive timestamps ($t''_{\text{i:RX}}$) are computed on the child node on successful reception. We evaluate the spread of $t''_{\text{i:RX}} - t'_{\text{i:TX}}$ to estimate timestamp jitter in the combined system. Figure 4.8 shows the probability-normalized distribution of timestamps for a pair of static nodes separated by various distances and in different configurations. Differing distances do not drastically affect timestamping accuracy ($\sigma_{7.52m} = 0.40\ nsec$ & $\sigma_{17.69m} = 0.45\ nsec$) which stays accurate to under 0.5 $nsec$. However, timestamping consistency drops significantly once received signal quality goes below a minimum signal energy threshold due to non-line-of-sight or the nodes being too far away. We thus determine that it is essential to identify and prune poor links in the clock synchronization network. Channel metrics such as first-path power and received signal power may be used to identify NLOS links.

4.6.2 Timing Tree

In order to evaluate network synchronization, we tested our ranging protocol on a seven-hop network deployed across a 300 square meter area of a building floor. Figure 4.9 shows the node locations with communication

links that are colored based on the variance of ranging jitter. It is worth noting a few interesting links. The topology was generated using a semidefinite programming graph realization solver with an error in 3D of less than 0.1 m per node as compared to laser ground truth. We manually aligned the node cluster to fit the map. First, the connection between node 2 (master) and 6 has abnormally high variance. Many synchronization protocols would use minimal hop-count as the primary metric for picking a path. In this case, that would lead to higher synchronization error for nodes 6 ($\sigma = 0.34\ m$) and 7 ($\sigma = 0.34\ m$). Instead, a variance-based routing scheme may pick node 1 as a relay for 6 ($\sigma = 0.03 + \epsilon_{\text{hop}}\ m$) and 7 ($\sigma = 0.04 + \epsilon_{\text{hop}}\ m$). In this chapter, we present a proof-of-concept algorithm that demonstrates that our platform is capable of nanosecond-level synchronization. One could imagine multiple approaches for optimizing both time distribution as well as applying more sophisticated synchronization algorithms that benefit from group coordination.

4.6.3 Clock Synchronization

Finally, we evaluate the clock synchronization protocol and disciplining functionality of our platform. The objective is to first achieve wireless frequency lock between two Pulsar nodes and then estimate the offset between their 1PPS signal line. We run our experiment on two nodes that are separated by approximately 3.6 m with Line-of-Sight in a reference-child topology. The child node runs a frequency discipline feedback loop with $K_p = 0.1$, $K_i = 0.1$ and $K_d = 25$, which stabilizes the fractional frequency for $\alpha = 2$ as described in Section 4.5.3. Since we focus on 1PPS line delay estimation, we set the protocol and discipline loop update rate as $\tau_{\text{update}} = 1sec$.

Ground truth is collected by connecting the 1PPS lines from both Pulsar nodes to a Saele logic pro analyzer that is digitally sampling both lines at 500 MSps over a period of 10 minutes. Since the logic analyzer was not calibrated to the same accuracy as the reference CSAC through GPS, we apply a correction to all time measurements based on the time difference between subsequent 1PPS edges of the reference node.

$$\frac{\Delta t_{\text{corrected}}}{\Delta t_{\text{measured}}} = \frac{\Delta T_{\text{1s:Ref}}}{\Delta T_{\text{1s:analyzer}}}$$

The error between ground truth and our uncorrected PPS offset estimates are shown in Figure 4.10 (a). This estimate is based on the raw transmit and receive timestamps for the beacon message and does not consider the contribution of time-of-flight, clock-phase-offset and other errors in the system. Message passing as implemented by the protocol in Section 4.5.3 helps us determine the time-of-flight between the two nodes. The missing clock phase offset parameter is measured off-board on an oscilloscope or PMU once

per run. Figure 4.10 (b) shows the phase offset between the 1PPS line and 38.4 MHz frequency line to be stable within 1 $nsec$ of variation over a period of 30 minutes.

All of the individual components of error can then be compensated for to achieve the distribution in Figure 4.10 (c) which shows our final synchronization accuracy to be better than 5 $nsec$. The mean ($\mu = 2.12nsec$) and variance ($\sigma = 0.84nsec$) achieved are within the error bounds expected from various components such as the ground truth measurement on the Saelae logic analyzer, the timestamping inaccuracy introduced by the DW1000 chip, the jitter introduced by the PLL and the frequency errors in our clock source.

4.7 Summary

This chapter presented Pulsar, a clock synchronization platform for wireless clock synchronization of indoor devices. The platform combines UWB ranging radios with a stable CSAC timing source that improves upon the state-of-the-art in terms of accuracy. UWB radios are used to estimate TOF ranges between nodes such that the speed of light delays can be estimated and accounted for as part of the synchronization protocol. The CSAC provides long-term stability on the order of 1 μS of pairwise drift per 1.2 days and directly clocks the radio and a PPS output system to provide phase aligned PPS and 10 MHz output signals. We show that Pulsar provides better than 5 ns synchronization per hop across a network. We also develop and evaluate a synchronization protocol that highlights how the physical link topology can play an important role in synchronization. As technology evolves, precise clocks and accurate timestamping hardware will become more accessible over time and would even be included in consumer devices (e.g. Apple's iPhone 11 is equipped with a UWB radio [12]). Pulsar paves a way to precisely synchronize communication devices like LPWAN gateways in currently unreachable indoor locations.

Chapter 5

Channel fingerprints using synchronized distributed reception

We now have a distributed receiver that can accurately synchronize the capture of the radio channel state, so let us explore how we can use this system to localize our transmitting devices. Localization of low-power devices over wide areas is a feature demanded by many users. Such a feature could be used to locate packages, track wildlife, locate the bikes in a bike share, etc. However, it is challenging to developing a system which can provide the necessary accuracy over a large area while keeping the tracked devices low-power, low cost and compact.

Traditional solutions for localization like GPS can provide sufficient accuracy, but are power-hungry, expensive and primarily work outdoors with an open sky view. These solutions are too restrictive for modern IoT deployments that are heavily cost constrained, resource constrained or reside indoors. This has given rise to schemes based on beacon-proximity, time-of-flight (ToF) and time-difference-of-arrival (TDoA) of wireless messages. Beacon-proximity schemes require a large infrastructure of low-range RF beacons, which is difficult for large-scale deployment.

Gateways play an important role in the RF localization of end nodes, particularly for timestamping wireless messages for ToF and TDoA schemes. However, current LoRaWAN hardware and software implementations limit timestamp resolution to 1 μsec. This limits localization accuracy to worse than 300 m, which is often not sufficient for most applications. However, there is some opportunity with the use of varied frequency bands and the use of synchronized SDR-like gateways.

5.1 Contributions

- Through a study of radio localization methods and ray-tracing simulations, we determine that channel fingerprinting methods with time-difference-of-arrival features would be best suited for the constraints of LPWAN device localization.
- An FFT resampling technique that extracts timestamps and other features from a narrowband signal at a granularity much smaller than $1/bw$.
- Evaluation of the stability of channel features at different locations and different times through campus-scale experiments.

5.2 Motivation and Approach for Using Channel Fingerprints

There are a number schemes based on signal strength, time-of-flight (ToF) and time-difference-of-arrival (TDoA) of wireless messages for the purpose of localizing wireless devices. However, each of these methods has various tradeoffs and not all of these are ideal for LPWANs in urban scenarios. We want our localization system to be sufficiently accurate in a wide-area scenario with lots of multipath reflections. It must still retain the scalability and low-power advantages of an LPWAN system. The architecture of TDoA is excellent under these constraints, as a simple unsynchronized device can be localized by timing simultaneous receptions at different receivers. However, the TDoA localization algorithm is dependent on the assumption of predictable RF baths between the transmitter and receivers.

Urban environments are extremely dynamic and unpredictable for radio propagation, where such an assumption easily breaks down. The primary challenge in such a setting is multipath propagation. Multipath is a term used to describe any radio propagation paths caused by reflections, diffractions and propagation through materials. Channel response is a mathematical abstraction that is used to describe the delays, attenuation and other physical effects an RF signal undergoes as it propagates. Various commonly used features (e.g. signal strength, ToF) can be easily extracted from channel response. Additionally, channel response also captures the effects of multipath propagation. We use ray-tracing simulations to simulate channel response and understand the challenges posed by multipath. We look at the following questions:

- How significant is the multipath in urban scenarios?
- Can we make any estimates or generalizations about the multipath in an area?
- Is multipath dependent on frequency, and does it change predictably?

Through a number of simulations in urban and rural environments like the one we explore in Section 5.9.1, we see that urban environments with lots of structures (unsurprisingly) cause significant multipath. The scarcity of line-of-sight paths leads to an interesting insight: improving our ranging capabilities would not necessarily better our localization accuracy. Unfortunately, multipath is a very localized phenomenon in urban environments, with the paths sometimes changing drastically within a few meters. In rural scenarios with large open areas, we observe more predictable behavior like the presence of Fresnel zones. Though the actual paths taken by radio waves are not frequency dependent, their intensities are. Thus, the channel response we capture also ends up being unpredictable and localized. Generally, these simulations tell us that any simple physics-based methods are likely to be error-prone, and we must resort to radio fingerprinting techniques.

There are a number of different ways we could generate fingerprints to aid localization. Received signal strength is the most commonly used method, but we already know it is inaccurate and unstable for the large scales of LPWANs. Another commonly used channel feature is the time difference between the largest peaks in the channel response. However, we have found that it can be useful to consider some of the secondary features as they can provide some additional information about the propagation path. In this work, we will look at channel response as a fingerprint, as it can provide a richer representation of the various propagation paths.

A receive-only architecture, like that used in TDoA, is ideal for the constraints of LPWANs as it allows the low-power client devices to be simple as they are unsynchronized with the rest of the network. However, one of the major challenges in creating reliable channel features using unsynchronized LPWAN waveforms is the synchronization between all the gateway receivers. On further exploration of the clocks involved, we conclude that suffering a phase offset is inevitable given the limitations of current radio hardware. Thus, we generate channel response features using a complex matched filter that is agnostic to the receivers' phase offset. This entire process is almost completely unidirectional, which helps scalability.

Matched filtering on a narrowband LPWAN signal has limited capability to resolve the channel response of individual paths whose path differences are less than hundreds of meters. However, we do observe different responses at different frequencies as the intensities of every individual signal path changes significantly across frequency bands. Thus, in our system, we propose the use of multiple frequency bands – particularly, a combination of 433 MHz and 915 MHz ISM bands. As TV whitespaces are opening up for unlicensed use [30], we have the option of using a wider range of frequency bands to gather a richer set of channel features. We envision our gateways to be able to service at least two frequency bands simultaneously – one would behave like a regular gateway, while the other one can dynamically tune to the frequency bands where transmissions for fingerprinting are taking place.

A fingerprinting system requires a large database of observations. We believe a crowdsourcing approach could work along with the use of city buses, delivery trucks, etc., which are equipped with the necessary hardware and secondary localization equipment.

In this work, we focus on and evaluate the stability of channel features at different locations, which is a first step in developing a full fingerprinting system. The performance of a fingerprint-based localization system is dependent on the locality of the features being captured. This is still an open question. A comparison with time-difference-of-arrival systems is also left as future work.

5.3 Related Work

In this section, we present other methods that attempt to provide accurate location through the use of auxiliary systems or using LPWAN signals themselves.

5.3.1 LPWANs with Auxiliary Localization Systems

There are a number of LPWAN solutions that rely on a secondary localization technology to provide a location that is then communicated over an LPWAN network. OpenChirp, described in Section 2.4, has a GPS service that will take in strings provided by GPS modules and automatically map them for visualization. However, running a GPS receiver is a large drain on the battery life of a device. Other solutions such as TrackTag [15] reduce the power consumption of a receiver by just recording unprocessed GPS signals and analyzing them afterwards, once the device is retrieved. This is, of course, unsuitable for devices which will be deployed out in the field for a long period of time.

5.3.2 Localization and Ranging Using LPWAN Signals

It would be ideal if we could use the signals used by our communication system for localizing devices. In this spirit, [94] describes a wildlife tracking system that localizes simple 433 MHz FSK transmitters over a wide rural area. The LoRa alliance itself has also been developing techniques that would use the timestamping capabilities in existing gateways augmented with observations about WiFi beacons [19, 29]. Though promising, it still relies on a secondary radio for precise localization. [61] present a localization system using LoRa backscatter communication. However, this system shows limited range due to reliance on backscatter and would have difficulty supporting more than one backscatter device on such a network.

5.3.3 Fingerprints for Localization

Previous work has also explored the use of a variety of fingerprints for the purpose of localization, mostly in indoor scenarios. [9, 10, 41] explore the use of different techniques using WiFi RSS to identify locations. [32] expands on this further with the use of SLAM to not only localize the WiFi client devices, but also the access points as the devices move around the environment. Bluetooth consumes significantly less power than WiFi, and we have seen a number of similar systems using a large number of low-power, low-cost BLE beacons [33, 71]. Similarly, ToF fingerprints can be used with a number of radio technologies like UWB [36] and even for acoustic signals [87]. Other work has explored the use of RSS fingerprints [8] or TDoA fingerprints [62] for improving localization indoors.

Unfortunately, many of these approaches operate at short distances and would need a vast infrastructure deployment of beacons or access points to be able to operate at city-scales. Thus, existing work has also explored the use of cellular GSM signals [90] for the purposes of localization. [68, 3] explore the use of LPWAN RSS signal fingerprints, and some works go further with the use of machine learning techniques for improved accuracy [44]. In our work, we will explore the potential of using the complete channel response, which provides much richer information.

5.4 Radio Localization

In this section, we will explore various techniques used for radio localization and find the paradigm that is best suited for localizing low-power devices over very wide areas in an urban environment. Note that wireless ranging, i.e. estimating the one-dimensional distance between two devices, is only a sub-problem of localization where we estimate the full three-dimensional location.

All of the methods described below can be used in two different directions:

1. An LPWAN client device can observe transmission from various gateways to locate itself.

2. LPWAN gateways can observe the transmissions from a client device to locate the client.

Method 2 is more suitable for LPWANs because of its focus on low-power, low-cost client devices supported by powerful gateways. Additionally, the more common LPWAN use case is for the network infrastructure to locate the devices rather than for the devices to locate themselves. Thus, even if we were to localize devices using method 1, we would have to send an extra uplink message to inform the network about the devices' locations.

5.4.1 Received Signal Strength

Estimating the range of a transmitter based on the power of the received signal strength (RSS) is one of the simplest and oldest methods used for radio localization. Physics determines that the intensity (and correspondingly the received power) of an electromagnetic wave decreases with the inverse square of the distance traversed (or by a power close to -2). The location of a transmitter can be estimated from the observed power of the received signal at known locations and by solving a multilateration problem (finding the intersection point of multiple spheres).

RSS outputs are already available on most radio receivers, and this paradigm can work with unidirectional transmissions. Unfortunately, RSS is extremely sensitive to obstacles, and any changes in the environment lead to inaccuracies of thousands of meters [54]. In Figure 2.6, we could already see the effects of RSS warping around buildings and other terrain features. In addition, RSS measurements vary significantly over time, and it becomes inaccurate at the long distances traversed by LPWAN transmissions, particularly in urban multipath-prone scenarios.

5.4.2 Time of Flight

Time of flight (ToF) uses propagation time observations to estimate the distance traversed by a radio signal, since the speed of light is constant (or varies little in the air). To know propagation time, we must measure both the transmission and reception time of a signal accurately. Multiple ToF measurements at known locations are used to formulate a multilateration problem.

ToF is a significantly more stable and reliable measurement compared to RSS. It is, however, still susceptible to errors from multipath where the actual propagation path gets elongated from reflections, material transmission and diffraction. It has had successful use with ultra wideband ranging systems with the capability to separate the different paths in an observed signal [21, 38]. The greater downside of ToF is the requirement to measure both the transmit and receive times accurately at a precision of nanoseconds. Accurate synchronization and the use of power-hungry expensive clocks is necessary even on low-power client devices. Precise synchronization typically requires the use of bi-directional communication, which significantly reduces scalability.

Phase-based methods are a special form of ToF methods. They look at the phase change of a signal to get more precision in the estimation of propagated distance. In the context of LPWANs, they have similar advantages and limitations to ToF.

5.4.3 Time Difference of Arrival

In time difference of arrival (TDoA) methods, we measure the time difference between the reception of the same signal at multiple locations. This method trades off the need to synchronize the transmitting device for a greater number of observations of the receive time. The receivers must be accurately synchronized, but this is achievable on gateway-class hardware. Multiple observations can again be used to formulate a different multilateration problem (one which solves the intersection of hyperboloids rather than the intersection of spheres).

The greatest advantage of TDoA over ToA is the ability to localize an unsynchronized transmitter through unidirectional communication. Thus, a device can use a low-cost, low-power clock and not worry about synchronization. The relative location of the receiving gateways is very important in TDoA multilateration and is explored using the concept of geometric dilution of precision (GDOP) [89]. TDoA is also susceptible to errors from multipath propagation, and the resulting error in localization might be exacerbated due to bad GDOP.

In analogy with phase-based methods and ToF, phase-difference methods are a special form of TDoA that can provide additional granularity in measurements.

5.4.4 Direction of Arrival

A well-arranged geometry of receivers can estimate the direction along which a transmission was received using measured differences in phase and time. This geometric arrangement is often packaged together to create a single receiver unit. A single direction of arrival gateway would have multiple antennas arranged in a linear array or grid. We can use multiple direction of arrival (DoA) measurements from several such gateways to estimate location through techniques like ray-tracing.

DoA has the potential to identify and separate out individual multipath if they differ enough. However, DoA hardware is significantly larger in size and complexity. Good antenna arrays are typically the same scale as the wavelength of the signal (33 cm for 900 MHz, 70 cm for 433 MHz), and each receiving element either needs its own RF frontend or a method to multiplex between them quickly [60, 59]. Due to the many advantages, we can foresee DoA gateway deployment done by an organization, but these would be too large, expensive and complex for unplanned user-deployed gateways.

5.4.5 Fingerprint Methods

All the above methods attempt to find a closed form solution based on the physics of radio propagation. Physics-based methods can be effective in line-of-sight (LoS) scenarios like space, short-range proximity,

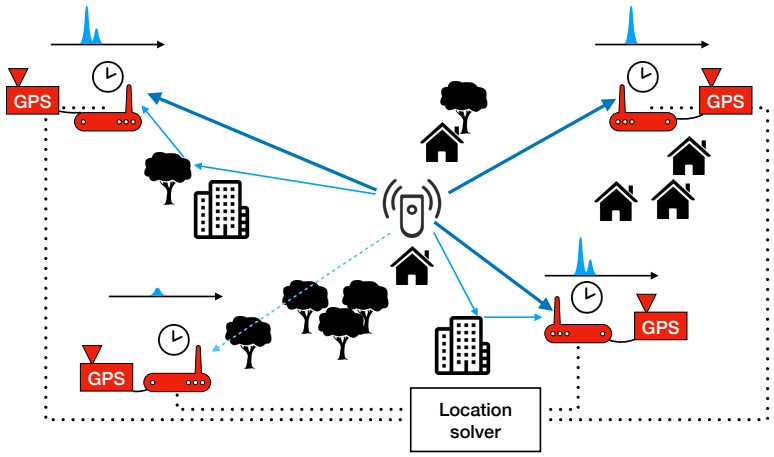


Figure 5.1: Overview of location fingerprinting system

flat land, etc. However, multipath propagation starts to significantly degrade the accuracy of localization as the physics for radio propagation is different with reflection, transmission and diffractions. This can be particularly problematic when the LoS path is completely blocked. To elaborate on how common multipath occurs in urban scenarios, consider FM radios; it's very rare to see an FM transmission tower in urban scenarios, but the coverage is almost ubiquitous because of multipath propagation.

Fingerprinting methods, on the other hand, depend on the repeatability of measurements – a much simpler assumption to satisfy in multipath-prone environments. Fortunately, many of the features we used for the physics-based methods in previous sections could also be used in a fingerprinting system. In a fingerprinting method, we collect a large number of observations in the deployment area (with ground truth location provided by another localization system) and localize our devices based on the closeness of features to existing observation points. Location estimation can then be refined using physics-based methods but constrained to a smaller search area. The accuracy of fingerprinting systems is dependent on the locality of the features, the understanding of which is unfortunately still an open problem.

In summary, using channel features from gateways receiving the same transmission is fitting for LPWANs due to energy consumption, cost of devices and scalability.

5.5 System Overview

Figure 5.1 provides an overview of our location fingerprinting system. Signals captured at different frequency bands provide much richer information about the channel.

Fingerprinting Mode: We introduce a fingerprinting mode that uses regular LoRa transmissions on client devices. If a client can support multiple frequency bands, it can use this fingerprinting mode by first

sending an initial transmission on the primary ISM band that notifies the surrounding gateways about its fingerprinting operation. The decision to fingerprint could be made by the client itself, or the gateway could request this operation during one of its previous exchanges. Following this initial transmission, the client will subsequently transmit on the different frequency bands available to it, with the transmissions occurring at predetermined time differences from the initial transmission (the gateways can be informed about these time offsets through the initial transmission). We believe that our fingerprinting mode will only be used occasionally as compared to regular communication, so it would not overwhelm the available spectrum. Many of these operations can be built on top of existing LoRaWAN operations.

Signal Processing at the Gateway: To capture rich channel information, gateways need the ability to listen to a wide variety of frequency bands. For this purpose, we envision future gateways to have at least two frontends capable of receiving simultaneously. One of the frontends continuously listens to the main ISM frequency band and behaves like a regular gateway. The other frontend will hop across different frequency bands whenever a fingerprinting transmission is expected. This can be determined by looking at the initial transmission from the client. There are a number of low-cost RF ICs, like the AD9363, that can perform this function of simultaneously receiving multiple frequency bands. Channel response (and subsequently channel fingerprints) can then be estimated from these captured signals by processing them as described in Section 5.7.

5.6 Precise Synchronization of LPWAN Gateways

To understand the appropriate processing algorithm to use for extracting channel features, we must first look at the time synchronization challenges in the system. Synchronizing receivers across large distances is difficult because of the variety of clock domains present on the hardware of a gateway. In this section we'll take a closer look at these synchronization challenges.

A modern software radio receiver (and transmitter) has multiple clocks and oscillators for different operations. The heart of software radio systems are the analog-to-digital converter (ADC) and an RF mixer. The mixer converts a high frequency radio signal into baseband that can then be converted into a digital value using the ADC. Once the signal is converted into digital, it temporarily stays on the ADC's local buffer until it is either discarded or copied over to the host computer's buffer.

A number of timing problems show up while the signal traverses this path. For simplicity, say the same exact signal is being processed on a number of different receivers and our objective is to get the exact same output on each receiver. A frequency and time reference (synchronized by GPS) is a shared reference between all these devices.

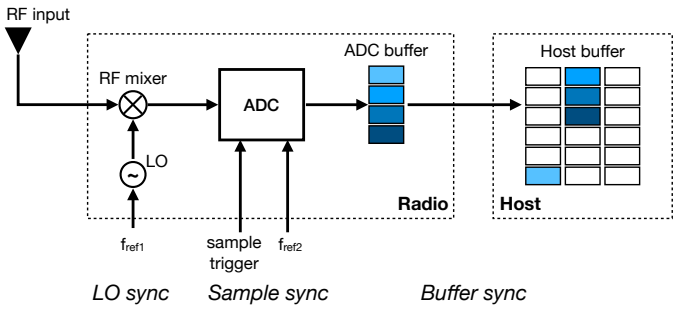


Figure 5.2: Illustration of clock systems in a software radio and associated synchronization problems

LO Synchronization: The mixers on each receiver will shift the high frequency radio signal into a low frequency baseband signal. A reference sinusoidal waveform is provided by the local oscillator (LO). The reference waveform is ideally at the same frequency as the carrier frequency (up to 928 MHz for ISM band) of the transmission. Unless the reference waveforms are exactly identical on all the receivers, we would induce phase and frequency errors in the baseband. The problem of synchronizing these reference sinusoids is called the LO sync problem. Existing PLLs can accurately set the frequency of the LO, but correcting or even measuring the phase of the LO is a much harder task without available hardware solutions. Some radios will have an additional conversion step to an intermediate lower frequency that is easier to synchronize using existing hardware.

Sample Synchronization: Once the signal is in baseband, the ADC will sample it at a rate provided by another digital clock. ADCs sample signals at particular instances of time. The same signal will look different on two different receivers if captured at different instances of time, even when the sampling rate was identical. Even if the digital clock frequencies across different receivers are exactly identical, we will capture a different signal if we sample it at different time offsets. This is known as the sample synchronization problem. Fortunately, many available ADC units and RF frontend units can be programmed to sample at the exact instance using a hardware trigger input (typically controlled by a MCU or FPGA that is also running off the same reference clock).

Buffer Synchronization: After passing through, the ADC is digitized and stored on the ADC buffer. The host computer reads the buffer value and associates these samples with the exact time they were sampled. Many low-cost SDRs (e.g. RTL-SDR) only have a continuous buffer that we can read from without any references to the sampling time. Samples could thus be offset by entire sampling intervals. This is called the buffer synchronization problem. Fortunately, more advanced SDRs will populate their buffers in a predictable manner after a sampling trigger signal is provided.

In summary, we run into three major synchronization issues on a software radio: LO sync, sample sync and buffer sync. Sample sync and buffer sync can be solved by the appropriate combination of software and

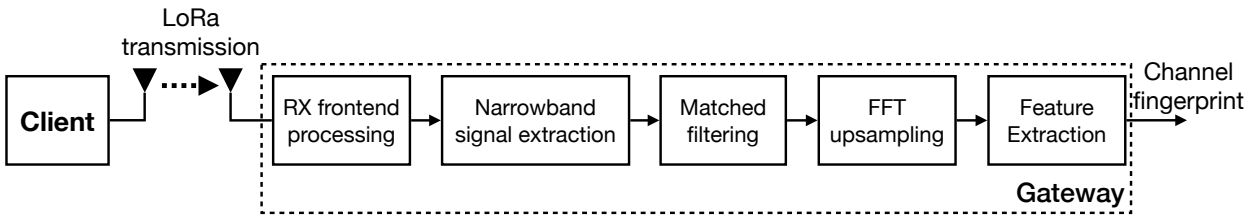


Figure 5.3: The processing pipeline to estimate channel response

hardware. LO sync for phase is still an open problem. We elect to work on localization techniques that are agnostic to this phase offset. Note that some systems attempt to solve the LO sync problem using line-of-sight reference beacons. However, at large distances, these systems become impractical as it is difficult to place beacons that can have LoS to all the synchronized gateways. In our simulation model of downtown Boston, we could find no such location that had a direct view to all gateways.

5.7 Estimating Channel Response

In this section we discuss channel response estimation. Channel response is the basis for generating channel features. Figure 5.3 shows the processing pipeline used to extract channel response. We use multiple gateways that are tightly synchronized using GPS or a similar system like Pulsar. They also have RF frontends that could support multiple frequency bands. For each frequency band that is being used, each gateway performs the following operations as shown in the processing pipeline.

Next we'll describe some of the components of this pipeline.

5.7.1 Isolating Narrowband Signals

Most LPWAN gateways are designed to simultaneously receive a number of neighboring frequency bands. In our pipeline, we want to isolate the transmissions in individual frequency channels before processing them further. Thus, before we begin matched filtering, we have to first isolate the signal in a given band and shift it to baseband. This can be achieved by taking a window of samples, applying an FFT and selecting only the frequency bins where our narrowband signal is present. For a signal captured (with sampling time T_s and having n samples) at center frequency f_c, we can find the index of the lowest and highest FFT bucket:

$$i_{low} \leq nT_s(f_{high} - f_c)$$

$$i_{high} \geq nT_s(f_{high} - f_c)$$

Once we isolate the required frequency band, we can move the FFT response to zero and apply an inverse FFT to get the baseband version of the signal. Note that the sampling frequency of this signal is now

$$T_s' = T_s \times \frac{n_{FFT}}{n'_{FFT}}$$

Since the indices of the buckets are integers and unlikely to exactly match the upper and lower frequencies, we will be left with a frequency offset of $\delta f = f_{low} - f_c - i_{low}/(nT_s)$ at the lowest bucket.

5.7.2 Matched Filtering

The matched filter is a waveform-specific FIR filter that will maximize the SNR of a received signal if it contains the correct waveform. Since the filter is designed for a particular waveform, it is extremely good at rejecting noise and interfering signals. Matched filters are proven to be the ideal for maximizing SNR under AWGN conditions [76] and is thus apt for finding weak signals originating from distant transmitters in a noisy environment. Another advantage is that a matched filter requires no a priori time synchronization, which can be a drawback with some windowed filters. Finally, the magnitude of the matched filter response is unaffected by the LO synchronization we discussed in Section 5.6.

For a waveform $x(t)$, its matched filter which maximizes the response at a time T_m (scaled by an arbitrary constant α) is given by:

$$h(t) = \alpha x^*(T_M - t)$$

We will primarily target the upchirps and downchirps in the LoRa packet preamble and synchronization. The waveform of a linear frequency-modulated upchirp with bandwidth β and width tau in baseband is given below (and can be used in the expression above):

$$x_{upchirp}(t) = \exp\left(j\pi\beta t^2/\tau\right)$$

Stretch processing is a special matched filtering technique for linear chirp waveforms. It involves multiplying a received chirp with its reversed version, and the result is a sinusoidal signal whose frequency indicates the delay between the received upchirp and the reference downchirp. This is computationally easier than a full matched filter convolution operation. However, it operates on windows of data. We can run into issues if the waveform crosses the window border. Thus, for a continuous stream of signals, we have to approximately know where the received chirp begins. This is acceptable for systems like radars that know

the transmit time, but not for LPWAN. Additionally, the reference chirps must be larger (in both time and bandwidth) than the original chirps to ensure there is complete overlap if the chirps are delayed or offset in frequency [76]. It is a tradeoff, as a larger reference chirp is susceptible to noise and interference from other neighboring frequencies. As we've seen in the spectrogram in Section 2.4.1, a LoRa packet consists of multiple successive chirps that would activate the wider reference chirp multiple times and at different amplitudes depending on the amount of overlap. To avoid many of these pitfalls, we will use the standard form of matched filtering in this work.

5.7.3 Ambiguity Functions

An ambiguity function of a waveform is the result of the waveform convolved with its matched filter. Comparing the output of a received signal that has been passed through a matched filter and then comparing it with the ideal ambiguity function can provide hints about signal propagation, e.g. presence of multipath.

The complex ambiguity function of linear upchirp (used in the LoRa preamble) is given by:

$$\hat{A}(t, F_d) = \exp\left(j\pi F_d t\right)\left(1 - \frac{|t|}{\tau}\right)\frac{\sin\left[\pi\left(F_d + \beta t\right)\left(1 - |t|/\tau\right)\right]}{\pi\left(F_d + \beta t\right)\left(1 - |t|/\tau\right)} \qquad t \in [-\tau, \tau]$$

The ambiguity function that is commonly used in radar literature is the magnitude of the above expression and is given by:

$$A(t, F_d) = \left(1 - \frac{|t|}{\tau}\right)\left|\frac{\sin\left[\pi\left(F_d + \beta t\right)\left(1 - |t|/\tau\right)\right]}{\pi\left(F_d + \beta t\right)\left(1 - |t|/\tau\right)}\right| \qquad t \in [-\tau, \tau]$$

Analyzing the ambiguity function helps us understand some interesting properties about the output of matched filters. First, a matched filter can be used to identify the peak of the waveform even in the presence of phase errors, such as those caused by LO sync. Second, a frequency offset between the signal and matched filter results into a predictable delay (or advancement) in the output. If we are working in a TDoA-like system using well-synchronized gateways, this delay would be the same on all the receiving gateways and would cancel out if we only look at the difference between the responses.

Figure 5.4 shows the ambiguity function of the chirps used by SF7 and SF10 at various LoRa bandwidths. The ambiguity function also tells us that matched filters would not be able to distinguish multipath causes due to paths differing by only a few meters. However, they should still be able to lock onto the result of all the responses accurately. Note that the bandwidth of the ambiguity function is the same as the original signal.

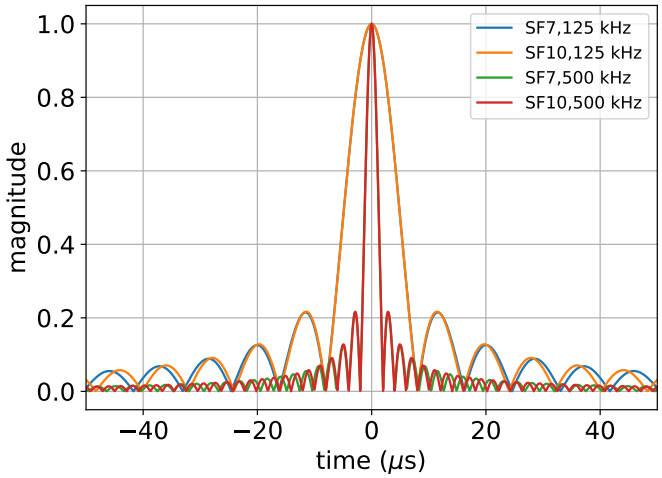


Figure 5.4: Ambiguity function of a linear chirp

5.7.4 FFT Upsampling

It is unlikely that the received signal, once filtered, would have its response peak at exactly one of our sampling instances. However, as we are working with a narrowband signal, we can use FFT resampling to better estimate the location of the signal peak and any other features like nulls, secondary peaks, etc. FFT resampling lets us interpolate between samples in the signal without affecting its frequency response. We can estimate the sample at any arbitrary time t using a large window of samples surrounding it. The larger the waveform being used, the larger these windows can be.

$$x(t) = \sum_i x_i \times \text{sinc}\left(\pi \frac{t - t_i}{T_s}\right) \tag{5.1}$$

5.7.5 Narrowband Response from Ray-Tracing Simulations

Multipath propagation is a complex phenomenon that is dependent on a large variety of variables. RF propagation simulations are essential to understanding the problems caused by multipath in a channel fingerprinting system. Due to advances in RF modeling and parallel computation, ray-tracing currently provides some of the most reliable and efficient RF simulations. We believe that ray-tracing simulations would play a key role not just in the design of fingerprint-based systems, but also in their operation as we start incorporating advanced techniques like machine learning to extract relevant features. However, the channel response outputs produced by most ray-tracing software (they produce a list of delta functions, which would use infinite bandwidth) must be modified to model the response of the narrowband response we observe in practice.

This process is the opposite of the one described in the previous section. Ray-tracing simulations will provide channel response outputs as a list of delay values and the complex signal associated with each delay (t_i, a_i). For analysis, we want to understand the shape of a narrowband signal and its filtered response. The narrowband response (with bandwidth $bw = 1/T_s$) of this channel response output can also be estimated using equation 5.1 but not the x_is and t_i (the signal and time entries in the channel response). We can perform a similar analysis using ambiguity functions to understand matched filter response. For example, the narrowband response of a chirp would be calculated as:

$$x(t) = \sum_i x_i \times A(t - t_i)$$

These resampling techniques are used to generate the channel response in Figure 5.5.

5.8 Implementation

We use a USRP N210 software radio with an SBX-40 daughter-card as our LPWAN gateway device. This daughter-card has two receive antenna inputs. We use a 3 dBi, 433 MHz and 915 MHz antenna on each input. The SDR is synchronized by a GPS disciplined oscillator (GPSDO) that outputs a 10 MHz frequency reference and a 1PPS time reference. The GPSDO synchronizes its internal OCXO with the received GPS signal which means that the SDR gateway is synchronized to actual GPS time. If we were to run in an indoor setup, we could use the Pulsar platform from Section 4.4. A Raspberry Pi 4 (RPi) is our host computer and signal processor, and though there isn't a simple way to synchronize it with GPS, we synchronize it over NTP. We rely on NTP time being within a few milliseconds of GPS time for approximately synchronizing the RPi and gateway.

The actual synchronization process on the gateway is straightforward. First, we wait for our reference clock (GPSDO or Pulsar) to be locked. We consider the time on the SDR clock to be our master reference time for the entire gateway system. The host computer only sends sampling commands to the SDR using timed commands (commands which take place at a fixed time in the future) to avoid sample and buffer sync problems. We do not worry about LO sync beyond the frequency sync provided by the USRP software.

To start a capture, the RPi sends a command (with regards to SDR time) with information about sampling rate and capture start time. The SDR can be configured to then run continuously or stop after a predetermined number of samples. On sending a capture request, the SDR will first start up its PLLs and lock on to the desired frequency before the capture start time occurs.

Though our current prototype uses a commercial SDR, we've already seen in Section 3.6 how most LPWAN gateways have hardware that is similar to a software radio. Many gateway designs are even using

the same RF ICs and designs created for software radios (e.g. AD9363). We envision future LPWAN gateways to look identical to SDR platforms.

5.9 Evaluation

We will evaluate our hypothesis through a combination of EM propagation simulations and a real deployment on the Carnegie Mellon University campus. For simulations, we use Remcom's Wireless InSite EM propagation software [75]. Our real-world experiments use two USRP N210 software radios as gateways. They are synchronized using individual GPS-disciplined OCXOs. We use the SX1262 development kit with an STM32 MCU transmitting 10 dBm LoRa frames at SF10 as our client device. Both the client devices and gateways are equipped with switchable 915 and 433 MHz antennas.

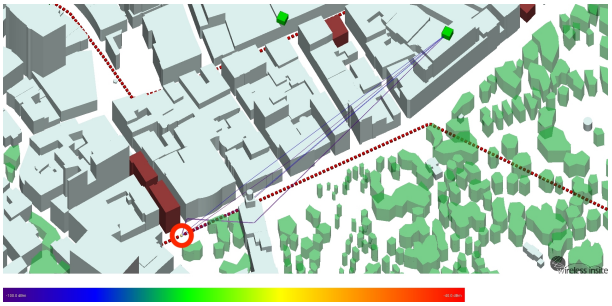
5.9.1 Channel Response Simulations at Different Frequencies

We simulate RF propagation using a model of downtown Boston (provided by the software developer) in the 915 MHz and 433 MHz ISM bands. Our simulation analyzes a trail of client devices transmitting to a constellation of 16 gateways (which we believe to be representative of an LPWAN deployment in an urban area). Though representing idealized scenarios, these simulations provide a number of interesting observations that we will elaborate upon using the two example locations in Figure 5.5. The bottom two rows show the resulting channel response after (1) being passed through a 5 MHz band-limited system (added for illustration purposes) and (2) being modulated by a chirp, followed by demodulation by its matched filter.

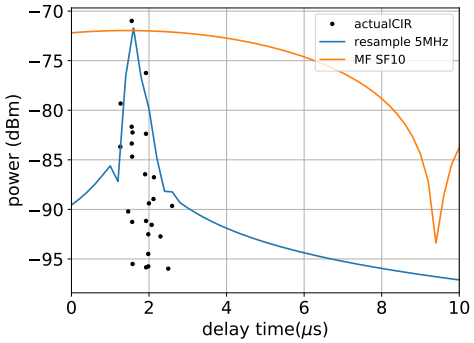
First, we see very few line-of-sight paths in a simulation of an urban environment, despite the excellent coverage by gateways. Second, the shortest paths often do not have high power. This is unfortunate for narrowband systems which only receive an averaged version of the signal and will peak at the location with highest average power. The matched filter response can track the averaged response faithfully at a granularity much finer than the sampling time of the signal. The only limit to upsampling is noise, and Figure 5.6 demonstrates how matched filters help reduce noise in the vicinity of the peak. Finally, though the channel response is dependent on our choice of frequency, the changes in response are extremely localized and unpredictable. We can thus say that any physics-only methods will be unreliable due to the overwhelming presence of strong multipath in urban scenarios, and we should use richer fingerprint-based techniques.

5.9.2 Matched Filtering Outputs across Location and Time

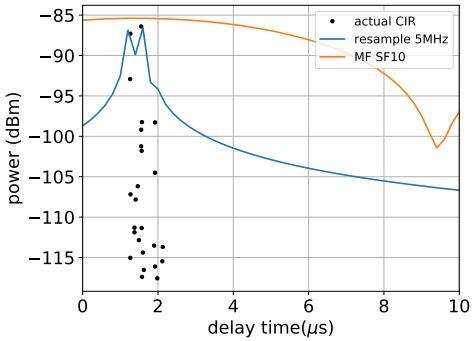
Figure 5.6 shows the complex matched filter response of a real signal capture from a transmitter that is ≈ 85 m and ≈ 105 m away from two gateways in a non-line-of-sight setting. The transmitter is using 125 kHz



Propagation paths

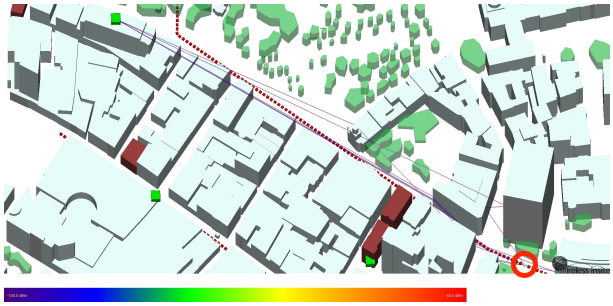


434 Mhz

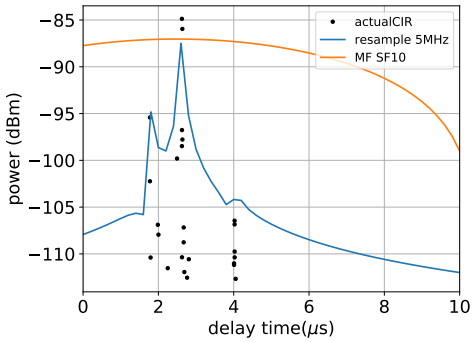


905 MHz

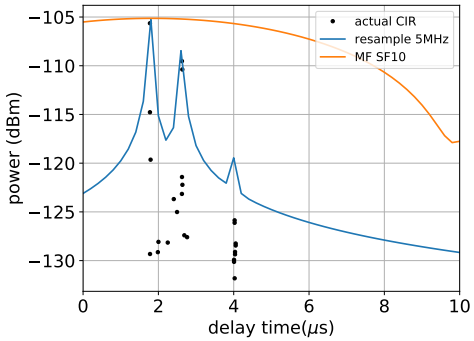
(a) Test position 113



Propagation paths



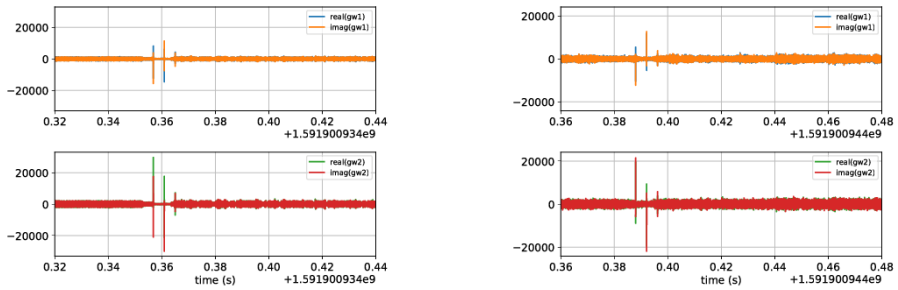
434 Mhz



905 Mhz

(b) Test position 144

Figure 5.5: Ray tracing simulations in an urban environment. Red circle shows location of transmitter and green cube is location of gateway.



(a) 905 MHz (b) 434 MHz

Figure 5.6: Matched filter outputs at different gateways

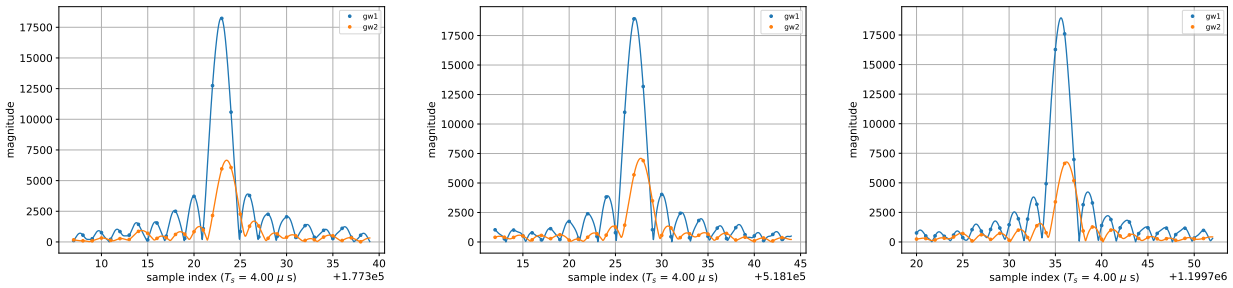


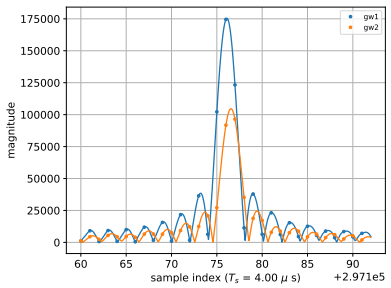
Figure 5.7: Comparison of upsampled output at different times

bandwidth chirps transmitted at 434 MHz and 905 MHz. The matched filter is tuned to target downchirps in the LoRa packet instead of the upchirps, so as to avoid confusion from data symbols (which are encoded using upchirps). We can see that the matched filter is able to sharply pick out the presence of downchirp symbols. We also observe the presence of a frequency offset that changes the phase of the response for each symbol. The phase change between symbols on both gateways is almost identical, which indicates that our GPS-based synchronization system for gateways is functioning accurately.

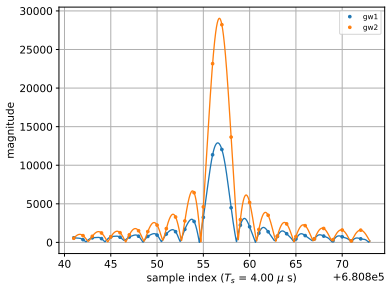
In Figure 5.7, we zoom into the time intervals with high response and upsample the response using FFT resampling. The plots show the magnitude of the upsampled response at another test point captured at three different instances of time (captures over tens of seconds). Despite the presence of a frequency offset between the device and gateways (which would trouble phase-based systems), the peak matched filter response is stable to under 50 ns over time and the shape closely matches the expected ambiguity function shown in Figure 5.4. A 50 ns uncertainty corresponds to $\approx$ 15 m range difference uncertainty.

5.9.3 Time-Based Fingerprints in Different Environments

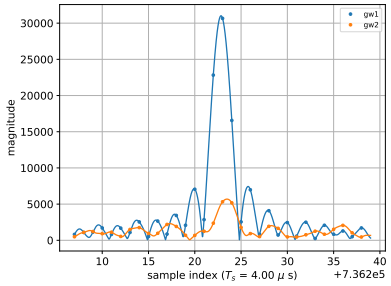
Figure 5.8 shows the upsampled matched filter responses from three locations in a variety of different environments. The first row shows the response from a location with line-of-sight to both gateways. The



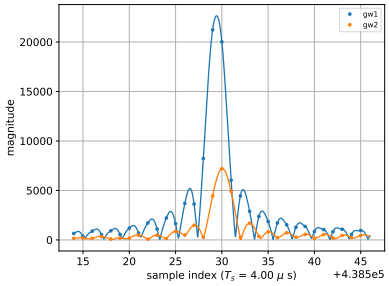
(a) Line-of-sight at 434 MHz



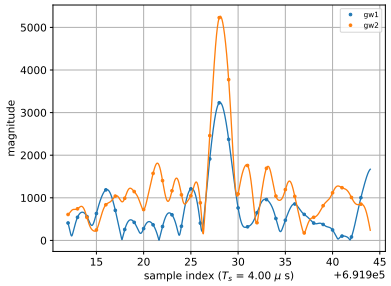
(b) Line-of-sight at 905 Mhz



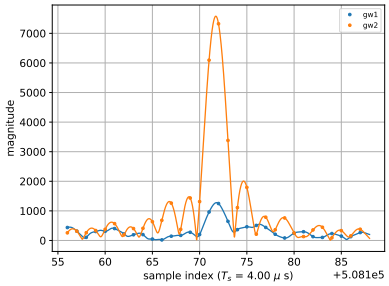
(c) Non line-of-sight at 434 MHz



(d) Non line-of-sight at 905 Mhz



(e) Heavily obstructed at 434 MHz



(f) Heavily obstructed at 905 Mhz

Figure 5.8: Upsampled matched filter results across frequencies and locations

responses at the 434 MHz and 905 MHz frequency bands vary in magnitude (and phase), but both closely resemble the expected ambiguity function. The second row shows the response from a location without line-of-sight to the gateways. The 905 MHz response looks normal, but the 434 MHz is distorted. The third location in the last row is heavily obstructed from both gateways and the responses for both frequencies are heavily distorted compared to the expected ambiguity function.

These results show that the relative positions and sizes of the peaks in the matched filter response, in addition to the location of the primary peak (typically used to timestamp a message), can be useful fingerprints of the locations. As a simple application, looking at the similarity between the response at different frequencies and the expected ambiguity function can indicate if we are working in a non-line-of-sight scenario.

5.10 Summary

In this chapter, we have looked at the problem of localization of low-power devices in challenging urban settings. Driven by radio propagation simulations and a study of various localization paradigms, we advocate for the use of fingerprinting methods using channel features. We also look at the time synchronization challenges that must be dealt with for TDoA-based methods to be used and propose the use of matched filtering. This method is agnostic to constant phase-offsets caused by LO synchronization in software radios. Finally, we propose the use of FFT upsampling to get more precise timing features from the signal and evaluate the results using a campus-scale testbed. Our evaluations show that the matched filter outputs are indeed stable and accurate up to $50nsec$ (corresponding to ≈ 15 m in range difference accuracy), which is $80\times$ smaller than the sampling interval of the signal. We also demonstrate the use of varied frequency bands to provide more information about the nature of radio propagation. A full analysis of the locality and fingerprint-based localization system is left as future work.

Chapter 6

Conclusions and future work

This dissertation explores the challenges of developing a distributed receiver system for LPWAN transmissions. We believe the work in this thesis significantly improves the performance and capabilities of LPWANs and will aid in accelerating their adoption. The following is a summary of our contributions:

- **Capabilities of LPWANs:** We deployed a campus-wide LoRaWAN network using four gateways that could communicate with low-power devices up to 5 km away. After evaluating the network coverage around campus, we determine that performance is promising but non-uniform, particularly inside buildings. We also developed and evaluated a low-power LPWAN client device. We then characterized its energy consumption profile to determine that slow data rates can significantly affect the battery life of LPWAN client devices.

- **Scalability Analysis of LoRaWAN:** We perform a theoretical analysis of the commonly deployed LoRaWAN Class-A (ALOHA) protocol and find that the ideal distribution of devices around a single gateway should follow a Pareto distribution if power control is in place. We also estimate that saturating a LoRaWAN gateway would take thousands of low update rate devices. However, this scalability rapidly falls apart if power control and interference are not properly managed. We also determine that a saturated LoRa network will significantly affect other ISM band communication.

- **Distributed Reception:** We develop a practical distributed reception system for LPWANs, named Charm, that improves data rates and coverage areas in the network and increases battery life of clients. By coherently combining across eight base stations, Charm improves the SNR of a typical LoRaWAN transmission by 3.16 dB, extending battery life by up to four times. We demonstrate an improvement in range to 200 m for an indoor deployment compared to the 60 m range of a LoRaWAN gateway under

the same circumstances. We use trace-driven simulation, based on city-wide drive tests, to estimate an overall increase in coverage area by up to 2× due to Charm over LoRaWAN.

- **Precise Time-Synchronization:** We develop Pulsar, a clock synchronization platform for wireless clock synchronization of indoor LPWAN gateways. Powered with a UWB timestamping radio and a precise CSAC, we show that Pulsar provides better than 5 ns synchronization per hop across a network and provides standard 1PPS and 10 MHz outputs to synchronize other devices. We also develop a protocol to extend time-synchronization to multiple hops.

- **Channel Fingerprinting:** We determine that localization in an unpredictable multipath-prone urban scenario would have to use fingerprinting methods, and simple TDoA methods would be error-prone. We develop a matched filter and FFT upsampling method to generate channel fingerprints that can generate timestamps accurate up to 50 ns, which is 80× smaller than the sampling interval of the signal. We also demonstrate that a distorted shape of the matched filter response indicates the presence of multipath.

6.0.1 Future Directions

During the development of distributed receivers in this thesis, we identified a number of interesting challenges and open questions in the areas of LPWANs and distributed reception that should be explored in the future.

The Effects of Power Control and Interfering Networks: We determined that LoRaWAN networks could scale well if all the networks in a given area were to collaborate and implement proper power control. Further exploration should be done about the consequences of breaking either of those assumptions. If devices do not properly control their output power (they could be communicating with a different gateway or network), we run into the risk of overwhelming the radio frontend of a nearby gateway, which could prevent all transmissions. Though this scenario might sound extreme, it is extremely likely in the real-world with competing networks. We should perform a study of scalability under these scenarios.

Use of Advanced MAC Protocols: Most LPWANs today use simple MAC protocols like ALOHA or CSMA. Though these are energy-efficient and simple to implement, they do not have nearly the efficiency or throughput as more advanced protocols like TDMA. However, adapting a complex MAC protocol for very low-power consumption can be challenging, particularly when we cannot assume, for example, continuous access to reliable clocks. Thus, we should explore how advanced MAC protocols can be introduced into LPWANs and if they affect any of the major advantages of existing LPWAN systems.

Use of Whitespaces: We have discussed the opening of TV UHF whitespaces to unlicensed communication and the potential advantages it could bring to localization systems and LPWANs. This is another

direction that should be explored further. However, the use of whitespace frequencies is more complicated compared to other ISM band spectrum, as there is an additional requirement to check with an online database for available frequency bands in a region, which can change over time. The important questions to ask would be: (1) how can we perform selective handoff to whitespace frequencies so as to decongest the ISM band spectrum in a region? and (2) how does the availability of additional frequency bands affect existing LPWAN localization using either RSS or TDoA?

Charm with Synchronized Gateways: During the development of Charm in Chapter 3, we focused on removing the need to have access to precise timing and clocks. However, with the development of Pulsar and the proliferation of other precise time synchronization technologies like White Rabbit, IEEE 802.11 FTM, etc., we can expect precise timing systems to be more accessible and commonplace. Future work could be carried out to understand how the algorithms proposed in Charm could be simplified with the assumption of time synchronization and if it could further improve coherent combining performance.

LO Synchronization: In Section 5.6, we introduced the problem of local oscillator synchronization, which prevents radio frontends from phase synchronizing their RF mixing hardware, which in turn introduces an arbitrary phase offset in the received baseband signal. Developing a technique for LO synchronization over large distances has the potential to improve all existing communication systems that use multiple gateways or basestations, including LPWANs and cellular.

Localization Using Channel Response Fingerprints: In Chapter 5, we showed a proof-of-concept use of TDoA features as location fingerprints. This must be followed up by a thorough field test spanning a large area and using a larger number of gateways. These tests could then aid in the development of appropriate fingerprint matching algorithms. With the popularity of machine learning-based techniques, we believe our matched filtering method can provide sufficiently rich features to be able to train an accurate model.

Localization Using Direction-of-Arrival: In Section 5.4.4, we briefly introduced the concept of direction-of-arrival (DoA). This is an extremely promising technique that could enable very precise direction-finding of LPWAN devices using only a handful of gateways. The affordability and accessibility of radio ICs with multiple radio frontends is rapidly increasing, thus making it practical to develop low-cost DoA gateways. Inspired by direction-finding hardware for FM radios, it would also be interesting to develop a portable direction-finding device that can work with LPWAN transmitters. We should explore the capabilities and limitations of DoA systems in the sub-GHz band frequencies.

Bibliography

[1] LoRaBug repository https://github.com/OpenChirp/LoRaBug. 16

[2] The things network https://www.thethingsnetwork.org/, accessed 10 Jan 2017. 9

[3] Michiel Aernouts, Rafael Berkvens, Koen Van Vlaenderen, and Maarten Weyn. Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas. *Data*, 3(2):13, 2018. 78

[4] David W Allan. Time and frequency(time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 1987. 63

[5] David W Allan and HE Machlan. Time transfer using nearly simultaneous reception times of a common transmission. *26th Annual Symposium on Frequency Control*, pages 309–316, 1972. 58

[6] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999. 45

[7] Vladimir Atanasovski and Alberto Leon-Garcia. *Future access enablers for ubiquitous and intelligent infrastructures*. Springer, 2015. 8

[8] Martin Azizyan and Romit Roy Choudhury. SurroundSense: mobile phone localization using ambient sound and light. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(1):69–72, 2009. 78

[9] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, volume 2, pages 775–784. Ieee, 2000. 78

[10] Victor Bahl and Venkat Padmanabhan. Enhancements to the RADAR user location and tracking system. 2000. 78

[11] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. AirSync: Enabling distributed multiuser mimo with full spatial multiplexing. *IEEE/ACM Transactions on Networking*, 21(6):1681–1695, 2013. 58

[12] Brian Barrett. The biggest iPhone news is a tiny new chip inside it. *Wired*, Sep 2019. 73

[13] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan 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*, pages 59–67, 2016. 9, 26, 50

[14] Orne Brocaar. Lora server - open-source lorawan server https://docs.loraserver.io/loraserver/, accessed Jan 10, 2017. 20

[15] Alison K Brown and Peter Brown. Ultra low-power GPS recorder (TrackTag). In *Proceedings of ION 61st Annual Meeting*, volume 6. Citeseer, 2005. 77

[16] Aleksandra Checko, Henrik L Christiansen, Ying Yan, Lara Scolari, Georgios Kardaras, Michael S Berger, and Lars Dittmann. Cloud RAN for mobile networks—A technology overview. *IEEE Communications surveys & tutorials*, 17(1):405–426, 2015. 30, 33

[17] I Chih-Lin, Jinri Huang, Ran Duan, Chunfeng Cui, Jesse Jiang, and Lei Li. Recent progress on C-RAN centralization and cloudification. *IEEE Access*, 2:1030–1039, 2014. 33

[18] Semtech Coporation. *AN1200.13 LoRa modem designer's guide*, 2013. Revision 1. 24

[19] Semtech Corporation. LoRa Geolocation Solution for LPWAN https://www.semtech.com/company/press/semtechs-lora-geolocation-solution-for-low-power-wide-area-networks-is-now-available. 06 2016. 3, 77

[20] Semtech Corporation. *SX1301 LoRa Baseband Signal Processor*, 2017. 15

[21] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, 2009. 79

[22] DecaWave. *APS011 Sources of Error in DW1000 based Two-Way Ranging (TWR) Schemes*, 2014. v1.0. 67

[23] Decawave. http://www.decawave.com/. accessed 29 Jun 2020. 55

[24] Aitor Del Coso, Umberto Spagnolini, and Christian Ibars. Cooperative distributed MIMO channels in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 25(2):402–414, 2007. 32

[25] Jean-Daniel Deschênes, Laura C. Sinclair, Fabrizio R. Giorgetta, William Swann, Esther Baumann, Ian Coddington, and Nathan Newbury. Optical two-way time synchronization at the femtosecond level over a 4-km free space link. In *Imaging and Applied Optics 2015*, page LTh1C.3. Optical Society of America, 2015. 58

[26] Mischa Dohler, Athanasios Gkelias, and Hamid Aghvami. A resource allocation strategy for distributed MIMO multi-hop communication systems. *IEEE Communications Letters*, 8(2):99–101, 2004. 32

[27] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 309–321, 2017. 32

[28] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 36(SI):147–163, 2002. 57

[29] Bernat Carbonés Fargas and Martin Nordal Petersen. GPS-free geolocation using LoRa in low-power WANs. In *2017 global internet of things summit (Giots)*, pages 1–6. IEEE, 2017. 77

[30] FCC. Second report and order and memorandum opinion and order. 2010. 4, 76

[31] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with glossy. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 73–84. IEEE, 2011. 57

[32] Brian Ferris, Dieter Fox, and Neil D Lawrence. WiFi-SLAM using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007. 78

[33] Fabio Forno, Giovanni Malnati, and Giuseppe Portelli. Design and implementation of a bluetooth ad-hoc network for indoor positioning. *IEE proceedings-Software*, 152(5):223–228, 2005. 78

[34] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149, 2003. 57

[35] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–12, 2019. 9

[36] Bernhard Großwindhager, Michael Rath, Josef Kulmer, Mustafa S Bakr, Carlo Alberto Boano, Klaus Witrisal, and Kay Römer. SALMA: UWB-based single-anchor localization system using multipath

assistance. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 132–144, 2018. 78

[37] GSM Association. 3GPP Low Power Wide Area Technologies (White Paper). 2016. 8

[38] Ismail Guvenc, Sinan Gezici, and Zafer Sahinoglu. Ultra-wideband range estimation: Theoretical limits and practical algorithms. In *2008 IEEE International Conference on Ultra-Wideband*, volume 3, pages 93–96. IEEE, 2008. 79

[39] Mesud Hadzialic, Branko Dosenovic, Merim Dzaferagic, and Jasmin Musovic. Cloud-RAN: Innovative radio access network architecture. In *Proceedings ELMAR-2013*, pages 115–120. IEEE, 2013. 33

[40] Ezzeldin Hamed, Hariharan Rahul, Mohammed A Abdelghany, and Dina Katabi. Real-time distributed MIMO systems. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 412–425, 2016. 32

[41] Ville Honkavirta, Tommi Perala, Simo Ali-Loytty, and Robert Piché. A comparative survey of WLAN location fingerprinting methods. In *2009 6th workshop on positioning, navigation and communication*, pages 243–251. IEEE, 2009. 78

[42] IBM. LoRaWAN in C `https://www.research.ibm.com/labs/zurich/ics/lrsc/lmic.html`, viewed 30 Jun 2020. 16

[43] IEEE. IEEE 1588 precision time protocol (PTP) version 2. 2008. 53

[44] Thomas Janssen, Rafael Berkvens, and Maarten Weyn. Comparing Machine Learning Algorithms for RSS-Based Localization in LPWAN. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 726–735. Springer, 2019. 78

[45] Hermann Kopetz and Wilhelm Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, 100(8):933–940, 1987. 57

[46] Swarun Kumar, Diego Cifuentes, Shyamnath Gollakota, and Dina Katabi. Bringing cross-layer MIMO to today's wireless LANs. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 387–398, 2013. 32

[47] Link Labs. Symphony link `https://www.link-labs.com/symphony/`, viewed 30 Jun 2020. 9

[48] Erik G Larsson, Ove Edfors, Fredrik Tufvesson, and Thomas L Marzetta. Massive MIMO for next generation wireless systems. *IEEE communications magazine*, 52(2):186–195, 2014. 32

[49] Roger A Light. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software*, 2(13):265, 2017. 19

[50] Roman Lim, Balz Maag, and Lothar Thiele. Time-of-flight aware time synchronization for wireless embedded systems. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 149–158, USA, 2016. Junction Publishing. 58

[51] Kate Ching-Ju Lin, Shyamnath Gollakota, and Dina Katabi. Random access heterogeneous MIMO networks. *ACM SIGCOMM Computer Communication Review*, 41(4):146–157, 2011. 32

[52] Cheng Liu, Karthikeyan Sundaresan, Meilong Jiang, Sampath Rangarajan, and Gee-Kung Chang. The case for re-configurable backhaul in Cloud-RAN based small cell networks. In *2013 Proceedings IEEE INFOCOM*, pages 1124–1132. IEEE, 2013. 33

[53] LoRa Alliance. LoRaWAN – What is it? A Technical Overview of LoRa and LoRaWAN. Technical report, 2015. 30

[54] LoRa Alliance. LoRaWAN Geolocation Whitepaper. Technical report, January 2018. 3, 79

[55] M. Centenaro et al. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *arXiv preprint arXiv:1510.00620*, 2015. 8

[56] M. Maroti and B. Kusy and G. Simon and A. Ledeczi. The flooding time synchronization protocol. *Proc. ACM Sensys*, 2004. 57

[57] David L Mills. Computer network time synchronization. In *Report Dagstuhl Seminar on Time Services Schloß Dagstuhl, March 11.–March 15. 1996*, volume 12, page 332. Springer, 1997. 63, 64

[58] David L Mills. NTP architecture, protocol and algorithms. Technical report, technical report, Electrical Engineering Department University of Delaware, 2002. 55, 57, 70

[59] Joe Moell. Wide-range antenna arrays for the roanoke doppler. *73 Magazine*, Jun 1995. 80

[60] Joseph D Moell and Thomas N Curlee. *Transmitter Hunting: Radio Direction Finding Simplified*, volume 2701. McGraw Hill Professional, 1987. 80

[61] Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota. 3D localization for sub-centimeter sized devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 108–119, 2018. 77

[62] Masayuki Ochiai, Masahiro Fujii, Atsushi Ito, Yu Watanabe, and Hiroyuki Hatano. A study on indoor position estimation based on fingerprinting using GPS signals. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 727–728. IEEE, 2014. 78

[63] Charalampos Orfanidis, Laura Marie Feeney, Martin Jacobsson, and Per Gunningberg. Investigating interference between LoRa and IEEE 802.15.4g networks. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2017. 9, 26

[64] Bradford W Parkinson and Stephen W Gilbert. NAVSTAR: Global positioning system—Ten years later. *Proceedings of the IEEE*, 71(10):1177–1186, 1983. 57

[65] Stella Parks and J. Kenji López-Alt. *BraveTart: Iconic American Desserts*. 2017. iv

[66] Tara Petrić, Mathieu Goessens, Loutfi Nuaymi, Laurent Toutain, and Alexander 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)*, pages 1–7. IEEE, 2016. 8

[67] K Pister and Lance Doherty. TSMP: Time synchronized mesh protocol. *IASTED Distributed Sensor Networks*, pages 391–398, 2008. 68

[68] David Plets, Nico Podevijn, Jens Trogh, Luc Martens, and Wout Joseph. Experimental performance evaluation of outdoor TDoA and RSS positioning in a public lora network. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2018. 78

[69] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. Sourcesync: A distributed wireless architecture for exploiting sender diversity. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 171–182, New York, NY, USA, 2010. ACM. 58

[70] Hariharan Shankar Rahul, Swarun Kumar, and Dina Katabi. JMB: scaling wireless capacity with user demands. In *Proceedings of the ACM SIGCOMM 2012 Conference*, pages 235–246. ACM, 2012. 32, 58

[71] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304, 2012. 78

[72] Gowri Sankar Ramachandran, Fan Yang, Piers Lawrence, Sam Michiels, Wouter Joosen, and Danny Hughes. μPnP-WAN: Experiences with LoRa and its deployment in DR Congo. In *2017 9th Interna-*

tional Conference on Communication Systems and Networks (COMSNETS), pages 63–70. IEEE, 2017. 8

[73] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. NB-IoT system for M2M communication. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–5. IEEE, 2016. 8

[74] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: A survey. *IEEE Commun. Surv. Tutorials*, 19(2), 2017. 8

[75] Remcom. Wireless InSite EM Propagation software v3.3.3 https://www.remcom.com/wireless-insite-em-propagation-software, accessed 28 Jun 2020. 89

[76] Mark A Richards. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005. 85, 86

[77] Dario Sabella, Peter Rost, Yingli Sheng, Emmanouil Pateromichelakis, Umer Salim, Patricia Guitton-Ouhamou, Marco Di Girolamo, and Giovanni Giuliani. RAN as a service: Challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network. In *2013 Future Network & Mobile Summit*, pages 1–8. IEEE, 2013. 33

[78] Samsara. https://www.samsara.com/, viewed 30 Jun 2020. 2

[79] Ramon Sanchez-Iborra and Maria-Dolores Cano. State of the art in LP-WAN solutions for industrial IoT services. *Sensors*, 16(5):708, 2016. 8

[80] Mamoru Sawahashi, Yoshihisa Kishiyama, Akihito Morimoto, Daisuke Nishikawa, and Motohiro Tanno. Coordinated multipoint transmission/reception techniques for LTE-advanced [Coordinated and Distributed MIMO]. *IEEE Wireless Communications*, 17(3):26–34, 2010. 32

[81] Thomas Schmid, Zainul Charbiwala, Zafeiria Anagnostopoulou, Mani B. Srivastava, and Prabal Dutta. A case against routing-integrated time synchronization. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 267–280, New York, NY, USA, 2010. ACM. 56, 67

[82] Thomas Schmid, Zainul Charbiwala, Roy Shea, and Mani B Srivastava. Temperature compensated time synchronization. *IEEE Embedded Systems Letters*, 1(2):37–41, 2009. 58

[83] Semtech Corporation. *SX1261/2 Long Range, Low Power, sub-GHz RF transceiver*, 2019. 4, 10

[84] Sensus. https://sensus.com/, viewed 30 Jun 2020. 2

[85] Wei-Liang Shen, Kate Ching-Ju Lin, Shyamnath Gollakota, and Ming-Syan Chen. Rate adaptation for 802.11 multiuser MIMO networks. *IEEE Transactions on Mobile Computing*, 13(1):35–47, 2013. 32

[86] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. Argos: Practical many-antenna base stations. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 53–64, 2012. 32

[87] Oliver Shih and Anthony Rowe. Can a phone hear the shape of a room? In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 277–288, 2019. 78

[88] Sigfox. https://www.sigfox.com/, accessed 10 Jan 2017. 8

[89] James J Spilker Jr, Penina Axelrad, Bradford W Parkinson, and Per Enge. *Global Positioning System: Theory and Applications, Volume I*. American Institute of Aeronautics and Astronautics, 1996. 80

[90] Claude Mbusa Takenga, Quan Wen, and Kyandoghere Kyamakya. On the accuracy improvement issues in GSM location fingerprinting. In *IEEE Vehicular Technology Conference*, pages 1–5. IEEE, 2006. 78

[91] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M Voelker. SAM: enabling practical spatial multiple access in wireless LAN. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 49–60, 2009. 30, 32

[92] Viktor Toldov, JP Meijers, Román Igual-Pérez, Riaan Wolhuter, Nathalie Mitton, and Laurent Clavier. Performance evaluation of LoRa radio solution for PREDNET wildlife animal tracking project. In *LPWAN 2016*, 2016. 8

[93] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. Mitigating inter-network interference in LoRa networks. *arXiv preprint arXiv:1611.00688*, 2016. 9

[94] Adi Weller Weiser, Yotam Orchan, Ran Nathan, Motti Charter, Anthony J Weiss, and Sivan Toledo. Characterizing the accuracy of a self-synchronized reverse-GPS wildlife localization system. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE, 2016. 58, 77

[95] Dirk Wubben, Peter Rost, Jens Steven Bartelt, Massinissa Lalam, Valentin Savin, Matteo Gorgoglione, Armin Dekorsy, and Gerhard Fettweis. Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN. *IEEE signal processing magazine*, 31(6):35–44, 2014. 30, 33

[96] Xiufeng Xie and Xinyu Zhang. Scalable user selection for MU-MIMO networks. In *Proceedings of the 2014 IEEE INFOCOM*, pages 808–816, 2014. 30, 32

[97] Vivek Yenamandra and Kannan Srinivasan. Vidyut: exploiting power line infrastructure for enterprise wireless networks. *ACM SIGCOMM Computer Communication Review*, 44(4):595–606, 2014. 32, 58