Charm: Exploiting Geographical Diversity Through Coherent Combining in Low-Power Wide-Area Networks

Adwait Dongare
Carnegie Mellon University
adongare@cmu.edu

Revathy Narayanan
Carnegie Mellon University
IIT Madras
revathyn@andrew.cmu.edu

Akshay Gadre
Carnegie Mellon University
agadre@andrew.cmu.edu

Anh Luong
Carnegie Mellon University
anhluong@cmu.edu

Artur Balanuta
Carnegie Mellon University
Instituto Superior Técnico, Lisbon
artur@cmu.edu

Swarun Kumar
Carnegie Mellon University
swarun@cmu.edu

Bob Iannucci
Carnegie Mellon University
bob.iannucci@west.cmu.edu

Anthony Rowe
Carnegie Mellon University
agr@ece.cmu.edu

ABSTRACT

Low-Power Wide Area Networks (LPWANs) are an emerging wireless platform which can support battery-powered devices lasting 10-years while communicating at low data-rates to gateways several kilometers away. Not all such devices will experience the promised 10 year battery life despite the high density of LPWAN gateways expected in cities. Transmission from devices located deep within buildings or in remote neighborhoods will suffer severe attenuation forcing the use of slow data-rates to reach even the closest gateway, thus resulting in battery drain.

This paper presents Charm, a system that enhances both the battery life of client devices and the coverage of LPWANs in large urban deployments. Charm allows multiple LoRaWAN gateways to pool their received signals in the cloud, coherently combining them to detect weak signals that are not decodable at any individual gateway. Through a novel hardware and software design at the gateway, Charm carefully detects which chunks of the received signal need to be sent to the cloud, thereby saving uplink bandwidth. We present a scalable solution to decoding weak transmissions at city-scale by identifying the set of gateways whose signals need to be coherently combined over time. In evaluations over a test network and from simulations using traces from a large LoRaWAN deployment in Pittsburgh, Pennsylvania, Charm demonstrates a gain of up to 3× in range and 4× in client battery-life.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Hardware** → **Sensor applications and deployments**; • **Networks** → *Wireless access points, base stations and infrastructure*;

KEYWORDS

LPWAN, sensor networks, coherent combining, diversity

ACM Reference format:
Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: Exploiting Geographical Diversity Through Coherent Combining in Low-Power Wide-Area Networks . In *Proceedings of IEEE/ACM Information Processing in Sensor Networks, Porto, Portugal, April 2018 (IPSN 2018),* 12 pages.
DOI: None

1 INTRODUCTION

Low Power Wide Area Networks (LPWANs) are increasingly seen as an attractive communication platform for city-scale Internet-of-Things (IoT) deployments. They offer the ability to wirelessly connect energy-constrained devices to gateways over distances of many kilometers. LPWANs also have power and cost advantages over alternatives like cellular networks, particularly in deploy-once, low maintenance and low throughput sensing applications.

While LPWANs are far from pervasive, the capabilities of networks like LoRaWAN [20, 25], SigFox [11] and Ingenu's RPMA [16] have attracted investment and have spawned early deployments. These technologies operate on the unlicensed ISM spectrum, allowing businesses and consumers alike to deploy their own devices and gateways. With Comcast recently announcing integration of LPWAN radios into future set-top boxes in the U.S. [30], LPWANs are likely to grow rapidly. Given that each LPWAN gateway promises a range of up to ten kilometers [20], major cities are likely to see a fast-paced expansion in LPWAN coverage.

Despite the expected rise in density of LPWAN gateways, not all devices will experience the promised 10 year battery life. 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

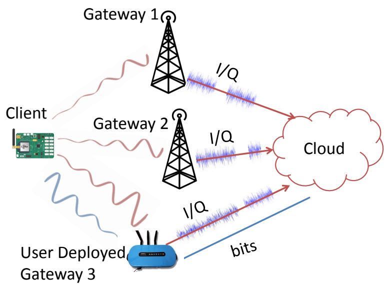


Figure 1: Charm: LPWAN joint decoding in the cloud

these devices may remain battery deprived or simply out of network reach in perpetuity, even as thousands of gateways proliferate city-wide.

This paper presents Charm, a system that enhances the coverage of LPWANs and the battery life of client devices in large urban deployments. 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 battery-life for nodes already in range – allowing client devices to spend less energy per transmitted bit. Charm is built on the LoRaWAN platform [20], a popular and widely available LPWAN technology. 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 Wi-Fi [19, 34] and cellular [1, 6] 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. We provide an overview of our approach to address each of these challenges.

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 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 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.

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.

We evaluate Charm in both indoor and outdoor environments using 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 which services a large 10 sq.km. area that we use to acquire traces for large-scale simulations. 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 2x due to Charm over LoRaWAN.

Contributions: We make the following novel contributions:

- 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.

2 RELATED WORK

Low-Power Wide-Area Networks: Recent years have seen much interest in LPWANs, including the development of new hardware and standards. Private enterprises such as Semtech [20] and SigFox [27] have developed LPWAN chipsets that use extremely narrow bands of unlicensed spectrum. In contrast, cellular standardization bodies have developed two standards for LPWAN communication for cellular base stations to communicate with low-power IoT devices over licensed spectrum: LTE-M [14] and NB-IOT [28]. Unlike LoRaWAN and SigFox, these technologies require devices to periodically wake up to synchronize with the network – a burden on battery life.

Several recent measurement studies have been conducted to evaluate the performance and range of LPWAN networks [31, 32] and perform theoretical capacity analysis [18]. Early pilot deployment efforts are also underway with SigFox deploying their hardware to connect security alarms to the cloud in Spain [27], smart blood refrigerators in the Democratic Republic of the Congo [13] and smart city applications [22]. These efforts motivate the challenge of limited range, performance and battery-drain of LPWAN clients. A recent system, Choir [26], has demonstrated improving range and scalability of LPWANs through collaborations of weak client radios. In contrast, this paper seeks to use collaboration between gateways without any modifications to client behavior whatsoever to improve the battery life of even a single client.

Distributed MIMO and Coherent Combining: A large body of work has proposed the use of multiple-antennas (MIMO) to improve SNR and reduce interference [17, 29, 34]. In the WiFi context, past systems have used multi-user MIMO to improve performance on the uplink [19, 33, 34]. In the cellular context, massive MIMO proposals have demonstrated scaling gains of towers with a large number of antennas [4, 7]. There has been much theoretical work on distributed MIMO overall in both the sensor networking context [2] and wireless LANs [12] and cellular networks [24]. More recently, practical distributed MIMO systems, primarily in the LAN-context have demonstrated both multiplexing and diversity gains [8, 9, 35]. 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.

Cloud Radio Access Networks (Cloud-RAN): Multiple research efforts from the industry and academia have advocated the use of PHY layer processing at the cloud as opposed to the base stations [5, 23]. In the cellular context, CloudRAN aims to perform baseband processing at the cloud, allowing base stations to be simple and easy to deploy [1, 6]. 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 [3, 15]. 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 BACKGROUND

In this section, we describe the two key topics that enable Charm: coherent combining, and the PHY and MAC layers of LoRaWAN.

3.1 Coherent Combining in Distributed MIMO

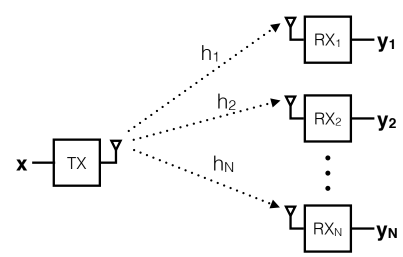


Figure 2: Coherent combining helps receivers collaboratively improve signal-to-noise ratio

Wireless radios leverage multiple antennas (MIMO or multiple-input multiple-output) to improve throughput. This paper 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 decodability.

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 is 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 paper describes the challenges in implementing coherent combining in the low-power wide-area context where neither assumption holds.

3.2 Primer on LoRaWAN PHY and MAC

LoRaWAN is a popular LPWAN technology that operates in the sub-GHz ISM band (900 MHz in the U.S.) and bandwidths of 125-500kHz. LoRaWAN clients can transmit at low-data rates (few kbps) to gateways up to 10 km away in free space and last up to 10 years on AA batteries. Below, we detail a few key design decisions of LoRaWAN.

LoRa, The PHY: LoRa's physical layer is based on chirp-spread spectrum modulation, i.e. using a chirp signal that continuously varies in frequency. This makes it resilient to interference, multi-path fading and Doppler effects. Every LoRaWAN packet begins with a preamble of sixteen repeated chirps followed by data. 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.[1] Increased spreading factors are used to simultaneously slow down transmissions and improve 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 paper therefore strives to reduce spreading factor (improve data rate) for weak transmitters.

The MAC: 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. Gateways are simple and relatively inexpensive forwarders that send received packets to a cloud LoRaWAN server, and can be commanded by the server to transmit data to clients at a specific time. 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. 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). In this paper, we refer to these as user-deployed gateways. The penultimate goal of this paper is to make individual unreliable user-deployed gateways more valuable by pooling together PHY-layer processing at the cloud.

[1] More precisely, increasing spreading factor from n to $n+1$ scales data rate by $(n+1)/2n$.

4 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 Sec. 8.1 demonstrate that lowering transmit time results in a direct and significant impact on battery life.

Fig. 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 accommodate 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 paper 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

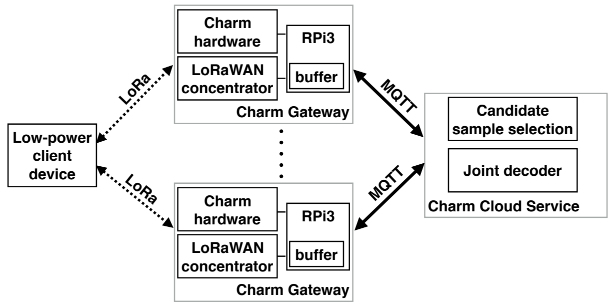


Figure 3: Architecture of Charm

processing system at the cloud that can operate at city-scale. Next we elaborate on each of these components.

5 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, that 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.

5.1 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 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.

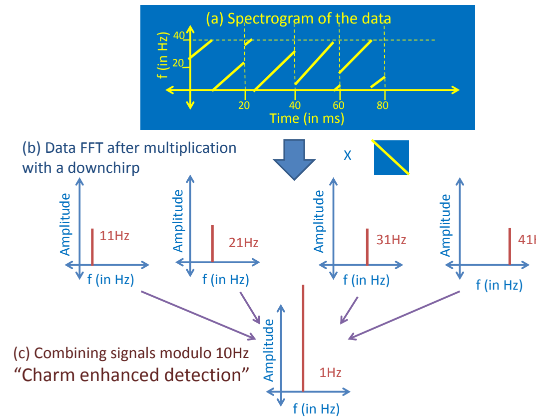


Figure 4: The enhanced Charm packet detection process: The chirp signal (a) is multiplied by a downchirp in the Fourier domain (b). Windows of the resulting signal are then combined together (c) for threshold detection.

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 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 are 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 δf = 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:

$$y_i(t) = he^{j2\pi(f(t)-x_i\delta f)t} + n_1 \qquad (1)$$

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).

$$y_i(t) = he^{j2\pi(f(t) - x_i \delta f)\frac{k}{\delta f}} + n_1 = he^{j2\pi f(t)t} + n_1 \qquad (2)$$

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 $\frac{max(abs(fft(C)))}{mean(abs(fft(C)))} > \tau$ **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.

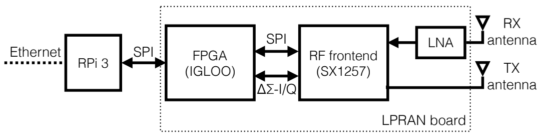
Two properties of CFO make its impact on Charm's algorithm above particularly damaging: (1) CFO unlike data introduces a frequency shift that is not discrete, but continuous. As a result, it is not simply eliminated by looking at the chirp in the Fourier domain "modulo δf" akin to the data as described 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 eliminate 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.

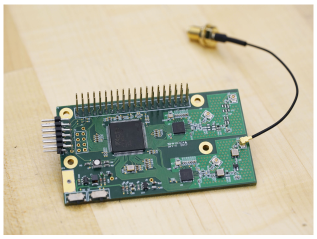
Charm overcomes this problem by realizing that while estimating Δf_{CFO} from the data symbols alone is challenging, it is sufficient to estimate Δf_{CFO} modulo δf to detect the LoRa packet. To see why, recall that the frequency offset over a packet Δf_{CFO} can be decoupled into two components: $[\frac{\Delta f_{CFO}}{\delta f}]\delta f + \{\frac{\Delta f_{CFO}}{\delta f}\}\delta f$, 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 all 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.

5.2 Programmable Hardware Design

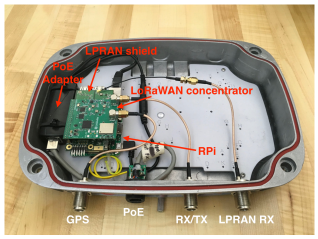
Charm must process raw I/Q samples from the gateway and selectively relay this information to the cloud in real-time. However, existing LoRaWAN gateway hardware cannot provide the raw I/Q streams necessary for joint decoding. In contrast, deploying a full software-defined radio (SDR) at the gateway allows packet decoding, it comes with high cost in term of power, sensitivity and unit price. We therefore develop a custom Charm hardware platform shown in Figure 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



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



(b) Charm programmable gateway



(c) Gateway with Charm hardware and LoRaWAN concentrator

Figure 5: Charm Hardware Platform

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 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:

$$x_i = \sum_{j=64*i}^{64*i+63} s_j \tag{3}$$

, 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.

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 are 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.

6 CHARM IN THE CLOUD

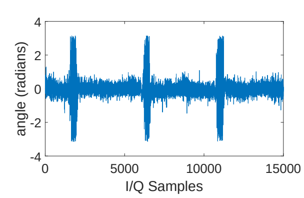
At the cloud, Charm seeks to coherently combine received signals from multiple gateways to recover weak received signals. At a high level, Charm collates I/Q samples from multiple gateways and estimates their packet start time and wireless channel. It then uses standard coherent SIMO combining (see Sec. 3.1) 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 nee d 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.

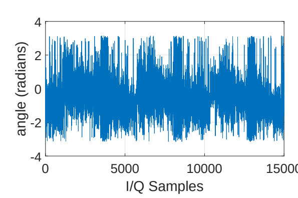
6.1 Time Synchronization at the Cloud

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 synchronize 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 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. Fig. 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.

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 Fig. 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's 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 a 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 coherent 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 these reliably.

6.2 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.

Which Signals Should We Combine?: 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 [21] 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.

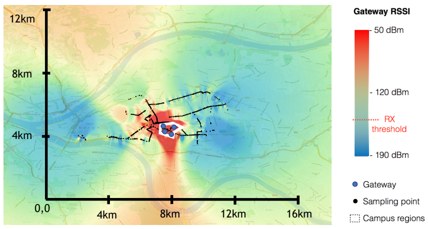
Joint Decoding Algorithm: Algorithm 2 below describes Charm's joint-decoding algorithm end-to-end. At a high level, our approach retrieves the wireless channels of the signals to be combined at any instance, their timing offsets and frequency offsets computed as described in the above sections. We, then 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.

Algorithm 2: Joint decoding algorithm

packets = receive_data(candidates);
for *p in packets* **do**
  p = $e^{j2\pi(\Delta_f)t}$ p ; `// Freq Offset Correction`
  p = $e^{j2\pi f(\Delta_t)}$ p ; `// Timing Offset Correction`
  h(p)= $\frac{p}{reference}$; `// Channel Estimation`
end
combined_packet=zeros(p);
for *p in packets* **do**
  combined_packet = combined_packet + h*p ;
end
decode(combined_packet);
SEND ACK;

6.3 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,



(a) OpenChirp network coverage heatmap

(b) Rooftop gateway

Figure 7: Openchirp LoRaWAN deployment in Pittsburgh

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 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 challenge 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 are of 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.

7 INTEGRATION WITH LORAWAN

Charm is implemented as a service running on a campus-wide LoRaWAN network installed at Carnegie Mellon University. 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 LoRaWAN network is powered by the open-source OpenChirp (http://www.openchirp.io) framework that allows students and faculty to login with their campus accounts and create device endpoints for capturing and sharing data. OpenChirp provides services that can be attached to data streams that can perform operations ranging from basic data storage to binary-to-JSON packing and unpack. A RESTful interface is used to configure meta-information about devices and set access control privileges that define how other users can interact with data streams. 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 (as compared to the normal data rate negotiation process) by spoofing improved SNR values during the join process. In this way, Charm can seamlessly operate with existing LoRaWAN devices with no modification.

Figure 7 shows examples of our gateway hardware deployed in the field along with the coverage in and around campus. The network is currently supporting a wide-range of applications from student projects, study-space monitoring, and building occupancy sensing all the way to mechanical room environmental sensing and utility sub-metering for the campus facilities maintenance team. The client transmitters in our experiments use the Semtech SX1276 LoRaWAN chipset. The figure also shows an example coverage heat map generated by nodes deployed throughout campus and the neighboring area. We see that the network with just four outdoor gateways is able to cover almost $10km^2$ of urban space.

8 RESULTS

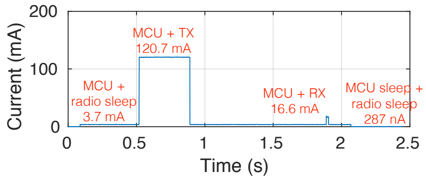
We evaluate Charm both through proof-of-concept experiments and large-scale trace-driven simulations. We perform our experiments in various indoors and outdoors environments across the campus.

8.1 Role of Transmission Rates on Battery Life

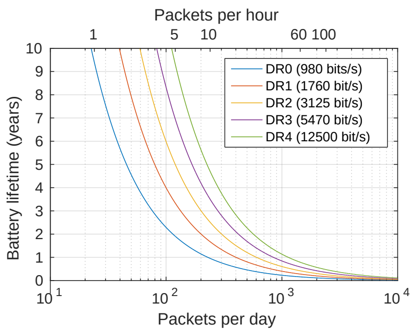
We study the energy profile of a typical battery-operated LoRaWAN client, as in Figure 8a. The device performs some local computation, sends a LoRa message, waits for an acknowledgment and then goes to low-power sleep mode. The radio transmission consumes the highest amount of energy (= area under the curve × voltage) by a large margin. Thus, any optimization to battery life must focus on reducing the energy of transmissions.

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. 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 8b shows the estimated battery life of a client device 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



(a) Typical client device current consumption for a complete LoRaWAN transmission. The device is powered at $3V$.



(b) Estimated lifetime of a client device powered by two AA batteries sending 36 byte packets at various data rates based on the energy profile.

Figure 8: Power statistics

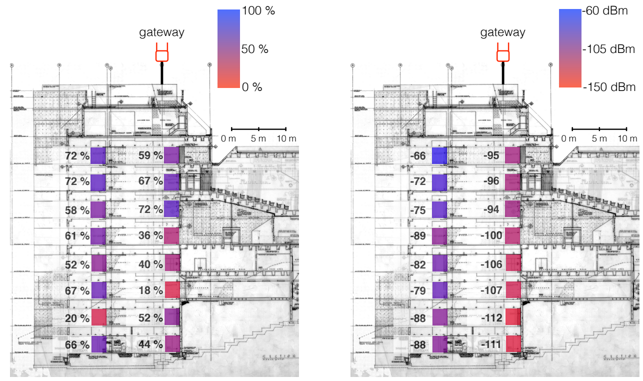
rate increases the spreading factor, which have better sensitivity on the receiver. Thus, LoRa devices communicating at the highest spreading factors (and correspondingly using the lowest data rates) can communicate at much longer range and with higher reliability. The downside is a significant increase in their transmission time which severely affects battery life. This demonstrates that Charm can significantly improve battery life should it allow clients to transmit at higher data rates.

Figure 9 shows a penetration test experiment inside an on-campus poured-concrete building. Despite a gateway present on the roof of the building, the received signal strength varies by as much as 46 dBm at various locations inside the building. A number of client devices, deep inside structure, would have been forced to use the the slowest data rate, but can now benefit from Charm.

8.2 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 10a 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 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



(a) Success rate for bi-directional packet exchange between client device and gateway

(b) RSSI at the gateway for successful transfers

Figure 9: RF signal penetration experiments in a large poured-concrete building

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.

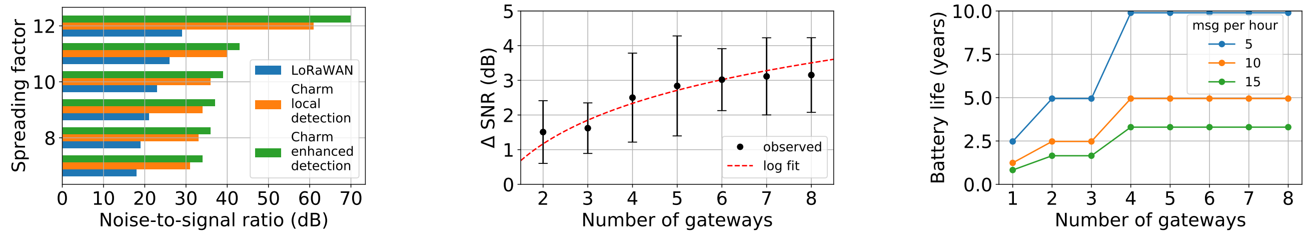
8.3 Diversity Gain

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 10b, reveal remarkable SNR improvements, which logarithmically increases 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 10c 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.

8.4 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.

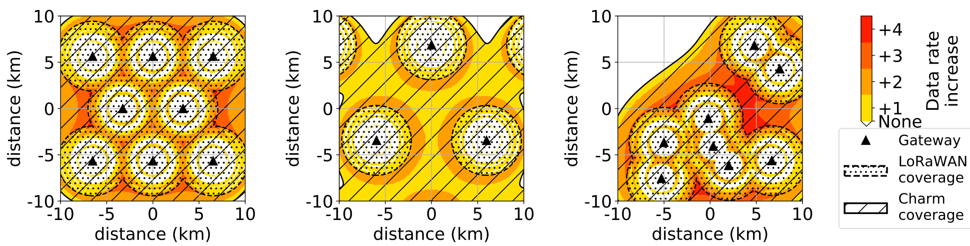


(a) Local packet detection capability for low SNR receptions. Methods using data and preamble symbols outperform preamble-only methods.

(b) Diversity gain: Combined SNR increases logarithmically as more gateways receive a transmission

(c) Client battery life improves as a larger number of receiving gateways permits switching to higher data rates

Figure 10: Benchmarks



(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 11: Improvement in coverage area and data rates due to Charm in three sample deployments: (a) planned dense, (b) planned sparse and (c) random arrangement of gateways.

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

Table 1: Range in congested indoor urban settings

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

8.5 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 in a varied urban environment at different data rates and spreading factors using GPS-connected LoRa client devices. 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 [10] 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 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 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 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 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. The dotted regions show regions which are covered by regular LoRaWAN while the hatched regions are covered by Charm. Imagine the regions with no LoRaWAN coverage having a data rate of DR"-1" = 0 bps (the next data rate is DR0 = 960 bps using SF = 12). The colored patches are regions where Charm shows an increase in data rates, with the darker red areas showing larger improvements than the lighter yellow areas. As seen in each of the sub-figures, Charm shows an improvement in the coverage area (hatched regions are larger than the dotted regions), an increase in client data rates (colored areas inside the dotted regions) as well as both simultaneously (colored areas outside LoRaWAN's dotted area).

Some specific examples of Charm's improvements are as follows: In the planned dense deployment of Figure 11a, Charm improves coverage area by 46% and substantially boosts the data rate around the centroid areas. For the planned sparse deployment of Figure 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 11c, Charm not only improves 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 9. 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 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.

9 CONCLUSION AND FUTURE WORK

This paper 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.

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. Our future work will explore further improvements to network scalability along two dimensions: (1) A full-scale distributed MIMO system atop LPWAN in the cloud, that can also handle collisions from a large number of clients. (2) Offloading of TV whitespace spectrum at peak demand, based on an FCC license recently granted to our university.

ACKNOWLEDGMENT

This research is supported in part by the National Science Foundation under award CNS-1329644 and the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

REFERENCES

[1] A. Checko et al. 2015. Cloud RAN for mobile networksâĂŤA technology overview. *IEEE Communications surveys & tutorials* 17, 1 (2015), 405–426.
[2] A. Del Coso et al. 2007. Cooperative distributed MIMO channels in wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 25, 2 (2007).
[3] C. Liu et al. 2013. The case for re-configurable backhaul in cloud-RAN based small cell networks. In *Proceedings of the 32nd IEEE Infocom,*. 1124–1132.
[4] C. Shepard et al. 2012. Argos: Practical many-antenna base stations. In *Proceedings of the 18th ACM MobiCom*. 53–64.
[5] D. Sabella et al. 2013. RAN as a service: Challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network. In *Proceedings of the Future Network and Mobile Summit (FutureNetworkSummit)*. 1–8.
[6] D. Wubben et al. 2014. Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN. *IEEE signal processing magazine* 31, 6 (2014), 35–44.
[7] E.G. Larsson et al. 2014. Massive MIMO for next generation wireless systems. *IEEE Communications Magazine* 52, 2 (2014), 186–195.
[8] E. Hamed et al. Real-time distributed MIMO systems. In *Proceedings of the 2016 ACM SIGCOMM 2016 conference on Applications, technologies, architectures, and protocols for computer communication*. 412–425.
[9] H.S. Rahul et al. JMB: scaling wireless capacity with user demands. In *Proceedings of the 2012 ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. 235–246.
[10] M. Bor et al. 2016. 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*. 59–67.
[11] M. Centenaro et al. 2016. Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications* 23, 5 (2016), 60–67.
[12] M. Dohler et al. 2004. A resource allocation strategy for distributed MIMO multi-hop communication systems. *IEEE Communications Letters* 8, 2 (2004), 99–101.
[13] G Ramachandran et al. 2017. PnP-WAN: Experiences with LoRa and its deployment in DR Congo. *COMSNETS 2017* (01 2017).
[14] GSM Association. 2016. 3GPP Low Power Wide Area Technologies (White Paper). (2016).
[15] I. Chih-Lin et al. 2014. Recent progress on C-RAN centralization and cloudification. *IEEE Access* 2 (2014), 1030–1039.
[16] Ingenu. 2015. *How RPMA Works*. Technical Report. 1–5 pages. http://www.ingenu.com/get-started/resources/
[17] K. Lin et al. 2011. Random access heterogeneous MIMO networks. In *ACM SIGCOMM Review*, Vol. 41. ACM, 146–157.
[18] K Mikhaylov et al. 2016. Analysis of capacity and scalability of the LoRa low power wide area network technology. In *European Wireless 2016; 22th European Wireless Conference; Proceedings of*. 1–6.
[19] K. Tan et al. 2009. SAM: enabling practical spatial multiple access in wireless LAN. In *Proceedings of the 15th ACM MobiCom*. 49–60.
[20] LoRa Alliance. 2015. *LoRaWAN – What is it? A Technical Overview of LoRa and LoRaWAN*. Technical Report. https://www.lora-alliance.org/portals/0/documents/whitepapers/LoRaWAN101.pdf
[21] M. Ankerst et al. . 1999. OPTICS: Ordering Points To Identify the Clustering Structure. In *Proceedings of the International Conference on Management of Data(SIGMOD)*. 49–60.
[22] M. Centenaro et al. 2015. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *arXiv preprint arXiv:1510.00620* (2015).
[23] M. Hadzialic et al. 2013. Cloud-RAN: Innovative radio access network architecture. In *Proceedings of the 55th International Symposium ELMAR*. 115–120.
[24] M.Sawahashi et al. 2010. Coordinated multipoint transmission/reception techniques for LTE-advanced [Coordinated and Distributed MIMO]. *IEEE Wireless Communications* 17, 3 (2010).
[25] N. Sornin et al. 2015. LoRaWAN Specification. (2015), 1–82. https://www.lora-alliance.org/portals/0/specs/LoRaWANSpecification1R0.pdf
[26] R. Eletreby et al. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In *Proceedings of the ACM Special Interest Group on Data Communication*. 309–321.
[27] R. Sanchez-Iborra et al. 2016. State of the Art in LP-WAN Solutions for Industrial IoT Services. *Sensors* 16 (2016), 708.
[28] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. 2016. NB-IoT system for M2M communication. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*. IEEE, 1–5.
[29] S. Kumar et al. 2013. Bringing cross-layer MIMO to today's wireless LANs. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. 387–398.
[30] S. Marek. 2016. Comcast Will Test LoRaWAN IoT Networks in Two Markets. (2016). https://www.sdxcentral.com/articles/news/comcast-will-test-lora-iot-network-two-markets/2016/10/
[31] T. Petrić et al. 2016. Measurements, Performance and Analysis of LoRa FABIAN, a real-world implementation of LPWAN. (2016).
[32] V. Toldov et al. 2016. Performance evaluation of LoRa radio solution for PREDNET wildlife animal tracking project. In *LPWAN*.
[33] W. Shen et al. 2014. Rate adaptation for 802.11 multiuser MIMO networks. *IEEE Transactions on Mobile Computing* 13, 1 (2014), 35–47.
[34] X. Xie and X. Zhang. 2014. Scalable user selection for MU-MIMO networks. In *Proceedings of the 2014 IEEE INFOCOM*. 808–816.
[35] V. Yenamandra and K. Srinivasan. 2014. Vidyut: exploiting power line infrastructure for enterprise wireless networks. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. 595–606.