This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

Cross Technology Distributed MIMO for Low Power IoT

Revathy Narayanan, Swarun Kumar and C Siva Ram Murthy, *Fellow, IEEE*

Abstract—The Internet of Things (IoT) is scaling rapidly to billions of low power devices, with diverse radio technologies sharing common unlicensed spectrum. Inevitably, this results in rampant cross-technology collisions between the devices that lead to wasteful re-transmissions, draining the battery life of low-power devices significantly. We present CharIoT, the first cross-technology distributed MIMO receiver system that exploits the potential of distributed MIMO to facilitate better co-existence and decoding of a large number of simultaneous low power uplink transmissions from unmodified low-power clients. CharIoT is a recovery-based system that intelligently collects radio samples from teams of light-weight IoT gateways and streams them to the cloud to effectively resolve collisions. At the cloud, CharIoT develops a suite of technology-specific software filters that decouple collisions across diverse technologies, facilitating seamless co-existence across low power radios. An implementation of CharIoT on inexpensive RTL-SDR gateways connected to a Raspberry Pi decodes collisions of four popular IoT technologies in the 868MHz ISM bands – LoRa, XBee, Z-Wave and SIGFOX showing gains in throughput of up to 4× and battery life of up to 3.5 years.

Index Terms—Internet of Things, Low Power, IoT Radio Technologies, IoT Gateway, Distributed MIMO

1 Introduction

Recent studies project the estimate of connected Internet of Things (IoT) devices worldwide to reach around 75.44 billion by the year 2025 [1]. License-exempt Sub-1GHz ISM bands, exclusively reserved for the operation of low power IoT devices [2], play a key role in facilitating this connectivity. This band is inherently heterogeneous, with multiple IoT technologies sharing the spectrum – 900 MHz in the US, and 868 MHz in much of the rest of the world. Much of these technologies – LoRa, XBee, SIGFOX and Z-Wave, to name a few – target battery-constrained devices, supporting low data rates and lacking sophisticated medium access protocols. The result is rampant collisions in these shared low power ISM bands especially when these devices follow 'wake and transmit' model for operation [3]. These collisions are particularly critical in smart buildings, enterprises and factories, where several diverse devices share the spectrum [4]. State-of-the-art solutions to mitigate cross-technology collisions take one of two approaches: (1) The current industry approach is to use gateways with multi-technology radio chips that coordinate diverse and unmodified devices, yet require hardware upgrades to gateways to support new technologies; (2) The academia has seen a recent spurt of cross-technology communication systems [5], [6], [7], that allow radios of one technology to mimic packets of another, especially to avoid collisions, albeit at reduced efficiency [8]. However, there remains a gap for a solution that does not sacrifice energy efficiency of the low power clients while remaining simultaneously upgradable to new technologies.

- *Revathy Narayanan and C Siva Ram Murthy are with the Department of CSE, Indian Institute of Technology, Madras, Chennai.*
- *Swarun Kumar is with Carnegie Mellon University, Pittsburgh.*

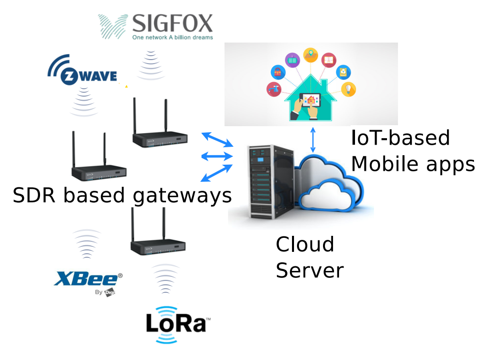


Figure 1: CharIoT: A cross-technology distributed MIMO framework for low power IoT receptions.

We propose CharIoT, to the best of our knowledge, the first *cross-technology distributed MIMO* receiver system that resolves collisions of low power IoT devices across technologies within the Sub-1GHz ISM bands (Fig. 1). To be more precise, CharIoT is a recovery-based system where the gateway receivers recover collisions that inevitably occur from Sub-1GHz low power transmitters relying on wake and transmit model [3]. CharIoT achieves this using a team of programmable RTL-SDR based gateways connected to Raspberry Pis, each costing a few tens of dollars[1] and connected to a wired Ethernet backbone, without modifying clients whatsoever. CharIoT processes received signals across these gateways at the cloud to resolve collisions across low power IoT clients regardless of their radio technology.

While past works discuss the implementation of MIMO and Distributed MIMO in high power context mostly catering to the 2.4 GHz bands [10], [11], [12], CharIoT aims to implement a generalizable distributed MIMO solution across low power technologies in Sub-1GHz ISM bands. Realizing

1. CharIoTś per gateway cost is 60$ while the existing programmable multi-technology gateway platforms cost around 550$ [9].

this design however requires tackling several challenges, which makes CharIoT an IoT-specific design unique from the state-of-the-art collision resolution systems.

(a) Long packet lengths: Synchronizing the team of RTL-SDR gateways is fundamental in realising the distributed MIMO design. Achieving precise time synchronization for low power transmissions can be extremely challenging owing to their packet lengths, which span longer in time domain. In addition, packets in CharIoT stem from diverse technologies, different in lengths and undergoing collisions. CharIoT implements a two-level synchronization – an initial coarse grained synchronization using Network Time Protocol, and later a fine grained synchronization taking into account possible collisions, to achieve precise synchronization.

(b) Channel Estimation: Accurate channel estimation is key to reaping the benefits of the distributed MIMO architecture, and to efficiently detect the number and nature of the colliding transmissions. Practically this requires dynamic channel estimation of the set of radios that collided at any instance across technologies. Doing so is particularly challenging when even the preambles of diverse radio technologies collide. While one could naively request the transmitters to send their preambles in a collision-free manner, this is impractical for low power transmissions which transmit very infrequently and where the cost of synchronization is too high. In addition, low power transmissions have signal powers comparable to noise floors which make traditional channel estimation techniques highly error-prone.

CharIoT facilitates channel estimation dynamically by using a key commonality across low power technologies. All low power transmissions, irrespective of their modulation format, use highly encoded preambles to build redundancy. This redundant information across the received preambles can be coherently combined across gateways to boost the power of the signal even at low signal-to-noise ratio (SNR), despite the interference from other technologies.

(c) Collision resolution in CharIoT: A key innovation behind CharIoT is the development of novel software filters to improve the performance of distributed MIMO amidst collisions across radio technologies. These filters enable decoupling collisions across diverse low power transmissions even if they overlap in time and frequency. These filters exploit the fact that different modulation schemes of signals smear their power across frequencies differently. For instance, technologies that use Frequency Shift Keying smear energy on specific frequencies. Others using chirps transmit energy along frequencies that increase linearly in time. By learning exactly which technologies exist within a collision, one can effectively filter out parts of the spectrum where they focus energy to reduce their interference to other technologies. CharIoT generalizes this approach across diverse classes of low power IoT technologies.

Software filters can outperform traditional techniques like Successive Interference Cancellation (SIC) in decoding a large number of simultaneous transmissions, thereby leading to significant battery gains for low power devices. This is because while SIC relies on power differences across received signals to decode concurrent transmissions, CharIoT's opportunity for a larger transmission decoding stems from the unique nature of low power IoT transmissions. High power transmissions like WiFi provision the device with a higher degree of rate adaptation and support for closely separated data rates, thereby enabling the device to utilize the bandwidth closer to the optimal Shannon capacity limit. But such a data rate flexibility is limited for a low power transmitter due to the additional complexity and cost it entails. Hence low power transmitters commonly transmit at data rates significantly sub-optimal to the Shannon limit. Yet, by decoding collisions across multiple low power transmitters, regardless of technology, one can ensure that, while clients remain below Shannon capacity individually, *collectively* they edge closer to Shannon capacity.

CharIoT can quickly adapt to collisions of new radio technologies using a simple software update from the cloud. Distributed MIMO implementation further boosts the performance of CharIoT by coherently combining the filtered signals received across gateways at the cloud. Further, in cases where these filters fail (for eg., for same technology collisions), distributed MIMO architecture enables CharIoT to resolve collisions using the traditional techniques of MIMO multiplexing and zero-forcing.

Limitations and Scope: : We emphasize that CharIoT (1) Considers static low power clients working in an array of wireless technologies. (2) Focuses primarily on the uplink transmissions across low power transmitters and collision decoupling across them. Downlink transmission and the details on acknowledgement system is out of scope for this paper and is considered as a future extension (see Section 10) (3) Currently shows a proof of concept of a fully operational system using a building-sized testbed. Yet, CharIoT proposes techniques which are extensible and generalizable in large scale futuristic IoT deployments.

Evaluation and Results: We implement CharIoT across two testbeds – (1) a 1,830 m^2 T-shaped indoor environment, (2) a 2680 $m^2\times$ 10 m two-floor building complex. Ten Raspberry Pis equipped with RTL-SDRs and Ethernet backhaul to the cloud form the receiver gateways, deployed in a distributed MIMO setup. The testbeds include simultaneous reception and decoding of transmissions from 16 commodity clients, all working in 868 MHz (EU unlicensed) following four different technologies – LoRa, XBee, Z-Wave and SIGFOX.

- CharIoT decoded simultaneous transmissions providing 4$\times$ throughput gains with ten gateways.
- CharIoT achieved on average a battery life gain of 293.96% (about 3.5 years) across technologies.

Contributions: This paper presents CharIoT, the first cross-technology distributed MIMO receiver system to alleviate uplink collisions across low power IoT radio technologies. CharIoT involves a team of cloud assisted gateways, detecting collisions from low power transmissions and shipping their corresponding radio samples to cloud. The gateways are synchronized using CharIoT's specialized synchronization algorithm that is generalizable across low power transmissions. At the cloud, CharIoT develops novel software filters to disentangle collisions received across low power IoT technologies, based on the properties unique to their modulation. We implement a prototype of CharIoT on inexpensive RTL-SDR gateways and demonstrate simultaneous decoding of collisions across four popular low power IoT technologies in large indoor testbeds.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

2 Related Work

Related work can be broadly categorized into three:

Cross technology communication (CTC): With multiple technologies occupying the ISM bands, bridging their diversity and enabling their co-existence by facilitating cross-technology communication has been a widely studied solution in the research community. Over the past decade, multiple techniques to facilitate CTC, despite the physical incompatibilities of technologies, have been proposed [5], [6], [7], [8], [13], [14], [15]. Solutions in this domain have primarily focused on packet level modulation including packet length [5], timing [6] and energy/data traffic patterns [7], [13]; thereby enabling software solutions that allow the devices to cross-talk without modifying the legacy hardware. But due to the inevitable loss in efficiency when transforming one modulation to mimic another, the industry has continued to favor dedicated multi-technology gateways to mediate cross-technology communication.

Cross technology collision mitigation: Apart from CTC, solutions addressing cross technology collision resolution using a multitude of techniques have been well studied in the past literature [16], [17], [18], [19], [20], [21]. Even though the initial work in this domain focused on hardware modifications [22], [23], more recent ones have proposed effective software-based solutions [4], [24] that still require computation at the clients, modifications to the client hardware or assuming high-power clients (e.g. WiFi). Unlike prior work, CharIoT focuses exclusively on recent low-power IoT standards without modifying the client hardware.

Distributed MIMO: Distributed MIMO solutions have been well addressed in the past focusing on scaling gains, diversity gains and improved system performance [10], [12], [25], [26], [27]. Much of this work focused on high power context (e.g. WiFi and cellular) [11], [24], [28], [29] with some efforts in the low-power space [25], [30] that do not explicitly target collisions. Some recent efforts tied to LoRa [3], [31] tackle collisions between the chirp-spread spectrum transmitters. Similarly, systems like ZIMO [11] and mZig [32] resolve collisions across low-power ZigBee transmissions by reaping the benefits of MIMO based architecture. While these systems implement oversampling at receivers to receive more number of ZigBee transmissions, CharIoT caters to a diverse number of radio technologies unique in their own sense. Therefore, CharIoT can be considered as the first-of-its-kind study that provides a generalizable cross-technology distributed MIMO to resolve collisions across low power technologies.

CharIoT is an extension of our recent system GalioT [33] that decodes cross-technology collisions at a single gateway through software filters. However, there were several challenges that a single radio could not resolve.

1) Larger number of technologies colliding at the same time in a more noisy environment can reduce the efficiency of software based decoding if received only at a single gateway.
2) Near-far effects are more prominent with the usage of a single gateway. The transmissions too close to the gateway can get clipped while transmissions from far can get buried under the noise floor.
3) Single antenna systems cannot separate collisions within short range technologies with smaller symbol periods.

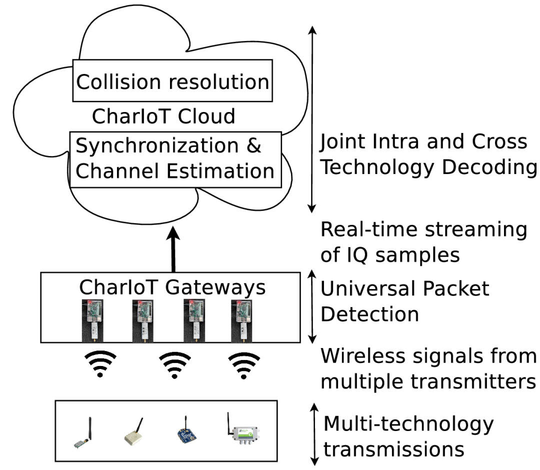


Figure 2: CharIoT architecture.

3 CharIoT – An Overview

CharIoT aims to decode uplink data streams from potentially unsynchronized weak transmissions within and across radio technologies that operate in low power ISM bands. CharIoT achieves this by gathering received I/Q samples from a team of gateways. These gateways intelligently detect collided transmissions and deliver them to the cloud, where they are collated to recover individual transmissions. CharIoT's design allows the RF-frontends at its gateways to be light-weight and inexpensive– a $20 RTL-SDR connected to a Raspberry Pi and an Ethernet backhaul. Recent literature depicts how the RTL-SDR dongles despite being in default receive only mode can be hacked to work in transmit mode [34] thereby enabling the transmission of acknowledgement packets. An alternate easier option is to use slightly more expensive SDRs which can work as transceivers [35].

CharIoT's goal is to maximize network throughput and avoid power-intensive re-transmissions from low-power devices, while conserving the backhaul bandwidth. In particular, CharIoT aims to recover the maximum possible uplink transmissions, greater than the collective number of antennas, across gateways. Specifically, CharIoT relies on the fact that low-power IoT radios often transmit at data rates much lower than their Shannon capacity. This is because selecting between a large number of data rates fundamentally makes the transmission system complex. CharIoT develops unique software filters that decouple collisions of transmissions from multiple low-power technologies, even if they are received concurrently at a single-antenna. We then develop a first-of-its-kind cross-technology distributed MIMO solution that generalizes this approach by concurrently processing received signals across multiple gateways at the cloud. Figure 2 presents an overview of CharIoT's architecture. The rest of this section summarizes the key system design challenges at both the gateway and the cloud in realizing this architecture:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

CharIoT Gateway: At each gateway, CharIoT streams samples from received collisions to the cloud in real-time. Compared to traditional wireless contexts such as cellular or WiFi (bandwidths of few tens of MHz), low-power IoT technologies operate at significantly lower bandwidths (few hundred kHz) making such a design possible with a light-weight Ethernet backhaul. Yet, despite the limited bandwidth requirement, CharIoT radios would still need to stream megabits of data per second which can pose immense strain on the network. This motivates the need for CharIoT to carefully inspect received samples to send only the collided signals to the cloud, while processing non-collisions locally and discarding the received ambient noise.

CharIoT addresses this challenge by building a universal preamble that can be correlated with the received signal to detect any collision across technologies. In other words, our universal preamble is designed to correlate well with the preambles of all IoT radio technologies we intend to decode. To see why a universal preamble is possible at all, we study the preambles of various radio technologies and make two observations: (1) First, many radio technologies often use similar preamble sequences. This is by no means an accident, given that a few simple sequences exist that are amenable to correlation (i.e. correlate poorly with noise and well with signal). (2) Second, some pairs of preambles are mutually orthogonal. This again is intentional, to avoid erroneously confusing packets of one technology as that of another. Motivated by these observations, CharIoT constructs a universal preamble that is a combination of key preambles that are mutually orthogonal. Sec. 4 details our approach, as well as mechanisms to optimize detection across gateways.

Pre-processing at the Cloud: At the cloud, CharIoT develops a variety of techniques to estimate wireless channels and synchronize transmissions across technologies and base stations. A key challenge CharIoT tackles is the need to isolate the preamble of packets belonging to any given radio technology, even as it collides with the data (or preamble) of packets from other technologies. Sec. 5 describes how we isolate these preambles to estimate wireless channels and synchronize collisions received across base stations.

Collision Mitigation at the Cloud: Cloud processing allows CharIoT to enhance the decoding of various low power transmissions, despite collisions within and across radio technologies. To decode collisions, CharIoT uses two approaches: (1) Software filters to separate collisions of different radio technologies; (2) Multiplexing gains across synchronized gateway antennas to decouple same technology collisions.

First, we develop software filters that account for the differences in energy spread of radio technologies over the spectrum. For instance chirp-based technologies encode information by spreading energy across frequencies that increase over time, while frequency modulation focuses energy on specific discrete frequencies. We use these differences in where useful data is concentrated within the received spectrum to greatly reduce cross-technology interference. Finally, CharIoT uses the multiplexing gains of distributed MIMO to further increase the number of concurrent received transmissions. Specifically, we rely on the principle that in general, n synchronized antennas can decode up to n concurrent transmissions. Sec. 6 elaborates on how we harmonize these diverse collision mitigation techniques to decode collisions across radio technologies.

Table 1: IoT technologies under consideration with their modulation and preamble information in Sub GHz bands.

Technology	Modulation	Sync Preamble
LoRa [38]	CSS	sequence of 1s
Z-Wave [39]	BFSK,GFSK	m bytes '01010101'
XBee [40]	GFSK	4 bytes '01010101'
SIGFOX [41]	D-BPSK	4 bytes configurable

4 CHARIOT AT THE GATEWAY

To be efficient, CharIoT must detect and process the received Radio Frequency (RF) samples, including collisions, in real-time. Though well-structured implementations at the cloud can facilitate this to an extent, this architecture is highly reliant on the streaming bandwidth of the backhaul. Indeed, even narrow band technologies transmitting at a mere few hundred kilohertz of bandwidth can generate gigabits of I/Q sample streams, posing immense strain to the backhaul if required to ship the samples in real-time. To make CharIoT operate efficiently, even with typical home cable backhauls that offer modest bandwidths, we therefore design a packet detection scheme at the gateway that pre-processes the received signals to vastly reduce unwanted samples shipped to the cloud. This mechanism is defined to identify and locally process any regular non-collided received signals and ship only collisions to the cloud, while discarding noise.

While there are several techniques to perform local collision detection, developing a methodology that can systematically identify packets across radio technologies while remaining scalable poses new challenges. First, simple energy based thresholding [4] is not compatible with low-power technologies where the signal powers are comparable to, and often below the noise floors. Second, correlation with each known preamble across technologies is computationally expensive [36], especially when scaled to a large number of technologies. Even for a smaller number of technologies, multiple correlation computations while simultaneously streaming in megabits of data can bottleneck the memory constrained RPi based gateways thereby resulting in sample drops[2]. Thus, CharIoT strives to achieve the same computational expenditure for accurate detection of collisions, irrespective of the number of colliding technologies. Table 2 depicts the key difference in the packet detection performed at CharIoT's gateway as opposed to that of the existing gateways.

4.1 Universal Preamble

Motivated by the above two considerations, CharIoT introduces the concept of a universal preamble. The idea of a universal preamble is very simple – a preamble that is no

2. RPi 3 Model B+ has a maximal clock frequency of 1400 MHz, with each instruction taking a minimal of 40 clock cycles [37]. For data streaming at 1 Mbps from RTL-SDR, fetching each individual preamble from the memory by itself can consume up to 100 × 40 clock cycles, rendering individual correlations computationally very expensive.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

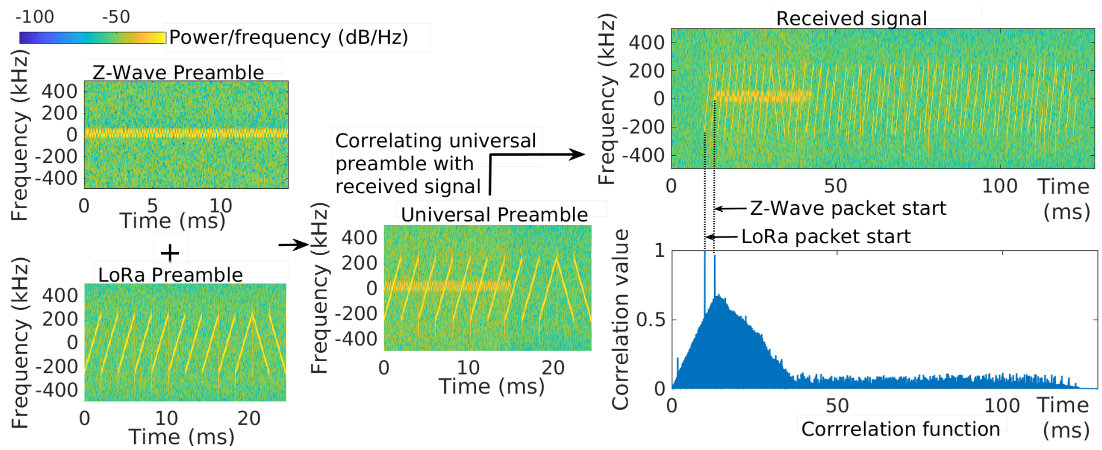


Figure 3: An example depicting construction and working of universal preamble with two technologies– LoRa and Z-Wave.

Table 2: Detection at the gateway – Existing v/s CharIoT.

State of the art	CharIoT
Correlates each preamble individually for each technology at the gateway	A global 'universal preamble' for correlation across technologies at the gateway

longer than a single preamble pattern that can correlate across multiple radio technologies even in the event of a possible collision. This ensures that the system's computational complexity will remain unchanged, even when newer technologies are introduced into the system, with the additional benefit of scalability through a simple software update.

Our design of the universal preamble is based on key observations from a thorough analysis of existing low-power IoT technologies. As observed from Table 1, we can see that the current low power standards satisfy one of the three properties for their existing preamble sequences: (1) Technologies following similar modulation schemes generally follow the same sequence of preambles. This is, by no means, an accident, since these patterns are carefully chosen to be simple and amenable to correlation or energy detection and only a few such sequences exist for short lengths. (2) Technologies with different modulation schemes ensure orthogonality across their preamble patterns. This again enables their receivers to avoid erroneously confusing packets across technologies. (3) Some of these technologies support configurable preambles, which for instance, can be tuned to correlate with the universal preamble.

Inspired by these observations, CharIoT's universal preamble is constructed as the following – First we coalesce the shortest representative from among a group of technologies following common preamble patterns. Second, we sum up all these representative preambles, which forms the universal preamble, that can correlate with all the technologies under consideration. Note that this involves addition of all representative preambles in the time domain *irrespective* of their center frequencies[3]. An example depicting the construction and packet detection using universal preamble for two technologies is shown in Fig. 3. To reduce the detection overhead, CharIoT also uses a collaborative detection across gateways using prior historical information. Specifically, each gateway detecting a collision estimates the historical likelihood that other gateways around it may have also received this collision. This automatically triggers receptions from surrounding gateways to be uploaded to the cloud, even when one of the gateways detects the presence of a collided signal.

3. To prove this, the microbenchmark on detection (Section 9.2) detects Z-Wave transmissions centered at 868.4MHz as well while the rest of the technologies are centered around 868MHz.

4.2 Analysis of Detection

In this section, we mathematically define the universal preamble, discuss its working, and analyze its performance. Let there be m technologies in the system and P_j represent the preamble corresponding to the technology j; $1 \leq j \leq m$. The general property of preamble dictates that their auto-correlation function should produce peak at zero and have negligible values or 'noise' elsewhere (see Fig. 4). Let this peak value be V_j.

If two technologies T_i and T_j share a similar modulation scheme, then the properties of their preambles are also similar. As a result the correlation of their respective preambles C_{ij} behave very similar to their auto-correlation functions. The generic property of preambles chosen for the technologies also dictates, for all other cases where the modulation schemes are not similar to one another, the preambles should be relatively uncorrelated to each other. That is the correlation C_{ij} of preambles P_i and P_j does not produce an unambiguous and significant peak anywhere thus rendering it relatively flat throughout the indices. Let this flat value be denoted by N_{ij}.

Let M represent the maximum-sized set of mutually uncorrelated preambles. Then the universal preamble U is defined as: $U = \sum_{j \in M} \tilde{P}_j$, where $\tilde{P}_j$ is the preamble P_j zero-padded at the end so as to set its length to the maximum preamble length across all technologies. Even with a different length, the zero-padding at the beginning ensures that both $\tilde{P}_j$ and P_j produces the same peak value and peak location when they are independently used for the detection of a signal. Let $C_{\{U\}j}$ be the correlation of the universal preamble U with a preamble P_j. From the definition of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

the universal preamble and zero-padded preambles, the distributive property of correlation gives us,

$$\begin{aligned} C_{\{U\}j} &= \sum_{i \in M} C_{ij} \\ &= C_{jj} + \sum_{i \in M, i \neq j} C_{ij} \end{aligned}$$

Here, the cross correlation sum produces only noise hence producing a result with a peak similar to that of C_{jj}. This extends to the case when the universal preamble is used to detect the start of a particular technology based signal. That is, for a signal generated using technology T_j, the universal preamble returns the same unique spike that is produced if P_j is used instead, as long as the sum total of the floor noises produced by the cross-correlations is smaller than the peak value V_j.

But the question remains – how scalable is the universal preamble? Or rather, what is the limit at which the universal preamble fails? From the above equation, the universal preamble fails to detect the packet start in technology T_j when $\sum_{i \in M} N_{ij} \geq V_j$. Since auto-correlation of identical preambles produces almost negligible noise, such a condition can arise only under two circumstances: 1) When the constituent preambles of the universal preamble are neither in perfect correlation nor are completely orthogonal – both the cases being in contradiction to current standards. 2) When number of technologies in the system is too high at about $\frac{V_j}{\sum_{i \in M} N_{ij}}$. This is practically a high value, since the peak value generated by the auto-correlation is generally very high in comparison to the noise floor generated by their auto-correlation or cross correlation.

To reduce any potential cumulative effect of noise introduced by the hardware properties of devices, CharIoT also normalizes each individual preamble with respect to its length before constructing the universal preamble. This preserves comparable powers for individual correlation peaks which are well distinguishable from noise components. Once the signal is detected, CharIoT conservatively ships samples corresponding to twice the maximum packet length across technologies around the detected preamble to the cloud.

Our evaluation in Sec. 9.2 constructs such a universal preamble for four common IoT technologies LoRa, SIGFOX, XBee, and Z-Wave, the first two using Chirp Spread Spectrum and Phase Shift Keying respectively, and the last two following Frequency Shift Keying.

5 Pre-processing at the Cloud

Prior to combining transmissions across gateways at the cloud, CharIoT needs to answer two questions: (1) First, how do we time synchronize receptions across gateways to correctly decode signals from any given transmitter? (2) Second, how many – and what kind of transmitters exist across the signals uploaded from various gateways? This section describes our approach to answer these questions: time synchronization and channel estimation.

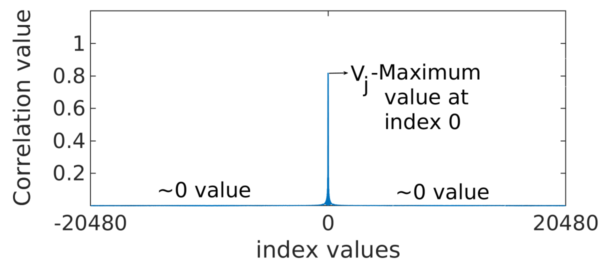


Figure 4: The figure depicts a sample auto-correlation for the preamble P_j which generates the peak V_j at the 0^{th} index. The value of auto-correlation is negligible elsewhere.

5.1 Synchronization at the Cloud

Achieving distributed MIMO requires precise time synchronization which is highly challenging in a low power cross technology context. To achieve coarse synchronization, CharIoT's gateways enable Network Time Protocol (NTP) based time synchronization but this is limited to millisecond accuracy, thereby failing to offer sample level synchronization needed for distributed MIMO.

CharIoT achieves fine-grained synchronization across gateways by looking for preambles of identical radio transmissions over the received samples. Specifically, CharIoT uses the universal preamble to identify the start of received packets over a modest time-window over which NTP remains accurate (few milliseconds). It then repeats this process across gateways. Should one collision be detected over this window, CharIoT can directly map the timing offset between any pair of gateways as the offset between the peaks of the correlation. CharIoT uses the DTW algorithm [42] to compute this offset, should the correlation produce multiple peaks.

One might wonder: what if multiple collisions are received over a few millisecond intervals? Note that this is relatively unlikely, given that low-power transmissions are typically at extremely low data rates when compared to Wi-Fi. Should an ambiguity between two (or more) offsets occur regardless, CharIoT cross-correlates the received samples at either of these offsets across gateways to synchronize the two transmissions. We note that this approach requires at least one low-power transmission to be detectable across gateways – a reasonable assumption given that the collisions are already detected (Sec. 4). It also at best would enable sample-level synchronization. However, sub-sample synchronization is required for distributed MIMO. CharIoT compensates for the additional phase and magnitude shifts of signals (due to sub-sample offsets) by accurately estimating wireless channels. We describe our approach to do so in the following section.

5.2 Channel Estimation

To separate collisions between radio transmitters, CharIoT needs to estimate wireless channels from individual radios within a collision. Despite vast literature on dynamic channel estimation techniques, CharIoT's cross technology low power paradigm makes channel estimation challenging for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

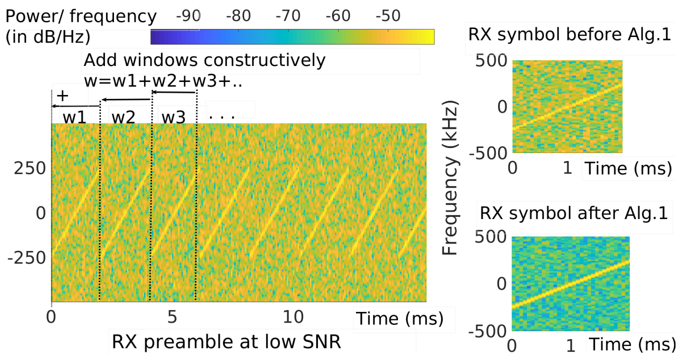


Figure 5: An example demonstrating Alg. 1 in LoRa.

three reasons: (1) First, signals considered here are often closer to noise floors which make channel estimations highly error prone. (2) Second, in cross technology environments, any technique provided should generalize across technologies. (3) Lastly, the estimation technique should work despite the collision of preamble.

CharIoT facilitates accurate dynamic channel estimation by making a key observation applicable across low power transmissions. Low power transmissions, irrespective of the modulation scheme, use long highly encoded preambles that are redundant in nature. One can therefore exploit these repeated patterns to enable coherent combination of preamble symbols thereby boosting its power above the noise floor. In addition, coherent addition of repeated symbols of the signal under consideration only boosts its power from within a collision, hence enabling its channel information to be deciphered.

Table 3: Channel Estimation – Existing v/s CharIoT.

State of the art	CharIoT
Received preamble is compared as such with the transmitted preamble	Coherent addition of repeated preamble symbols to boost its power before comparing with transmitted preamble

This concept can be better understood from the example given in Fig. 5. The figure shows a highly noisy LoRa preamble received at CharIoT's gateway. LoRa preamble consists of repeated up-chirp symbols as can be seen from the figure (each up-chirp is a frequency sweep from the lowest to the highest possible frequency). Considering each symbol as a window, the set of windows $\{w1, w2, w3 \ldots\}$ can be added coherently to provide a less noisy received symbol, which facilitates a more accurate channel estimation.

Implementing this technique in a practical system has its own challenges. Specifically, the effect of hardware imperfections at the transmitter and receiver leads to offsets between adjacent symbols [3]. This offset needs to be compensated beforehand to enable coherent combination of redundant preamble patterns.

CharIoT's algorithm to estimate channels within collisions takes a three pronged approach: (1) First, we divide adjacent windows of the received signal by the transmitted preamble ,in the Fourier domain, to obtain a coarse estimate of the channel, which still includes frequency and timing offsets; (2) Second we cross-correlate windows of channels to estimate the slope and intercept of the phase shifts between them, which correspond to frequency and timing offsets respectively; (3) Third, we compensate for this slope and intercept across all received channel estimates (this implies compensating the hardware offsets) and add them up constructively.

An important aspect of our approach is the choice of window size. The window size being too large can cause the phase variation due to time varying offset to wrap around 2π, making the channel calculations inaccurate. But the window value being too small fails to capture the offset variations with sufficient resolution. To make the channel calculations more accurate, the same channel calculation can be repeated iteratively decreasing the window size granularity every time. Table 3 points out the key differentiating factor in CharIoT's channel estimation technique as opposed to the state of the art solution and Algorithm 1 summarizes our approach.

Algorithm 1 Channel Estimation at the cloud

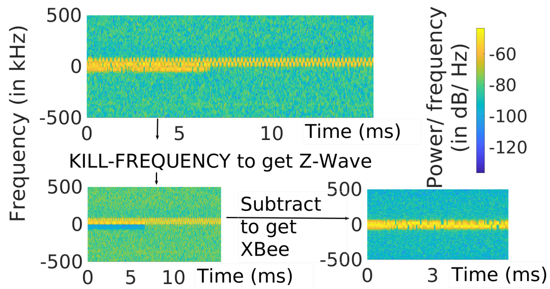
```
1: procedure CHANNELEST(τ, ρ, w)          ▷ τ-Transmitted preamble,
   ρ-Received preamble,w-window size,t-sample time
2:     Abs = rms(ρ)/rms(τ)                 ▷ Absolute value of channel
3:     nw = ⌈length(ρ)/w⌉                  ▷ nw-number of windows
4:     H = ρ./τ                            ▷ Coarse channel calculation
5:     H = [h_1 h_2 ... h_nw], h_i = H[(i−1)w : i∗w]   ▷ channel windows
6:     f_off = 1/(2πt[1:w]) ∠ 1/(nw−1) Σ_{i=1}^{nw} (h_i .∗ h_(i+1))   ▷ frequency offset
7:     ρ = ρ ∗ e^{i2πf_off t}              ▷ Compensating offset
8:     if f_off ≠ 0 then
9:         ρ_new = Σ_{i=1}^{nw} ρ[(i−1)w : i∗w]   ▷ Constructive addition
10:        Update w, ρ = ρ_new, τ = τ[1 : w], Repeat from step 3
11:    ph_off =mean(∠Σ_{i=1}^{nw} (h_i))   ▷ Phase offset
```

6 Collision Decoding at the Cloud

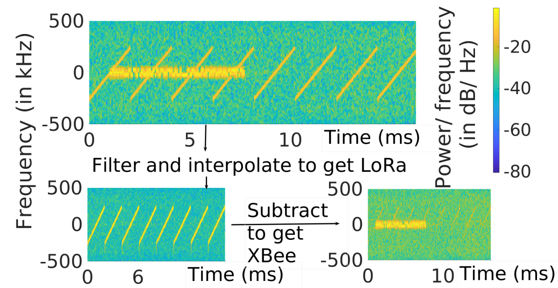
The key agenda of CharIoT is to disentangle all instances of collisions at the cloud, within and across multiple radio technologies. CharIoT's unique architecture develops a technology-agnostic methodology that combines the benefits of two complementary solutions: (1) software filters; (2) exploiting MIMO multiplexing gains of distributed MIMO, with signal copies received across antennas. While the first decouples collisions across radio technologies, the latter provides robust collision resolution even for transmissions of the same radio technology. Hence, as mentioned in Table 4, while the existing multi-technology gateways still rely on successive interference cancellation for disentangling collisions, CharIoT's gateways use a stream-lined methodology based on the nature of collisions to reap the maximum collision separation. We elaborate these solutions below.

6.1 Software Filters

CharIoT develops specialized software filters designed to disentangle cross-technology collisions in low power paradigms. We call these 'kill' filters since they are intended to kill a specific radio technology based on its modulation. The key idea behind software filters is simple – collisions across technologies, despite shared center frequency, smear their powers differently across frequencies. This allows us to develop filters that eliminate energy spread across certain frequencies while preserving others. This is particularly useful for cross technology collisions where modulation based differences make the frequency differences more prominent.



(a) KILL-FREQUENCY separating XBee & Z-Wave



(b) KILL-DSS separating LoRa & XBee

Figure 6: Collision mitigation in CharIoT using software filters.

CharIoT implements two of such filters popular across the modulation schemes of low power technologies.

(1) KILL-FREQUENCY: CharIoT observes that many low power technologies distribute energy unequally even within the same bandwidth; that too in a few frequency bands. Modulation schemes specified by short range low power standards like ITU-T and IEEE 802.15.4g adopt Frequency Shift Keying (FSK), which distributes powers over two or few frequency bands making them separable in the frequency domain. Same is the case for phase based modulation schemes where an energy spike is observed in one of the frequencies while maintaining low power in others. Three popular low power technologies – XBee and Z-Wave in short range category, and SIGFOX in long range category, use these frequency and phase based modulations respectively. CharIoT exploits this to filter out these specific frequencies to eliminate their signals. Figure 6a depicts an example for KILL-FREQUENCY filter for the separation of XBee and Z-Wave. XBee is killed first to recover Z-Wave, which in turn is subtracted from collision to retrieve back XBee.

(2) KILL-DSS: Direct Sequence Spread Spectrum (DSS) implementations in low power context provides high level of noise tolerance for the signal, along with providing a significant degree of collision resiliency. Modulation schemes like Chirp Spread Spectrum (CSS) (a derivative of DSS used in LoRaWAN) hence offers higher immunity across other narrow band interferers. These kinds of modulation schemes offer a unique challenge for collision mitigation since it distributes energy evenly across frequencies making kill-frequency filter infeasible. CharIoT uses a unique method for separating collisions from these technologies. Since each symbol in this category of technologies is spread across frequencies, CharIoT considers the uncollided frequency portions to interpolate the collided portion of the symbol. For instance, in the case of CSS, each symbol is encoded as different cyclic frequency shifts of the elementary chirp that runs from the lowest to highest frequency across the bandwidth. The received sequence when multiplied with a sequence of inverted elementary chirps (that run from the highest to the lowest frequency of operation) can result in a product that appears similar to the narrow band signal reception centered at frequencies corresponding to the starting frequencies of various chirps. The signal thus obtained can be cancelled out from the received signal, akin to the KILL-FREQUENCY approach. Figure 6b shows decoupling of LoRa and XBee from collision using KILL-DSS filter.

Table 4: Collision Decoding – Existing v/s CharIoT.

State of the art	CharIoT	
SIC	Software Filters	Zero forcing
Relies on decoding all collisions based on power levels	Collision separation based on differences in modulation and frequency offsets	Collision decoding using time offsets across receptions if software filters fail
Fails to decode signals with similar power levels	Works on decoupling collisions across signals having comparable power levels	

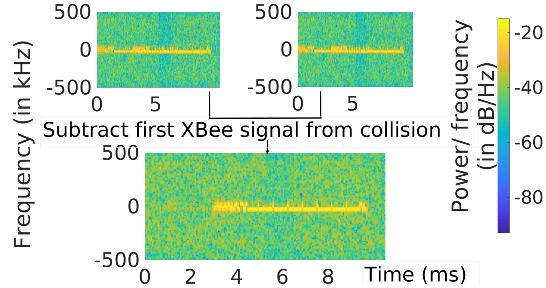


Figure 7: Interference nulling in XBee.

Applying these filters across individual receptions can improve the decoding accuracy [33] but larger number of technologies colliding at the same time in a more noisy environment can reduce the efficiency of software based decoding. This is where the advantage of CharIoT's multiple receptions come in. CharIoT coherently combines filtered data from multiple receptions thereby cancelling out the effect of noise while filtering. This enables CharIoT to filter and decode larger concurrent collisions at lower SNRs.

6.2 Exploiting MIMO Multiplexing Gains

To provide a more complete solution, CharIoT combines some existing techniques that enable separation of collisions within the same technology. Specifically, CharIoT builds on device-specific hardware offset based separation [3] and distributed MIMO zero forcing (see Fig. 7) to enable collision separation from same technology packets. Once separated, the decoupled signal from several gateway receptions are coherently added to boost its SNR in logarithmic scale.

Zero forcing or filters – which to apply when? Despite zero forcing being a more generalizable technique that decouples collision irrespective of whether the constituent signals are from same or different technologies, the technique comes with its own limitations. First, wide-area technologies like LoRa and SIGFOX have symbols that last for long duration

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

(SIGFOX packet lasts for more than 2 seconds), which leaves well distinguishable natural hardware offsets. Hence, rather than relying on zero forcing, the collisions within these technologies can be easily filtered using their hardware introduced offsets. Second, counter-intuitively, coherent combination of residual signals after software filters (also hardware offset filters) may often provide a much better SNR gain than zero-forcing. This is because filters (software/ hardware offset based) disentangle collisions within a single antenna while zero forcing sacrifices one antenna per signal during collision. Hence the former allows residuals from all receive antennas for coherent combination giving a better gain compared to the latter. However, zero-forcing plays an important role as well: It remains the only method that can separate collisions across signals across radio technologies when the symbol duration is small or when software filters otherwise fail (e.g. due to noise).

Putting it all together: For a given received signal, CharIoT executes Algorithm 2 at the cloud. CharIoT first attempts to correlate with individual preambles to estimate wireless channels at the peaks of where the correlations result in spikes (see Sec. 5, step 4 in Alg.2). It repeats this process across received signals to synchronize the different receptions across gateways (see Sec. 5) and tries to decode any possible signals through a simple SIC (steps 5-8 in Alg.2). Once an initial synchronizing and decoding is done, CharIoT attempts to use software filters to kill the existing radio technologies in the transmission one at a time and repeats the process (steps 9-11 in Alg.2). We choose the signals with minimum signal spread in the frequency domain to be eliminated first (BPSK having the minimal and CSS having the maximal frequency spread in our case). This not only improves the performance of interpolation but also aids in SIC during future iterations by removing noise to the easiest decodable signal (better frequency spread improves noise resiliency) in the reception. Should software filters fail – as detected by an invalid CRC – CharIoT attempts two standard cancellation techniques that apply across technologies as a fall-back: (1) *Separate using hardware offsets:* Among the remaining collided signals, similar looking packets are filtered along device-specific timing and frequency offsets to find their hardware based separations akin to [3] (steps 13-14 in Alg.2) (2) *Zero forcing:* Should all of these steps fail (i.e. CRCs fail), CharIoT applies distributed MIMO zero-forcing, which projects the received signal along the space orthogonal to a reception and repeats the process (steps 15-16 in Alg.2). As before, we choose to do this for the weakest first and then repeat for other receptions should this fail. Mathematically, we write $y_{isolate} = \sum_{i=1}^{n} h_i^{+*} y_i$, where $[h_i^{+}]_{i=1,\ldots,n}$ denotes the vector that is orthogonal to $[h_i]_{i=1,\ldots,n}$.

Complexities in processing collisions: An important question that might arise here is whether too many receivers imply additional computation complexity for collision resolution. CharIoT receivers, being positioned randomly across geographical locations bring in an inherent physical limit on the locality. This means collisions can be well-perceived only by certain receivers which are nearer to the transmitting clients. Only the receptions from these receivers are processed at any point of time at the cloud.

Algorithm 2 CharIoT's collision separation algorithm.

1: **procedure** $(y_1, y_2, \ldots y_n)$ ▷ y_i-Collision at i^{th} gateway, $y_i = h_{i1}x_1 + h_{i2}x_2 + \ldots h_{im}x_m$, $S(x_j)$-frequency spread of x_j
2: $\mathcal{X} = \{x_1, x2, \ldots x_m\}$, decodable=True
3: **while** $(\mathcal{X} \neq \phi)$ or (decodable == True) **do**
4: $\forall x_j \in \mathcal{X}$ attempt to estimate h_{ij} using Algorithm 1
5: **if** Estimate(h_{ij})=True **then**
6: Calculate $(y_{combined})_j = \sum_{i=1}^{n} h_{ij}^{*} x_j$
7: **if** Decode(x_j) = True **then**
8: $y_i = y_i - h_{ij}x_j, \mathcal{X} \leftarrow \mathcal{X} - x_j$ ▷ SIC
9: **if** $S(x_j) < S(x_k), \{x_j, x_k\} \in \mathcal{X}$ **then** ▷ different technology
10: $kill(x_j), \mathcal{X} \leftarrow \mathcal{X} - x_j$, Interpolate x_k if isDSS(x_k)=True
11: Repeat steps 4-8
12: **for** $x_j \in \mathcal{X}$ **do** ▷ signals with same spread
13: **if** OffsetFilter($\mathcal{X}$)=True **then**
14: separate $\mathcal{X}$, Repeat steps 4-8
15: **else** Calculate $y_{isolate} = \sum_{i=1}^{n} h_i^{+*} y_i$
16: Resolve x_j using zero forcing, Repeat steps 4-8
17: **if** $\mathcal{X} \neq \phi$ **then** decodable =False;
18: Decode any killed signals by subtracting decoded signals from the combination

7 Implementation and Evaluation

CharIoT captures the co-existence of uplink transmissions across four radio technologies popular in low power ISM bands- LoRa, SIGFOX, Z-Wave and XBee. We deploy around 16 transmitters in different combinations to capture their interactions and show the performance of our algorithms.

Radio technologies in CharIoT: The radio technologies chosen by CharIoT span wide ranging technologies in Sub-GHz ISM bands, both long and short-range. While LoRa and SIGFOX offer kilometer range connectivity, XBee and Z-Wave are meant for shorter range M2M communications that facilitate applications like smart automation. The four technologies we use employ completely different modulation schemes: (1) LoRa (Chirp Spread Spectrum) [43], [44]. (2) SIGFOX (Binary Phase Shift Keying) [45], [46], [47], [48]. (3) XBee (IEEE 802.15.4g PHY) [40], [49], [50], [51], [52] (4) Z-Wave, which follows the ITU-T standard (i.e. FSK) [39], [53] (refer Table 5).

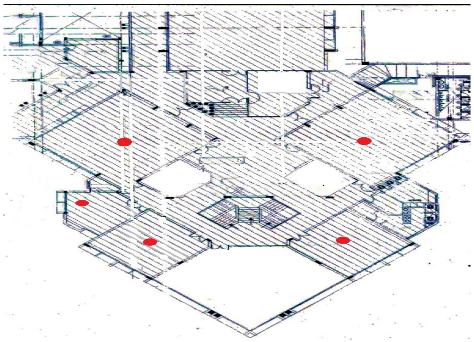
Table 5: Parameters for technologies in CharIoT.

<table>
<tr><th>Technology</th><th>Bit Rates</th><th>RSSI limits</th><th>TX Current</th></tr>
<tr><td>SIGFOX</td><td>100bps-EU
600bps-AUS
No Rate Adaptation</td><td>≤-135dBm- weak
-122 to-135dBm-good
≥-122dBm-excellent
limited to 3 retransmits</td><td>60mA-EU
240mA-AUS</td></tr>
<tr><td>LoRa</td><td><table><tr><th>SF</th><th>125kHz in kbps</th><th>500kHz in kbps</th></tr><tr><td>7</td><td>5.5</td><td>21.8</td></tr><tr><td>8</td><td>3.1</td><td>12.5</td></tr><tr><td>9</td><td>1.8</td><td>7.03</td></tr><tr><td>10</td><td>0.98</td><td>3.9</td></tr><tr><td>11</td><td>0.54</td><td>2.14</td></tr><tr><td>12</td><td>0.3</td><td>1.17</td></tr></table>Rate Adaptation Allowed</td><td><table><tr><th>SF</th><th>125kHz in dBm</th><th>500kHz in dBm</th></tr><tr><td>7</td><td>-126.5</td><td>-120.75</td></tr><tr><td>8</td><td>-127.25</td><td>-124</td></tr><tr><td>9</td><td>-131.25</td><td>-127.5</td></tr><tr><td>10</td><td>-132.75</td><td>-128.75</td></tr><tr><td>11</td><td>-134.5</td><td>-130.75</td></tr><tr><td>12</td><td>-135.25</td><td>-132.25</td></tr></table>configurable retransmits</td><td>SX1272
18mA to
125mA

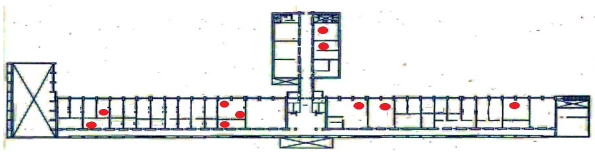
SX1276
20mA to
120mA</td></tr>
<tr><td>IEEE
802.15.4g
PHY
XBee</td><td>50kbps-R1
100kbps-R2
200kbps-R3
Rate Adaptation Allowed</td><td>≤-109dBm- R1
-106dBm-R2
-96dBm-R3
configurable retransmits</td><td>84.1mA-R1
83.9mA-R2
83.6mA-R3</td></tr>
<tr><td>ITU-T
low power
PHY
Z-Wave</td><td>9.6kbps-R1
40kbps-R2
100kbps-R3
Rate Adaptation Allowed</td><td>≤-102.7dBm- R1
-99dBm-R2
-93dBm-R3
configurable retransmits</td><td>42.1mA-R1
42.1mA-R2
42.1mA-R3</td></tr>
</table>

Evaluation and Testbed: CharIoT gateways use inexpensive RTL2832U SDR dongles plugged into Raspberry Pis with an Ethernet backhaul (see Fig 8c). Each RTL-SDR is configured at 868 MHz center frequency and receives samples at a bandwidth of 1 MHz (Sub-1GHz ISM band is very

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

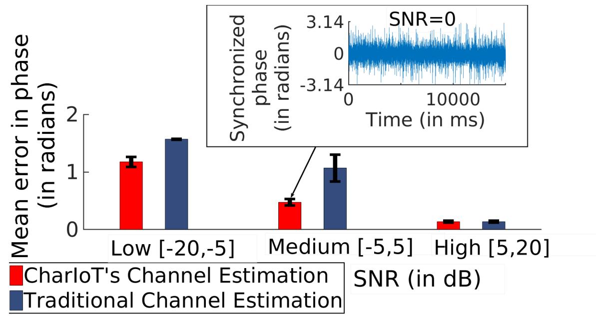


(a) TestBed 1- Top view of the two floor building. Red dots denote receiver positions on top floor. Same positioning is repeated on the floor below.



(b) TestBed 2- T shaped indoor testbed. Red dots denote distribution of 10 gateways in the space.

(c) CharIoT's gateway prototype- RTL-SDR+RPi.



(d) Microbenchmark on synchronization.

Figure 8: Implementation of CharIoT-Testbeds, gateway prototype and the microbenchmark on synchronization.

limited). CharIoT emulates distributed MIMO behaviour by synchronizing ten such gateways (see Sec. 5.1). We deploy CharIoT on two testbeds – a 2,680 $m^2\times$ 10 m two-floor apartment building, and a 1,830 m^2 T-shaped indoor area (as shown in Figs. 8a and 8b respectively). The transmitters are distributed randomly in and around the receiver testbed area and are in complete asynchronous mode of operation; all supporting the EU default standard for transmission. The duty cycle of the devices is adjusted to capture all possible scenarios, including collisions within and across technologies. We use SemTech SX1276 chipset for LoRa, TI CC1310 for XBee, UZB static controller for Z-Wave and ATA 8520-EK3 to run SIGFOX. LP-WAN transmitters corresponding to LoRa and SIGFOX are configured to transmit in powers comparable to the short range technologies. SIGFOX transmitters are set to test mode to transmit without hopping and Z-Wave static controller collisions are shifted to 868MHz center frequency (default is 868.4MHz). This difference in frequency is a vendor-specific property of the UZB dongle from Sigma designs and for proof of concept we study the shifted version of Z-Wave for collision recovery. Similar is the case with SIGFOX where frequency hopping is disabled in most regional configurations, including in India where we implemented our test-beds. The experiments, performed during working hours, ensure maximum effects of multipath, thereby emulating a real world smart environment.

Baseline: We compare CharIoT's collision resolution against traditional receivers that implement Successive Interference Cancellation – the state-of-the-art technique to decouple collisions in IoT gateway platforms [9], [24], [54].

Monitoring the effect of collisions: Previous studies [3], [32], [43], [55], [56], [57] discuss the effect of collision on different low power technologies. Since we are considering low-power devices which rely on 'wake up and transmit' model [43], we monitor the effect of collisions in terms of two important performance metrics that stem from resultant re-transmissions– the impact on throughput, and the effect on battery life-time of each radio transmitter.

Battery Models: Using well-known battery models from past literature [30], [43], we map the SNR gains to the improvement in the RSSI values and compare them against the technology-specific data rates and energy values for each technology– LoRa [38], [43], Z-Wave [39], [53], XBee [40], [51] and SIGFOX [48], [58] (refer Table 5).

8 Discussion– CharIoT's Design Space

Collisions between short and long range technologies: High proliferation of IoT within the limited Sub-1GHz ISM spectrum inevitably leads to collision within and across technologies – short as well as long range ones. Though previous studies [32] envision transmissions like convergecast leading to collision across short range technologies, collision aggravates with long range technologies in the picture. Long range technologies have transmission range in the order of kilometers making them spatially co-located and accessing the same channel as the short range technologies due to its large transmit radius. In addition, long range technologies transmit extremely long packets with larger spread over time [3], implying that their channel occupancy lasts for tens of seconds [59]. This means, for collisions to be avoided, channel occupancy of long-range and short-range technologies should not overlap, which is extremely unlikely considering the vision of tens of thousands of co-located IoT devices, especially in smart city scenarios [60].

Non-CSMA for low power transmitters: Though technologies like XBee support an optional slotted CSMA/CA implementation to avoid collisions, recent literature [3], [30], [33], [43], [57] suggests that medium access control by itself can be quite battery draining since it requires the radio front-end to be switched on for longer duration. OpenChirp [43] shows a detailed evaluation of the energy drainage caused by radio front end while channel sensing in the case of LoRa. These studies provide collision recovery from 'wake up and transmit' mode as a better option when compared to the CSMA-based collision avoidance for the energy-constrained Sub-1GHz radio technologies.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

However, we stress that the non-CSMA nature of transmission is specific to the low-power transmitter standard [3], and CharIoT simply aids such a system in resolving collisions that would have inevitably occurred.

How does CharIoT deal with corner-cases? Though CharIoT might have its failure modes like every system, we discuss some corner-cases where CharIoT performs significantly better than SIC.

(a) Dealing with diverse signals with partial overlaps in frequency: Partial signal overlaps are generally handled exclusively by the software filters. These filters start cancelling out signals one by one, starting from the ones with minimal spread, checking for decodability of the remaining ones till all signals are decoded. The decoded signals from multiple receptions are coherently combined to boost its SNR and improve the decoding accuracy.

(b) Dealing with a combination of signals with complete frequency overlaps: This requires a combination of software filters and MIMO multiplexing. Using software filters, CharIoT tries to minimize the number of diverse signals overlapping in frequency domain. These filters reach their limit when the overlapping signals have minimal difference in their radio modulation. Once this limit is achieved, it reverts to either offset based-filters (for LP-WANs) or Zero-forcing (for short-range technologies) to decouple same technology collisions.

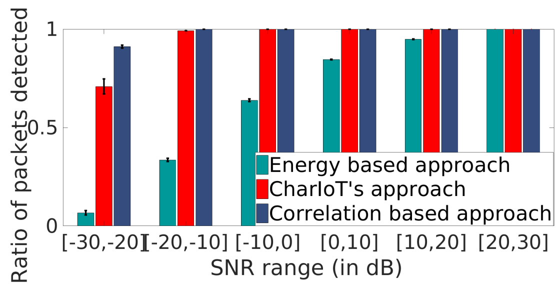
(c) Decoupling transmissions higher than the number of receivers: Combining the benefits offered by software filters on the top of state of the art techniques give CharIoT a significant advantage in handling collisions from a large number of transmissions. CharIoT's software filters in particular have the potential to decode multiple transmitters even from a single receive antenna, given that low-power clients are often significantly Shannon sub-optimal. While, these filters enable decoupling collisions across diverse transmissions, offset-based filtering [3] and separation based on power differences [52] are already tried and tested out techniques for separating same technology collisions across single antenna receptions. With the added advantage offered by the near-far effects of geographically separated antennas, CharIoT's architecture is hence extremely efficient in decoding a large number of transmissions, much higher than the number of receiver antennas.

9 Experimental Results

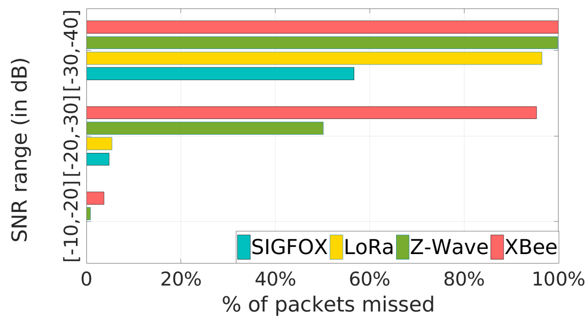
9.1 Microbenchmark: Channel Estimation

Setup: To show the effectiveness of our channel estimation algorithm, we compare the mean error incurred on channel estimation with and without our algorithm, for all the four technologies, at different SNR ranges.

Results: Fig. 8d shows that the error in channel calculation remains below 0.5 radians using our algorithm at medium SNRs, where the signal is close to the noise floor. Even at 0dB, the error in estimation can be seen as a band concentrating at zero with a noise band surrounding it. This implies a mean error reduction by more than half on what is incurred by the traditional channel estimation algorithms, even with very lengthy preamble patterns. At very low SNRs (-20dB), traditional channel estimation almost always fails to capture the effect of offsets because of which the channel error will fluctuate rapidly across 0 to π giving a mean error of $\pi/2$ (error bar $\sim$ 0 here). Our algorithm remains noise resilient with error$\leq$1 radian until -15 dB after which its performance gradually drops and touches $\pi/2$ for SNRs below -25dB.



(a) Detection comparison considering 1000 packets per technology



(b) Noise resilience/technology in universal preamble

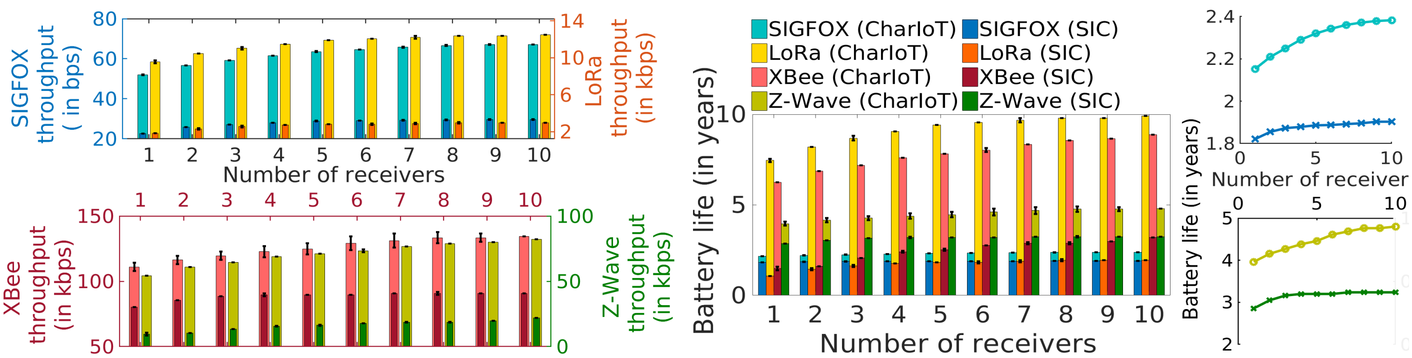
Figure 9: Packet detection.

9.2 Detection at the Gateway

Setup: CharIoT gateway is tuned to continuously listen to the 868MHz channel, sampling signals at 1MHz bandwidth. Around 16 transmitters, four from each technology- LoRa, SIGFOX, XBee and Z-Wave, are configured to transmit in low power with no synchronization and their duty cycles are adjusted to facilitate multiple instances of collisions. We collect the RTL-SDR traces to observe the performance of our detection scheme under SNRs varying from -30 to +30dB. Traces maintain the same number of packets from each technology to ensure consistency. The universal preamble is created by adding up the representative preambles from all technologies, with zeros padded at the end to compensate for unequal preamble lengths. We consider two cases – first case considering Z-Wave centered at 868.4 MHz while the others transmit centering around 868MHz and measuring the detection; in the second case, the same operation is done with all technologies including Z-Wave centered at 868 MHz. The performance of our detection scheme is compared with the existing energy detection scheme as well as the optimal correlation scheme with individual preambles that is computationally-intensive. Further, we lower the SNRs by up to -40dB to capture the resilience of individual technologies within the universal preamble at low SNRs.

Results: Fig. 9a, shows the performance comparison of each detection technique at different SNR regimes. The detection accuracy of CharIoTś scheme remains high even for SNR values as low as -10dB in contrast to the energy detection schemes proposed in the past literature [4]. It is to be noted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing



(a) Average Throughput

(b) Average Battery Life

Figure 10: Diversity gains after decoupling collisions using software filters for 4 transmitters, one per technology– LoRa, SIGFOX, XBee & Z-Wave. Depicts gains per technology averaged over 1000 iterations with error bar.

that the universal detection scheme remains resilient even to intra and cross collisions which are captured in this scenario. This is because of the near-orthogonal nature of the constituent preambles as mentioned in Sec. 4. In contrast, the energy based detection schemes proposed in the existing multi-technology literature [4] scale poorly with increasing noise levels and collisions. At SNR below 0dB, the performance of energy based detection shows a steep drop from approximately 85% to less than 0.05%. Our system on the contrary maintains nearly 70% detection accuracy even for SNR regimes up to -30dB. The slightly higher susceptibility of universal preamble to white noise compared to the individual preamble contributes to the small drop in detection at noise levels lower than -10dB. This is primarily contributed by the failure in detection of the second (lower SNR) packet in certain instances of collisions, even when at least one packet in a collision is very likely to be detected.

To get a better notion on the performance drops of universal detection at very low SNRs, we perform a detailed analysis on the resilience of each constituent technologies within the universal preamble. The analysis is performed at SNR regimes below -10dB going up to -40dB to stress the system better. As can be seen from Fig. 9b, as the SNR goes down, the percentage of packets missed drops steeply for technologies that belong to the short range category, while this drop is more gradual for LP-WAN technologies. The reason for such a behavior is simple. Since correlation is calculated as the sum of products, the performance of the technique is directly proportional to the length of the sequence we correlate with. Hence short range technologies having a shorter preamble have a lesser chance of maintaining the peaks vs. the LP-WAN technologies, which preserve peaks for lowering SNRs. Therefore, the trend observed here is directly attributed to the order of increasing preamble lengths – XBee with the shortest and SIGFOX with the longest preamble. Note that for Z-Wave, the detection is more accurate at 868.4MHz, performing exactly the same as the correlation based detection for SNRs as low as -30dB. This is because unlike at 868MHz, Z-Wave when centered at 868.4MHz is completely separable from the rest of the technologies in the frequency domain.

We also monitored the real-time traffic between the CharIoT gateway and the cloud after performing the local detection. After the detection, we observed a maximum of 5.712 Mb/s of samples being transmitted to the cloud over the Ethernet [4]. This is ideal for a normal Cat5 home Ethernet cable which supports data streaming of the order 10-100 Mb/s. Traffic from cloud to gateway is insignificant since they are typically beacon-sized acknowledgements.

9.3 Cross-technology Collision Resolution

Setup: Four transmitters, one from each technology – LoRa, SIGFOX, XBee and Z-Wave, are configured to transmit at 868MHz center frequency (EU standard ISM band). The duty cycle of each transmitter is engineered to ensure multiple collision instances. Ten synchronized CharIoT gateways listen to the channel and stream received signal collisions to the cloud where CharIoT processing is performed. CharIoT then runs step 9 of Algorithm 2 to process cross-technology collisions. The measurements were collected across weeks on the two indoor testbeds as depicted in the Fig. 8a and 8b.

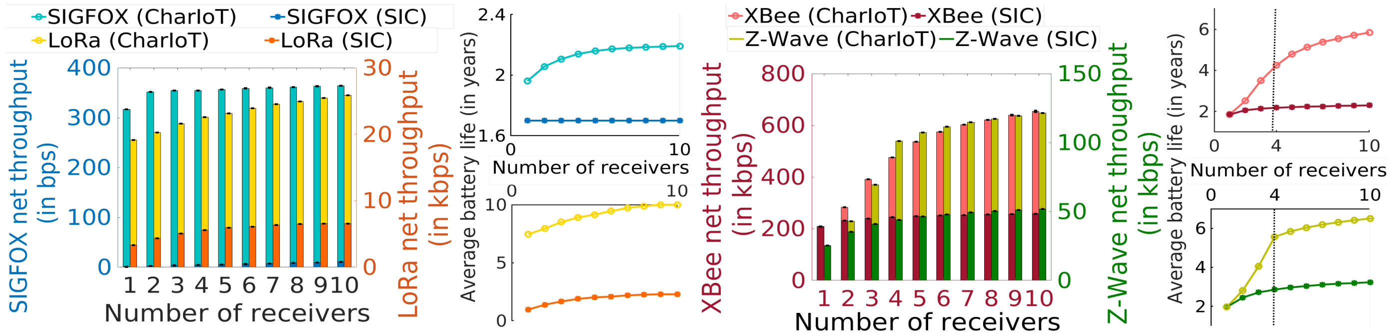
Results: Fig. 10 depicts the throughput and battery life gains corresponding to each technology after processing.

SIGFOX gains: SIGFOX's inherent re-transmission limit of 3 transmits per packet (1 transmit and 2 re-transmit) preserves its battery life to a large extent offering around 1-3 years of battery life for an AA Lithium battery with 3000mAh (1 packet/hour)(first bar in Fig. 10b). But collisions create a huge drop in its throughput –to about one-fifth of its maximum limit, even with SIC. These collisions can be mitigated to a considerable extent by CharIoT's software filters. SIGFOX being an Ultra-Narrow Band technology facilitates easy separation of BFSK signals using KILL-FREQUENCY filter and LoRa signals using KILL-DSS. With an SNR boost of 10-20dB offered, software filters alone can hence offer an average of 50bps for every transmitted packet in SIGFOX. With an extra boost of 3-4dB provided by coherent combining SIGFOX can achieve roughly four-fifth of its maximum possible throughput with up to 2.5 years of battery life. Note that despite 3-4dB of SNR gains provided by coherent combining, the diversity gains in SIGFOX are not considerable (3-4 months and 20bps extra due to strict limits on re-transmission.

LoRa gains: LoRa transmissions configured at 500kHz experience throughput of upto 21.8 kbps in an uncollided scenario. Under an event of collision with the other

4. Note that is an upper bound since our transmitter duty-cycles were configured apriori to collide more often and hence have more packets detected than the real-world scenario.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing



(a) Multiplexing gains for LoRa and SIGFOX

(b) Multiplexing gains for XBee and Z-Wave

Figure 11: Decoupling same technology collisions. Gains averaged over 1000 iterations with error bar are shown.

technologies – SIGFOX, XBee and Z-Wave, LoRa throughput encounters a drastic decrease, offering a maximum of ∼2.5kbps upon implementing SIC. This is only a tenth of the maximum rate offered by LoRa, leading to a steep decline in battery life. The second bar in Fig. 10b depicts the battery life in LoRa while using a 3000mAh AA battery. CharIoT can provide a sharp increase up to 5 × the current battery life and throughput (SNR increase ∼13 dB) using software filters. To retrieve LoRa symbols, CharIoT first applies KILL-FREQUENCY that negates other signals and later interpolates the filtered portions to retrieve the LoRa signal. Higher SNR thus obtained can be further boosted by coherent combination of the filtered portions obtained across other received antennas. This offers around 5-6dB of SNR increase, translating to an extra 3000bps in terms of throughput and 2.5 years of battery life.

XBee gains: XBee transmits short intermittent packets and offers comparatively better collision resilience due its smaller packet size. Hence cross-technology collisions generally cause a binary effect in XBee – either the packet is fully recovered or will be fully erroneous. The KILL-DSS filter is most effective here where the XBee packet can be removed from collision leading to around 7-10dB of an SNR boost on an average. KILL-FREQUENCY filters can disentangle collisions across other BFSK transmissions like Z-Wave but their effectiveness is comparatively smaller in the XBee context due to its short symbol size. Finally, diversity offers an additional 4-5dB gain doubling the gains provided by software filters, offering a total of 4.5 years of battery life (third bar in Fig. 10b) and around 125kbps throughput (see Fig. 10a) on average.

Z-Wave gains: Z-Wave offers longer range vs. XBee with longer symbol duration and lower bit rates. Hence these packets are more prone to collision with SIC failing to give considerable gains. Software filters in turn can offer a tremendous improvement of 12dB on an average. KILL-DSS can almost completely remove LoRa based signal components from Z-Wave while KILL-FREQUENCY is effective in disentangling SIGFOX and XBee collisions. Along with the diversity gains provided by multiple receive antennas, Z-Wave transmitters can have a battery life of upto 5 years (fourth bar in Fig. 10b) offering throughput up to 85 kbps (Fig. 10a) – over four-fifth of their maximum achievable throughput.

9.4 Separating Same Technology Collisions

Setup: CharIoT's multiplexing gains can be measured by keeping the same setup as the previous study, but considering only collisions within instead of across technologies. Hence, we deploy four transmitters belonging to one of the four technologies – LoRa, SIGFOX, XBee and Z-Wave, and we repeat the experiment for all the four over several iterations under different multi-path conditions to study multiple collision instances. The datasets retrieved from 10 synchronized CharIoT gateways capture –1) multiplexing gains of four transmissions from data collected at four gateways, 2) diversity gains after coherent combination of transmissions received from more than four antennas.

Results: Figures 11a and 11b depict the gains of our system while resolving intra-technology collisions. For LP-WAN technologies, offset based filtering can offer up to 10-15dB SNR gain in both SIGFOX and LoRa, which as seen from Fig. 11a offers battery and throughput performance comparable to that of software filters. The resultant gains are further improved by diversity combining, offering an average of 12-14kbps per device for LoRa and close to 80bps per device for SIGFOX. CharIoT can achieve close to the promised shelf life for both technologies for LoRa as well as SIGFOX–almost 10 years on 3000 mAh AA battery for LoRa and nearly 2.2 years (maximum being 2.4 years) for SIGFOX, respectively. Due to shorter symbol size, efficient separation of collision across XBee and Z-Wave rely on zero forcing to decouple its collisions. Hence Fig. 11b shows linear multiplexing gains up until four receivers, after which logarithmic diversity gains are seen for both the technologies. We observe 80kbps of net throughput gain on each antenna addition in the case of XBee and 40kbps for Zwave. This allows transmitters to maintain the R2 bit rate (see Table 5) configuration that allows considerable battery gains for the devices – 2 years for XBee and 4 years for Zwave.

9.5 Testing the Scaling Limits of CharIoT

Setup: Next, we stress test CharIoT by intentionally engineering both inter and intra-technology collisions at scale. We measure the decodability of 16 transmissions–4 from each technology, at each gateway one by one, across gateway pairs, and so on up to 10 gateways.

Results: Fig. 12 shows that our system achieves significant battery life (up to 293.96% gain – i.e. an additional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3029218, IEEE Transactions on Mobile Computing

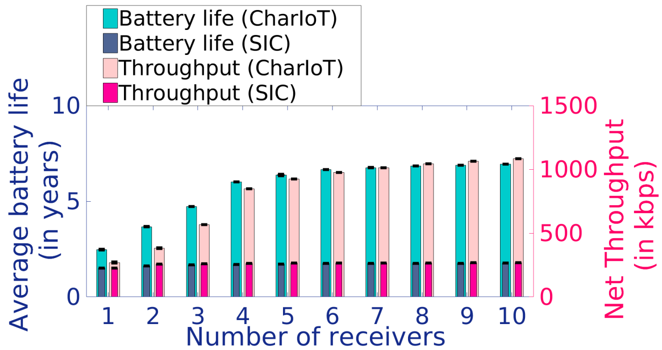


Figure 12: Shows CharIoT's battery life and throughput gains averaged over 1000 iterations. Decouples collisions from 16 clients – 4 from each technology.

3.5-5 years on average) and throughput gains (4× for 10 gateways) across number of single-antenna gateways averaged across all technologies. We make several observations. First, it appears that the gains in net throughput across transmitters is somewhat modest relative to the number of competing clients (sixteen). This is because XBee/Z-Wave have a significantly higher throughput than LoRa/SIGFOX (200/100 kbps vs. 21.2/0.1 kbps) causing every additional LoRa/SIGFOX client to contribute relatively small values to net throughput. Second, we note that broken down to individual technologies the net gains for LoRa, SIGFOX, Z-Wave and XBee over 10 gateways remain significant – respectively 7.1, 7.9, 6.2 and 3.3 × for throughput and 8.3, 1.3, 4.1 and 3.2 × for battery life. The differences in throughput/battery gains across technologies stem mainly from the nuances in their modulation and battery models as explained earlier. Finally, note that the gains saturate at about 4 single-antenna gateways with additional gateways providing minor additional gain (mainly diversity benefits). This is because the gains from software filters coupled with distributed MIMO disentangle all required transmissions with as little as four gateway antennas. This further validates our ability to decouple many more concurrent transmissions than total number of receive antennas.

9.6 Delay analysis – CharIoT versus Re-transmission Timeout

Setup: To show the effectiveness of CharIoT's collision resolution in preventing further retransmits, we monitor the time required for each processing–cross technology as well as intra-technology collision resolution at the cloud. Our cloud constitutes an Intel® Xeon(R) CPU E3-1226 v3 operating at 3.30GHz having 4 cores. This processing delay at the cloud is compared against the pre-set re-transmission timeout for each radio technology. Note that though the re-transmission timeout can be configured typically in a transmitter, we have given the values that suite the default technology specific parameter.

Results: Table 6 provides the maximum delay incurred for resolving collisions using software filters. Taking into account the delays for hardware-offset based filtering and zero forcing, the maximum processing time incurred by CharIoT (processing + round trip delay) is tabulated in Table 7. These values are further compared against the standard-compliant re-transmission timeout for each radio technology.

Table 6: Delay incurred from software filters in CharIoT.

Software Filters	Maximum delay in CharIoT
kill-FREQUENCY	0.38042825 s
kill-DSS	0.316633 s

Table 7: Delay evaluation – CharIoT versus time requirement of ACK for each technology.

Technology	Maximum delay in CharIoT	Re-transmission timeout
XBee	1.11938 s	1.6 s [61]
Z-Wave	0.93640 s	1.5 s [62]
LoRa	0.97031 s	1-3 s [63]
SIGFOX	2.308374 s	20 s [48]

10 Conclusion and Future Work

This paper presented CharIoT, the first distributed MIMO solution that mitigates collisions across low-power IoT radio technologies. CharIoT enables low-power IoT gateways to ship I/Q samples corresponding to collisions to the cloud. At the cloud, CharIoT employed novel software filters that separate transmissions received across low-power IoT technologies, based on properties unique to their modulation. We implemented CharIoT on inexpensive RTL-SDR gateways and showed simultaneous decoding of collisions across four popular low-power IoT technologies in large indoor testbeds.

We make some insights for building upon CharIoT in the future:

1. CharIoT for future technologies - CharIoT's synchronization and channel estimation algorithm have been designed to be generalizable to accommodate future low-power technologies like NB-IoT and WiFi HaLo. This is because CharIoT relies on some of the fundamental features that are common across low power technologies – Shannon sub-optimal data rates, long packets with extremely long and redundant preambles, signals of extremely low power comparable to that of noise floors and imperfect devices with offsets on time and frequencies. Finally, CharIoT also opens up the scope to identify similar 'kill' filters tailor-made for other modulation schemes.

2. CharIoT for long range with downlink support - Though CharIoT takes into consideration the LP-WAN technologies as well, the current implementation of CharIoT is restricted to indoor spaces. Implementing CharIoT for long range communication can lead to new challenges stemming from mitigating cross-technology collisions in wide-area settings. Also, CharIoT being the initial system prototype, does not handle the details involved for providing downlink transmission support. A full-fledged implementation of CharIoT needs to develop methodologies to enable a two-way transmission support, which we propose as a future work.

3. Universal preamble for non-IoT technologies - Low power technologies are inherently simple with less complex modulation schemes. Hence there is a very limited set of combination for preambles for each technology in this domain. This is why preambles in this context add up to make the concept of universal preamble work. For non-IoT technologies, many wireless systems allow complex designs and even more complex modulations. Based on the design and the regulatory concerns, universal preamble may or may not work in such a case and can be considered for a future study.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This research work was supported by Department of Science and Technology (DST), New Delhi, India, and National Science Foundation (NSF), USA.

REFERENCES

[1] "Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025," 2016. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] M. Park, "IEEE 802.11ah: Sub-1GHz License-Exempt Operation for the Internet of Things," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 145–151, 2015.

[3] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*. ACM, 2017, pp. 309–321.

[4] A. Hithnawi, H. Shafagh, and S. Duquennoy, "TIIM: technology-independent interference mitigation for low-power wireless networks," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. ACM, 2015, pp. 1–12.

[5] K. Chebrolu and A. Dhekne, "Esense: Energy sensing-based cross-technology communication," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2303–2316, 2012.

[6] S. M. Kim and T. He, "FreeBee: Cross-technology Communication via Free Side-channel," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2015, pp. 317–330.

[7] Z. Yin, W. Jiang, S. M. Kim, and T. He, "C-morse: Cross-technology communication with transparent morse coding," in *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE, 2017, pp. 1–9.

[8] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2017, pp. 2–14.

[9] Multitech, *Data Sheet: MultiConnect Conduit: Programmable Gateways (Wi-Fi/Bluetooth/GNSS)*, 2019.

[10] E. Hamed, H. Rahul, M. A. Abdelghany, and D. Katabi, "Real-time distributed mimo systems," in *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016, pp. 412–425.

[11] Y. Yubo, Y. Panlong, L. Xiangyang, T. Yue, Z. Lan, and Y. Lizhao, "Zimo: Building cross-technology MIMO to harmonize Zigbee smog with WiFi flash without intervention," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking*, 2013.

[12] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer MIMO to today's wireless LANs," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2013, pp. 387–398.

[13] W. Jiang, Z. Yin, S. M. Kim, and T. He, "Transparent cross-technology communication over data traffic," in *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE, 2017, pp. 1–9.

[14] W. Wang, S. He, L. Sun, T. Jiang, and Q. Zhang, "Cross-Technology Communications for Heterogeneous IoT Devices through Artificial Doppler Shifts," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 796–806, 2018.

[15] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent Cross-Technology Communication Among Heterogeneous IoT Devices," *IEEE/ACM Transactions on Networking*, pp. 932–947, 2019.

[16] P. Yang, Y. Yan, X.-Y. Li, Y. Zhang, Y. Tao, and L. You, "Taming cross-technology interference for Wi-Fi and ZigBee coexistence networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 1009–1021, 2016.

[17] J. Elias, S. Paris, and M. Krunz, "Cross-technology interference mitigation in body area networks: An optimization approach," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4144–4157, 2015.

[18] S. M. Kim, S. Wang, and T. He, "IoT networking: From coexistence to collaboration," in *Proceedings of the International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2016, pp. 212–217.

[19] Y. Zhang and Q. Li, "Howies: A holistic approach to ZigBee assisted Wi-Fi energy savings in mobile devices," in *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE, 2013, pp. 1366–1374.

[20] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the RF smog: making 802.11n robust to cross-technology interference," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2011, pp. 170–181.

[21] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving WiFi interference in low power Zigbee networks," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2010, pp. 309–322.

[22] S. S. Hong and S. R. Katti, "DOF: a local wireless information plane," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2011, pp. 230–241.

[23] I. Tinnirello, D. Croce, N. Galioto, D. Garlisi, and F. Giuliano, "Cross-technology Wi-Fi/ZigBee communications: Dealing with channel insertions and deletions," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2300–2303, 2016.

[24] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2008.

[25] R. Calvo-Palomino, H. Cordobés, F. Ricciato, D. Giustiniano, and V. Lenders, "Collaborative Wideband Signal Decoding Using Non-coherent Receivers," in *Proceedings of the ACM/ IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. ACM, 2019, pp. 37–48.

[26] A. Del Coso, U. Spagnolini, and C. Ibars, "Cooperative distributed MIMO channels in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 402–414, 2007.

[27] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker, "SAM: enabling practical spatial multiple access in wireless LAN," in *Proceedings of the Annual International Conference on Mobile computing and Networking (MobiCom)*. ACM, 2009, pp. 49–60.

[28] H. Rahul, S. Kumar, and D. Katabi, "Jmb: Scaling wireless capacity with user demands," *Communications of the ACM*, vol. 57, no. 7, pp. 97–106, 2014.

[29] V. Yenamandra and K. Srinivasan, "Vidyut: exploiting power line infrastructure for enterprise wireless networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communications (SIGCOMM)*. ACM, 2014, pp. 595–606.

[30] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 60–71.

[31] M. Hessar, A. Najafi, and S. Gollakota, "NetScatter: Enabling Large-Scale Backscatter Networks," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2019, pp. 271–284.

[32] L. Kong and X. Liu, "mZig: Enabling multi-packet reception in ZigBee," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking*, 2015.

[33] R. Narayanan and S. Kumar, "Revisiting software defined radios in the iot era," in *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)*. ACM, 2018, pp. 43–49.

[34] O. Steila, "The rtl2832+r820t rf generator hack 0.9," Available at http://www.steila.com/blog/index.php?controller=post&action=view&id_post=8 (2015/06/18).

[35] B. Technologies, "10 popular software defined radios (sdrs) of 2019," Available at https://blog.bliley.com/10-popular-sdrs-software-defined-radios-2018 (2018/12/31).

[36] S. Nagaraj, S. Khan, C. Schlegel, and M. Burnashev, "On preamble detection in packet-based wireless networks," in *Proceedings of the International Symposium on Spread Spectrum Techniques and Applications (ISSSTA)*. IEEE, 2006, pp. 476–480.

[37] R. P. Forums, "Cpu execution speed," Available at https://www.raspberrypi.org/forums/viewtopic.php?t=237225 (2019/04/01).

[38] "LoRaWAN 1.1 Specification," *LoRa Alliance*, 2017.

[39] "Short range narrow-band digital radiocommunication transceivers -PHY, MAC, SAR and LLC layer specifications," *ITU-T G.9959*, 2015.
[40] "IEEE standard for low-rate wireless networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.
[41] "Sigfox Specification," *Sigfox Alliance*, 2017.
[42] R. Varatharajan, G. Manogaran, M. Priyan, and R. Sundarasekar, "Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm," *Cluster Computing*, pp. 1–10, 2017.
[43] A. Dongare, A. Luong, A. Balanuta, C. Hesling, K. Bhatia, B. Iannucci, S. Kumar, and A. Rowe, "The openchirp low-power wide-area network and ecosystem: Demo abstract," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE Press, 2018, pp. 138–139.
[44] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov, "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN," in *Proceedings of the Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–7.
[45] M. Lauridsen, B. Vejlgaard, I. Z. Kovacs, H. Nguyen, and P. Mogensen, "Interference measurements in the European 868 MHz ISM band with focus on LoRa and SigFox," in *Proceedings of the Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.
[46] G. Ferré and E. Simon, "An introduction to Sigfox and LoRa PHY and MAC layers," 2018, hal-01774080,v1.
[47] G. G. Ribeiro, L. F. de Lima, L. Oliveira, J. J. Rodrigues, C. N. Marins, and G. A. Marcondes, "An outdoor localization system based on sigfox," in *Proceedings of the Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–5.
[48] *Sigfox Device FH mode White Paper*, 2019.
[49] C.-S. Sum, F. Kojima, and H. Harada, "Coexistence of homogeneous and heterogeneous systems for IEEE 802.15. 4g smart utility networks," in *Proceedings of the International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2011, pp. 510–520.
[50] C. Orfanidis, L. M. Feeney, M. Jacobsson, and P. Gunningberg, "Investigating interference between LoRa and IEEE 802.15. 4g networks," in *Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2017, pp. 1–8.
[51] J. Muñoz, T. Chang, X. Vilajosana, and T. Watteyne, "Evaluation of IEEE802. 15.4 g for Environmental Observations," *Sensors*, vol. 18, no. 10, pp. 3468 (1–37, 2018.
[52] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, p. 16, 2010.
[53] C. Paetz, *Z-Wave Essentials*. Christian Paetz, 2017.
[54] S. P. Weber, J. G. Andrews, X. Yang, and G. De Veciana, "Transmission capacity of wireless ad hoc networks with successive interference cancellation," *IEEE Transactions on Information Theory*, vol. 53, no. 8, pp. 2799–2814, 2007.
[55] B. Moeller and J. J. Bohn, "Resolving communication collisions in a heterogeneous network," 2016, US Patent 9,479,450.
[56] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SIGFOX communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, 2019.
[57] R. Abbas, A. Al-Sherbaz, A. Bennecer, and P. Picton, "Collision Evaluation in Low Power Wide Area Networks," in *Proceedings of the IEEE International Conference on Scalable Computing and Communication*, 2019, pp. 1–8.
[58] S. D. matter, *Oyster Sigfox Integration*, 2019.
[59] Arjan, "The Things Network: Fair Access policy guidelines," 2019. [Online]. Available: https://www.thethingsnetwork.org/forum/fair-access-policy-guidelines.
[60] E. Buckland, M. Ranken, M. Arnott, and P. Owen, "IoT Global Forecast & Analysis 2015–25," *Machina Research: London, UK*, 2016.
[61] Digi, "Digi XBee® 3 802.15.4," 2018.
[62] Niels Thybo Johansen, "Z-Wave Serial API Host Application Programming Guide," 2018.
[63] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, 2017.

Revathy Narayanan is currently a PhD candidate in the Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai. She was a recipient of Fulbright-Nehru Doctoral Research Fellowship which enabled her to work as a visiting researcher in WiTech Lab, Carnegie Mellon University. Her research aims at providing a holistic solution that facilitates cross-technology connectivity and interactions across low-power radio technologies in the Internet of Things domain.

Swarun Kumar is an Assistant Professor at Carnegie Mellon University, where he leads the Emerging Wireless Technologies (WiTech) lab. His research builds next-generation wireless network protocols and services. Swarun is a recipient of the NSF CAREER award and Google Faculty Research award. He received the George Sprowls Award for best Ph.D thesis in Computer Science at Massachusetts Institute of Technology and the President of India gold medal for his B.Tech at Indian Institute of Technology Madras.

C. Siva Ram Murthy (F'12) received the Ph.D. degree in computer science from the Indian Institute of Science, Bengaluru, India, in 1988. Since 1988, he has been with the Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, where he is currently the Richard Karp Institute Chair Professor of computer science and engineering. He is an elected Fellow of the World Academy of Sciences for the advancement of science in developing countries (TWAS, Italy), Indian National Science Academy (INSA), and Indian National Academy of Engineering (INAE).