Revisiting Software Defined Radios in the IoT Era

Revathy Narayanan[1], Swarun Kumar[2]
Indian Institute of Technology Madras [1], Carnegie Mellon University [2]

ABSTRACT

Several years ago, software radios were seen as the future of commercial wireless infrastructure, given that they could flexibly decode any wireless technology with a simple software update. Yet, despite their numerous advantages, the industry tilted in favor of dedicated chips for wireless infrastructure sacrificing generality for performance. Today, the advent of the Internet of Things (IoT) and the resulting fragmentation of wireless technologies has led to a new billion-dollar industry: multi-technology gateways that support many radio technologies. Predictably, commercial gateways do this through multiple radio chips each decoding dedicated wireless technologies.

This paper argues that software radios deserve a revisit in the IoT era. Specifically, IoT wireless technologies have significantly lax requirements in terms of throughput, latency and cost, meaning that they can be implemented on cheap software radio gateways that offer extensibility through software updates. We propose GalioT, a cloud-assisted IoT gateway that demonstrates the numerous benefits of software radios beyond programmability. We focus, in particular, on the rampant cross-technology interference in IoT networks since devices are low-power and simply "wake up and transmit". We use simple software radio base stations that detect signals including collisions of multiple radio technologies and ship them to the cloud. At the cloud, we exploit PHY-layer differences across radio technologies to concurrently decode them. A prototype implementation on an inexpensive RTL-SDR and Raspberry-Pi based gateway shows promising accuracy in concurrent decoding of XBee, Z-Wave and LoRa.

Figure 1: GalioT: A Multi-technology IoT framework

1 INTRODUCTION

In their early years, software defined radios (SDRs) were seen as the future of commercial wireless infrastructure. Programmable radios can simultaneously decode multiple wireless technologies – WiFi, LTE, bluetooth – allowing for software updates to support new radio technologies. However, software radio vision received a reality check in the ensuing years and struggled to cope with high throughput and stringent latency requirements of traditional wireless protocols at competitive costs. Thus, the industry moved towards dedicated chips, sacrificing generality for performance. Today, in the Internet of Things (IoT) era, wireless technologies are highly fragmented, with new devices having heterogeneous radio support even within small geographic boundaries. This creates a new billion-dollar industry: multi-technology gateways [20], which support many radio technologies through multiple dedicated radio chips.

In this paper, we argue that software radios deserve a revisit in the IoT context. First, the performance vs. cost tradeoff no longer holds because the vast majority of IoT devices are simple and low-power [1, 11], meaning that they operate at low-throughput and lax latency requirements [23, 24] allowing for inexpensive software radios; additonally bringing in the concept of universality across an array of radio technologies. Second and more importantly, software radios can address an important challenge faced by IoT gateways today: cross-technology interference, when signals from multiple client radios of different technologies collide. Traditional wireless technologies mitigate these collisions to some degree with wireless protocols that learn to sense and avoid collisions. However, in the IoT era, a majority of wireless technologies target low-power devices which often simply "wake up and transmit" and therefore are vastly more susceptible to collisions. These collisions are handled using retransmissions, resulting in extensive battery drain [11].

The rest of this paper presents GalioT, a software radio architecture for low-power Internet-of-Things. GalioT relies on simultaneous and concurrent decoding of signals across multiple radio technologies, thereby enabling an energy sustainable multi-technology environment. Our key innovation is developing unique software filters based on the modulation of each

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

radio technology to help separate and decode collided radio signals of multiple technologies that can normally be decoded by no IoT gateway today. We show how GalioT can decode collided signals across different radio technologies despite complete overlaps in both time and frequency by exploiting their underlying structure and energy distributions in time and frequency; even in high-noise environments. Our architecture is completely programmable, incooperating support for new radio technologies at the gateway and cloud using simple software updates. Prototype implementation of GalioT include a low-cost Raspberry PI and RTL-SDR, decoding three popular IoT technologies: XBee, Z-Wave and LoRa, demonstrating a programmable design that is an order-of-magnitude cheaper compared to today's commercial gateways.

At the gateway, GalioT develops a novel and efficient mechanism to detect signals across radio technologies, even as some may potentially interfere with others. A straw-man approach to do so is to detect energy spikes in the received signal to detect ongoing transmissions regardless of the radio technologies and sending these to the cloud. However, low-power transmissions are often well-below the noise and therefore are easily missed by a simple energy threshold. An alternative is to correlate with the unique preamble of each radio technology, but would add to the computational complexity of low-cost gateways and would scale poorly with every new radio technology. GalioT instead devises a universal preamble that combines unique signatures of both the preamble and data of all radio technologies it intends to decode. We show how our approach scales efficiently with the incorporation of new radio technologies while remaining robust to collisions across technologies.

Received signals at the gateway, including potential collisions, are separated and decoded at the cloud. GalioT's ability to decode multiple radio technologies gives it a natural advantage in dealing with cross-technology interference. For instance, knowledge that one or more technologies in the received signal are encoded in specific codes could help in decoding the remaining technologies by projecting the signal along orthogonal codes. GalioT generalizes this approach across different classes of modulation by developing unique filters that eliminate their interference to optimally recover and decode collided signals.

We implement GalioT on an RTL-SDR platform connected to a Raspberry Pi with an Ethernet backhaul. Our prototype cost is around 60$, much lower than commercial gateways.We consider various optimizations to overcome limitations in bandwidth and computational capabilities through real-time packet detection and compression. Our prototype evaluation considers three popular IoT wireless technologies: XBee, Z-Wave and LoRa. Our key findings include:

- Our universal preamble detects 50.89% more packets compared to energy detection at SNRs below -10dB.
- Our collison decoding algorithm improves throughput by 7.46 times as that provided by successive interference cancellation, mitigating battery-drain.

Finally, we describe open challenges when scaling GalioT such as hardware design, compression as well as opportunities such as multi-technology wireless sensing in Sec. 6.

Contributions: This paper proposes GalioT, a cheap, programmable inter-technology IoT framework that can decode cross-technology collisions. GalioT designs novel mechanisms to use modulation based properties to efficiently detect low-power signals at gateways across radio technologies and ship them to the cloud. We develop a cloud-based decoder to separate interfering signals across radio technologies by exploiting properties of their underlying modulation. A prototype implementation of GalioT, with three popular IoT radio technologies: XBee, Z-Wave and LoRa, reveals promising accuracy in resolving collisions between them.

2 RELATED WORK

Multi-technology IoT gateways: With increasing fragmentation of IoT technologies [9, 18, 22], multi-technology gateways that allow devices of diverse technologies to communicate are an emerging billion dollar industry. Many offer the flexibility of hardware updates by adding and removing Network Interface Card (NIC) modules to keep pace with new IoT technologies, such as the Samsung Smart Wink 2 and MultiTech Conduit [20]. GalioT achieves the same extensibility with software updates as opposed to hardware changes.

SDR based gateways: Early SDRs were proposed as universal gateways operating across technologies. Past work [10, 19, 25, 30] has developed smart home gateways using the expensive USRP radio with GNU radio support to effectively decode the packets in 2.4GHz ISM bands. However, these solutions struggle to cope with the latency requirements of Wi-Fi and incur significant cost (thousands of dollars). In contrast, GalioT focuses on low-power technologies that can be implemented on cheap programmable radios like RTL-SDRs, while remaining scalable to new technologies with the help of the cloud. Further, GalioT provides a comprehensive solution that deals with cross-technology collisions across asynchronous and heterogeneous low-power radios.

Cross-technology interference: Cross-technology interference is a well studied area with a vast variety of literature available on collision avoidance and mitigation techniques [12, 13, 17, 29, 32]. While the past literature focused on making hardware changes to facilitate interference avoidance [26, 31], more recent techniques propose effective software-based solutions [14–16] that still require computation at the edge or modifications to client nodes. Our approach fundamentally differs from these due to several reasons. First, we impose no changes in software or hardware to low-power client nodes. Second, our solution imposes no additional computational complexity with the addition of new radio technologies at the edge. Finally, we focus exclusively on recent low-power IoT standards in the ISM band which necessitate different joint decoding solutions from existing literature.

3 GALIOT OVERVIEW

At a high level, GalioT's objective is to intelligently decouple collisions of transmissions of multiple radios even if they follow different radio technologies. GalioT attempts to separate collisions even if the number of colliding transmissions exceeds the number of receiving antennas at the base station, by exploiting properties of the modulation of each radio technology. This is possible since low-power IoT technologies operate at a few fixed and low data-rates, meaning that they operate at extremely suboptimal data rates relative to the Shannon limit. This means that frequently (though not always, as we discuss in Sec. 5) base stations have the potential to decode collisions of such transmitters. GalioT's architecture favors simplicity at the base station and a high degree of programmability and extensibility at the cloud.

The GalioT Gateway: At the gateway, GalioT uses an inexpensive software radio receiver that listens to radio signals on unlicensed frequency bands. While a naïve receiver would simply upload all received I/Q samples to the cloud, this poses immense strain on backhaul bandwidth, which is often a commodity cable backhaul at most homes. However, detecting collisions from a large number of transmitting radio technologies is challenging at inexpensive software radios. Further, since transmissions across low-power technologies are typically asynchronous, packet detection should not only be resilient to cross technology interference but also to extremely high noise levels.

GalioT develops a single, unique universal preamble that captures unique signatures of the preambles and data of all the radio technologies it intends to decode. This approach is scalable across new radio technologies through simple software updates and we show how our system is robust to collisions across technologies, while remaining independent of the number of technologies involved. Collision decoupling is done by shipping these I/Q samples to the cloud and edge computing facilitates faster decoding, particularly in scenarios where there is no collision. Sec. 4 details our approach.

GalioT at the Cloud: Having collected collisions detected from the gateways at the cloud, GalioT simultaneously decodes multiple radio technologies. A strawman approach to do this is successive interference cancellation, where strong signals that are robust to interference are decoded first and subtracted out to recover the weaker signals. In contrast, GalioT operates even when signals of similar power levels collide by exploiting modulation-based properties across technologies. Specifically, GalioT implements unique software filters that use the modulation characteristics of any given technology, canceling its signals thereby improving the chances of decoding the other technologies. We show how this allows for concurrent decoding across technologies received at similar power levels greatly improving packet loss compared to independent processing of radio technologies. We present our approach in Sec. 5.

Finally, we present promising future research directions in exploring the design space of GalioT in Sec. 6.

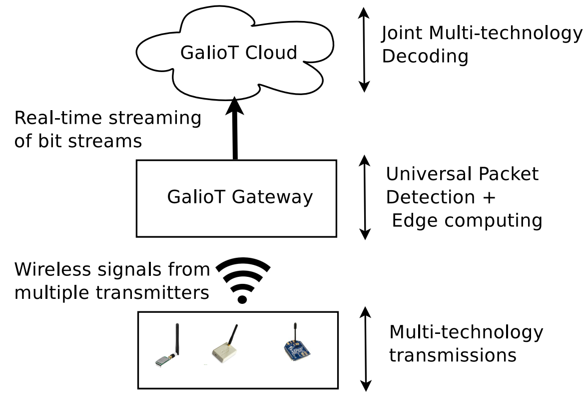


Figure 2: System overview of GalioT

4 UNIVERSAL PACKET DETECTION

This section describes a critical component of the GalioT gateway: the ability to efficiently detect collisions of radio technologies that need to be shipped to the cloud. While a strawman approach to cloud-based PHY-layer decoding would be to stream raw I/Q samples from the RTL-SDR to the cloud, doing so is extremely challenging. Even for narrow band technologies with a bandwidth of the order of hundreds of kHz, the resulting bit streams could be huge (tens of Gbps), that can neither be handled by low-cost platforms or home cable backhauls. GalioT therefore identifies received signals, including collisions, which are then shipped to the cloud, while discarding noise.

While there are several techniques for packet detection, developing one that remains efficient for a collision of low-power technologies, poses new challenges. First, solutions that detect packets based on an energy threshold [14] fail with low-power technologies whose signal power may be well below the noise floor. Second, correlation with each known preamble across technologies is proven to be the optimal solution [21], yet scales poorly with increased computation needed for each technology added to the system.

Technology	Modulation	Sync Preamble
LoRa [6]	CSS	sequence of 1s
Z-Wave [3]	BFSK,GFSK	m bytes '01010101'
XBee [2]	GFSK	4 bytes '01010101'
BLE [5]	GFSK	4 bytes '01010101'
WiFi Halow [4]	BPSK	configuration specific
SigFox [18]	D-BPSK	4 bytes unknown
Thread [2]	QPSK	4 bytes binary 0s
WirelessHART [2]	O-QPSK	4 bytes binary 0s
Weightless [7]	O-QPSK	4 byte binary 0s
NB-IoT [27]	OFDMA	LTE specific

Table 1: Some of the common IoT technologies with their modulation and preamble information

Approach: GalioT asks the following question: "Can we detect multiple radio technologies, including collisions by correlating against a single universal preamble sequence, no longer than a preamble of one radio technology?". An answer in the affirmative would ensure that the complexity of packet detection for multiple technologies is no greater than that of a single technology. GalioT would therefore scale efficiently with the introduction of a new radio technology, through a simple update to the universal preamble.

To design the universal preamble, we study the preambles and modulation schemes of a few most popular IoT technologies as summarized in Table 1. We note two interesting properties across preambles: (1) Several preamble sequences are common. This is, by no means, an accident, since preamble sequences are carefully chosen to be simple and amenable to correlation or energy detection and only a few such sequences exist for short lengths; (2) Most other pairs of preambles correlate poorly with each other and are often completely orthogonal. This is, again, by design, either due to differences in modulation or to prevent correlating for the preamble of one technology to be confused by another.

Inspired by these observations, GalioT's universal preamble detection takes two steps: (1) First it coalesces preambles across technologies that are common and selects the shortest representative sequence common to all these preambles; (2) Next, it constructs the universal preamble as the sum of all these representative preambles. Since the representative preambles are mutually orthogonal (or near-orthogonal), by definition, they produce distinct peaks for a single packet of any technology and multiple distinct peaks for collisions of packets across technologies.

Analysis: We explain this approach mathematically as follows. For simplicity, we ignore noise in this analysis. Let us assume n technologies in the system with preambles $P_1, P_2..P_n$ respectively. For any $j \in \{1, 2, \ldots n\}$, let us assume the preamble of the jth technology correlates with itself to provide a unique peak at location i_j. We define the auto-correlation of the preamble $C(P_j, P_j) = \delta_{i_j}$, where δ_{i_j} is the delta function with a unique peak at i_j.

Suppose two technologies j and l have different modulation schemes, then their preambles are (approximately) uncorrelated, i.e. $C(P_l, , P_j) \approx 0$. If they share the same modulation scheme, then their preambles are perfectly correlated, implying that $C(P_l, P_j) = \delta_{i_j}$. Let M denote a maximum-sized set of mutually uncorrelated preambles.

Generalizing for all technologies, let us define the universal preamble as $P = \sum_{j \in M} P_j$ with its length set to the maximum preamble length across technologies. Let P_j denote a preamble signal corresponding to the jth radio technology. Then correlation of P_j with respect to the universal preamble P is given as $C(P_j, P) = \sum_k C(P_k, P_j) = \delta_{i_t}$ for a unique $t \in M$ such that $C(P_t, P_j) \neq 0$. This implies that the universal preamble shows a unique spike, similar to that of the technology's own preamble. We then conservatively ship samples corresponding to twice the maximum packet length across technologies around the detected preamble to the cloud. This approach is hence independent of n.

Our evaluation in Sec. 7 constructs such a universal preamble for three common IoT technologies XBee, LoRa and Z-Wave which follow three different modulation schemes. In the future, we aim to generalize packet detection across the range of LP-WAN technologies in Table 1.

Edge vs. the Cloud: In addition to the cloud, GalioT can leverage edge computing to process I/Q samples from collisions. Decision of computation at edge vs. the cloud depends on multiple factors: latency requirements for the technology and the amount of computation required. Our present implementation takes a simple approach to resolve this: I/Q samples are pushed to the edge for decoding individual technologies(assuming no collisions) and shipped to the cloud only if decoding fails. Future implementations will explore factoring in SLAs to abide by quality-of-service requirements for different technologies and ensuring load-balancing between multiple edge computing nodes vs. the cloud.

We note that while GalioT detects packets corresponding to collisions of various radio technologies by correlating with a single universal preamble, it does not know which particular set of technologies exist within a collision. However, recall that the only goal of the gateway is to infer if a given sequence of I/Q samples needs to be shipped to the edge / the cloud or discarded as noise. In this respect, the gateway does not need to learn which radio technologies exist within the collision and can outsource this task to the cloud, which has significantly greater computational power. The next section describes our approach to separate the packets belonging to various radio technologies at the cloud.

5 DECODING COLLISIONS AT THE CLOUD

At the cloud, GalioT must separate collisions across multiple radio technologies. A strawman approach to do this is successive interference cancellation (SIC), where stronger signals are decoded first and subtracted out to recover weaker signals. However, SIC fails when multiple transmitters are received at low power with comparable signal strengths [28].

Approach and Filter Design: GalioT develops specialized filters, which we call "kill" filters, intended to kill a specific radio technology based on its modulation. Upon applying the "kill" filter of one technology, GalioT can iteratively decode the others and apply SIC to recover the originally killed technology. Decoding order in GalioT is dependent only on the power of the signal and is not technology dependent.

Below, we propose three such "kill" filters applicable for three popular classes of modulation techniques:

(1) KILL-FREQUENCY: GalioT identifies that despite having the same center frequencies, most low-power technologies implement a modulation that involves an uneven distribution of energies across their bandwidth of operation. FSK modulation schemes adopted by Z-Wave and many IEEE 802.15.4g implementations (e.g. XBee) distribute energy at two or more frequencies. The most common modulation schemes used here, BFSK and GFSK, produce a square wave pulse with energy concentrated at two different frequencies, making them separable in the frequency domain. GalioT exploits this opportunity to filter out these specific frequencies to eliminate

these signals. Further, uneven distribution of energy is seen even for phase modulation schemes such as BPSK, QPSK as well. These schemes concentrate energy on a specific band of operation (typically the center frequency) which makes it easier for them to be eliminated completely.

(2) KILL-CSS: Chirp Spread Spectrum (CSS), commonly used in LoRaWAN – a long range low-power technology – provides a unique challenge for collision mitigation since it distributes energy evenly across frequencies. CSS encodes each symbol as a chirp which sweeps across frequency band, with data bits encoded in the starting frequency.

GalioT identifies a unique opportunity to decode cross-technology collision of CSS to its immunity property across narrow band interferes. Specifically, each symbol encoded in CSS is essentially different cyclic frequency shifts of the elementary chirp that runs from the lowest to highest frequency across the bandwidth. GalioT multiplies the received sequence with a sequence of down chirps, which are inverted elementary chirps that run from the highest to the lowest frequency of operation. The resulting product appears similar to a narrowband signal reception centered at frequencies corresponding to the starting frequencies of various chirps. These can now be efficiently canceled out from the received signal, akin to the KILL-FREQUENCY approach above.

(3) KILL-CODES: Several IoT technologies encode transmissions as specific codes, for whom orthogonal codes exist. By simply applying the well-known orthogonal code to these transmissions as a filter, one can eliminate their effect. We can then decode other technologies by applying the same codes to various possible bit sequences. This allows for GalioT to recover other useful transmissions in the collision.

Our implementation focuses on decoupling three diverse radio technologies that follow the above classes of modulation, demonstrating the promise of our approach. In the future, we aim to create a generalized set of filters that span a wide-range of available IoT radio technologies (Table 1). We will also investigate the extent to which some of these filters could be preemptively applied at the gateway or the edge. We will further study the limits of our approach in decoding collisions at a range of SNRs, particularly at certain SNR regimes (e.g. extremely low values) where the Shannon limit may not permit decoupling collisions.

Formal Algorithm: The pseudo-code in Alg. 1 presents our algorithm more formally.Note that here, $S_1, S_2, \ldots, S_n$ represent the n signals which have collided and $\mathcal{P}(S_i)$ represents the power of signal S_i.

Algorithm 1 Decoding at the cloud

1: **procedure** CLOUDDECODE(S) ▷ S: Received Signal
2: **if** $S = S_i$ **then**
3: Decode(S_i); ▷ No collision
4: **else** Pick $S_i | \mathcal{P}(S_i) > \mathcal{P}(S_j)$
5: **if** Decode(S_i)=True **then**
6: Cancel S_i from S and repeat step 4 ▷ SIC
7: **else** find S_j with least power orthogonal to S_i.
8: **if** $S_j \in$ FSK or PSK **then**
9: KILL-FREQUENCY(S_j), go to step 5
10: **else if** $S_j \in$ CSS **then**
11: KILL-CSS(S_j), go to step 5
12: **else if** $S_j \in$ Orthogonal codes **then**
13: KILL-CODE(S_j), go to step 5
14: **else** find next least S_j, repeat from step 9
15: **if** Decode(S)=False **then**
16: $S_i \leftarrow$ next highest powered signal, Repeat from step 5

6 DESIGN SPACE – OPEN QUESTIONS

Below, we summarize long-term open research questions in various aspects of the GalioT framework that we plan to investigate at: the gateway, the backhaul and the cloud.

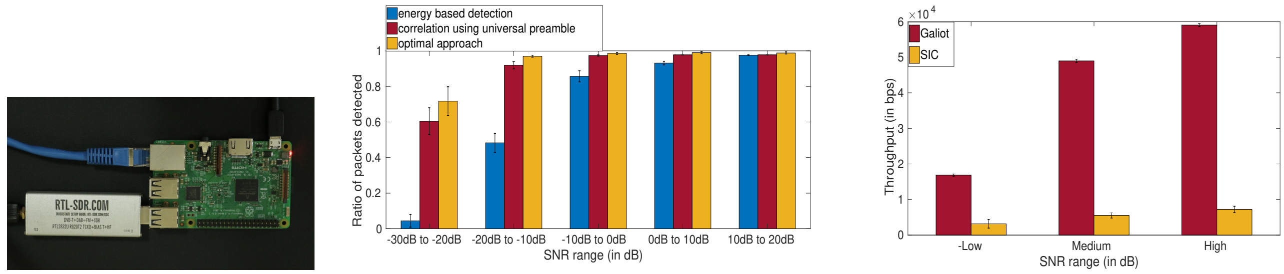
Multi-Technology Programmable Gateway: Our current gateways are commercially available SDRs that operate over limited bandwidth (few Megahertz). Spanning all unlicensed spectrum in the 900 MHz and potentially higher bands (e.g. 2.4 GHz) will require either replicated front-ends or new hardware that spans a wide bandwidth. It is interesting to also explore intermediate approaches for low-power technologies such as frequency hopping with a few frontends that dynamically learns the schedule or hardware that aliases a wide-range of frequencies into a smaller bandwidth that is easier to decode with low-cost ADCs, but at the expense of more collisions on occasion.

Limited Backhaul – Compute, Compress or Ship?: Each GalioT gateway needs to make an informed choice of whether to directly compute the data from I/Q samples locally or ship the samples directly to the cloud. While our current approaches favor the latter, it is interesting to explore hybrid solutions using edge detection, resolving the limitations of current detection and compressions schemes at gateway to conserve bandwidth and improve link utilization.

At the Cloud – Multi-Technology Wireless Sensing: GalioT aggregates I/Q samples from many IoT technologies at the cloud primarily for collision-mitigation. However, as recent work on wireless sensing using Wi-Fi have shown [8], wireless channels retrieved from I/Q samples carry information about a variety of events such as occupancy of a room, activities such as typing on a keyboard or movement of a user. Low-power IoT devices bring a new challenge to wireless sensing, given that they are diverse, transmit occasionally and are impacted by their hardware imperfections. Yet, collectively devices are also inexpensive, and pervasive, meaning that several heterogeneous wimpy devices may collectively offer more insights than one high-power wireless node.

7 PRELIMINARY RESULTS

We implement GalioT using an RTL SDR- Raspberry Pi platform connected to a laptop via an ethernet cable (see Fig. 3(a)). The RTL-SDR is configured to receive samples at a bandwidth of 1 MHz. We provide the proof of concept using three technologies, which provide support for 868MHz frequency band- LoRa, XBee and Z-Wave. We use SemTech SX1276 for facilitating LoRa, TI CC1310 for XBee and UZB static controller for Z-Wave transmissions. Duty cycle of the



(a) Prototype of GalioT

(b) Packet Detection

(c) Decoding Collisions

Figure 3: Depicts (a) the GalioT prototype and preliminary results on (b) packet detection, (c) decoding collisions.

devices have been adjusted to capture all possible scenarios, including intertechnology collisions.

Local Detection: Local detection at the gateway involves RTL-SDR continuously listening to the 868MHz channel, sampling over a 1 MHz bandwidth. Three transmitters, one corresponding to each technology, are configured to transmit intermittently at low-power. We use these three technologies specifically to justify our argument on how the universal preamble construction is unaffected by the similarities and dissimilarities in their modulation schemes. LoRa and XBee follow completely different modulation schemes while XBee and Z-Wave follow similar modulation schemes with similar preamble structure (as explained in Table 1). We collect RTL-SDR traces observing packets with and without collision under additive white Gaussian noise, with received SNRs from -30dB to 20dB. We create a universal preamble by adding up the preambles of all technologies with zeros padded at the end to compensate for unequal preamble lengths. We compare the performance of our scheme with existing energy detection scheme as well as the optimal correlation-scheme with individual preambles that is computationally-intensive.

Results: Figure 3(b) shows the average performance of each of the detection techniques under varying levels of noise. At low noise scenarios (SNR above 0dB) , universal preamble detection performs close to optimum, detecting even the collided packets. It is observed that universal preamble detection is as resilient to high noise scenarios as the optimal scheme, and there is a small drop in detection accuracy against even at -30dB. The small drop was primarily contributed by the failure in detection of the second packet in certain instances of collisions, even as at least one packet in a collision is very likely to be detected. Energy based detection schemes proposed in the existing multi-technology literature [14] is seen to scale poorly with increasing noise levels. At SNR below 0dB, there is a sharp drop in detection all the way from a total of 84% to 0.04%, while our universal preamble detection maintains a detection accuracy of 62% even at -30dB SNR. It is also seen that the universal preamble has higher susceptibility to the white noise in comparison with the individual preamble. Hence it will be interesting to refine the technique of universal preamble detection so as to provide better noise resilience, especially when more technologies are added into the system - a task for future work.

Decoding at the cloud: The signal streamed in real-time from the gateway across the three technologies-LoRa, XBee and Z-Wave are collected at the cloud, where the nature of each signal is identified by comparing the received signal with each of the preambles. Uncollided signals are processed directly by passing it to the corresponding demodulator while collided signals are passed through various kill filters as described in Algorithm 1. We stress the system in the presence of additive white Gaussian noise to study how the performance of the filters scale across varying noise levels. We study the performance of GalioT against a baseline successive interference cancellation system.

Results: Figure 3(c) shows the comparison of throughput across varying SNR regimes while implementing SIC with and without our kill filters. Our filters have resulted in an improvement in throughput gain by 818.36 % in high SNR (¿20 dB) and 532.4% in low SNR (¡ 5 dB). The high throughput gains stem from opportunities to transmit at higher data rates, which for low-power devices translate to large throughput gain and battery savings even from transmitting at one rate higher. We note that the gains at high SNR are slightly higher owing to increased opportunities for collision decoding due to the higher Shannon limit.

8 CONCLUSION AND FUTURE WORK

This paper introduces GalioT a programmable gateway architecture for low-power Internet of Things. With the ability to decode cross-technology collisions, GalioT can prove an effective replacement in cost, flexibility and scalability against the currently existing multi-technology gateway platforms. A prototype evaluation using three popular IoT technologies reveals an increase in average throughput by 745.96%. Future work will seek to improve GalioT by: (1) Demonstrating a large number of IoT technologies; (2) Test the scaling limits of collision-decoding. (3) Exploring the design space for innovation at the gateway, backhaul and cloud including multi-technology wireless sensing (see Sec. 6).

Acknowledgements: We thank NSF (grants 1657318 and 1718435) for support. We also thank Prof. C Siva Ram Murthy and Mrs. Poornima Swarun for their valuable support during the project.

REFERENCES

[1] The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[2] IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4 TM-2015 (Revision of IEEE Std 802.15.4-2011)*, 2015.

[3] Short range narrow-band digital radiocommunication transceivers - PHY, MAC, SAR and LLC layer specifications. *ITU-T G.9959*, 2015.

[4] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation). *IEEE Std 802.11ah-2016*, 2016.

[5] BLE Specifications. *Bluetooth SIG*, 2017.

[6] LoRaWAN 1.1 Specification. *LoRa Alliance*, 2017.

[7] Weightless-P System specification Version 1.03. *Weightless*, 2017.

[8] F. Adib and D. Katabi. See through walls with wifi! SIGCOMM, pages 75–86, 2013.

[9] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, and C. Savaglio. A mobile multi-technology gateway to enable iot interoperability. In *IoTDI*, 2016.

[10] C. J. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L. M. Contreras, H. Jin, and J. C. Zúñiga. An architecture for software defined wireless networking. *IEEE wireless communications*, 21(3):52–61, 2014.

[11] 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 *IPSN*, pages 60–71. IEEE Press, 2018.

[12] J. Elias, S. Paris, and M. Krunz. Cross-technology interference mitigation in body area networks: An optimization approach. *IEEE Transactions on Vehicular Technology*, 64(9):4144–4157, 2015.

[13] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: Making 802.11n robust to cross-technology interference. *SIGCOMM Comput. Commun. Rev.*, 41(4):170–181, 2011.

[14] A. Hithnawi, H. Shafagh, and S. Duquennoy. Tiim: technology-independent interference mitigation for low-power wireless networks. In *IPSN*. ACM, 2015.

[15] S. S. Hong and S. R. Katti. Dof: a local wireless information plane. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 230–241. ACM, 2011.

[16] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *MobiCom*, pages 317–330, 2015.

[17] S. M. Kim, S. Wang, and T. He. IoT networking: From coexistence to collaboration. In *RTCSA*, pages 212–217, 2016.

[18] . Morin, M. Maman, R. Guizzetti, and A. Duda. Comparison of the device lifetime in wireless networks for the internet of things. *IEEE Access*, 5:7097–7114, 2017.

[19] T. R. Mück and A. A. Fröhlich. Hyra: a software-defined radio architecture for wireless embedded systems. In *10th international conference on networks*, pages 246–251, 2011.

[20] MultiTech. *Data Sheet: MultiConnect Conduit: Programmable Gateways (Wi-Fi/Bluetooth/GNSS)*.

[21] S. Nagaraj, S. Khan, C. Schlegel, and M. Burnashev. On preamble detection in packet-based wireless networks. In *ISSSTA*, pages 476–480. IEEE, 2006.

[22] R. Narayanan and C. S. R. Murthy. A probabilistic framework for protocol conversions in iiot networks with heterogeneous gateways. *IEEE Communications Letters*, 21(11):2456–2459, 2017.

[23] J. Nieminen, C. Gomez, M. Isomaki, T. Savolainen, B. Patil, Z. Shelby, M. Xi, and J. Oller. Networking solutions for connecting bluetooth low energy enabled machines to the internet of things. *IEEE Network*, 28(6):83–90, 2014.

[24] N. Ramanathan, M. D. Yarvis, J. Chhabra, N. Kushalnagar, L. Krishnamurthy, and D. Estrin. A stream-oriented power management protocol for low duty cycle sensor network applications. 2005.

[25] M. Surligas, A. Makrogiannakis, and S. Papadakis. Empowering the iot heterogeneous wireless networking with software defined radio. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–5. IEEE, 2015.

[26] I. Tinnirello, D. Croce, N. Galioto, D. Garlisi, and F. Giuliano. Cross-technology WiFi/ZigBee communications: Dealing with channel insertions and deletions. *IEEE Communications Letters*, 20(11):2300–2303, 2016.

[27] G. T. . V15.2.0. Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran); (release 15). 2018.

[28] S. P. Weber, J. G. Andrews, X. Yang, and G. De Veciana. Transmission capacity of wireless ad hoc networks with successive interference cancellation. *IEEE Trans. on Information Theory*, 53(8):2799–2814, 2007.

[29] 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 Trans. on Mobile Computing*, 15(4):1009–1021, 2016.

[30] C. Zhang, P. Xie, D. Li, J. Zhang, and R. Yu. Wireless home gateway: software-defined radio architecture and applications. 2011.

[31] X. Zhang and K. G. Shin. Gap Sense: Lightweight coordination of heterogeneous wireless devices. In *INFOCOM*, pages 3094–3101, 2013.

[32] Y. Zhang and Q. Li. Howies: A holistic approach to ZigBee assisted WiFi energy savings in mobile devices. In *INFOCOM*, pages 1366–1374, 2013.