High-Throughput and Robust Rate Adaptation for Backscatter Networks

Si Chen, *Student Member, IEEE*, Wei Gong, *Member, IEEE*, Jia Zhao, *Graduate Student Member, IEEE*, and Jiangchuan Liu, *Fellow, IEEE*

***Abstract*— Recently backscatter networks have received booming interest because, they offer a battery-free communication paradigm using propagation radio waves as opposed to active radios in traditional sensor networks while providing comparable sensing functionalities, ranging from light and temperature sensors to recent microphones and cameras. While sensing data on backscatter nodes has been seen on a clear path to increasing in both volume and variety, backscatter communication is not well prepared and optimized for transferring such continuous and high-volume data. To bridge this gap, we propose a high-throughput rate adaptation scheme for backscatter networks by exploring the unique characteristics of backscatter links and the design space of the ISO 18000-6C (C1G2) protocol. Our key insight is that while prior work has left the downlink unattended, we observe that the quality of downlink is affected significantly by multipath fading and thus can degrade the uplink and overall throughput considerably. Therefore, we introduce a novel rate mapping algorithm that chooses the best rate for both the downlink and uplink. Also, we design an efficient channel estimation method fully compatible with the C1G2 protocol and a reliable probing trigger, substantially saving probing overhead. To combat interference, we further design an interference detector using clusters and lightweight countermeasures to make rate adaptation more robust. Our scheme is prototyped using commercial RFID readers and tags. The results show that we can achieve up to 2.6× throughput gain over state-of-the-art approaches across various mobility, channel, network-size, and interference conditions.**

***Index Terms*— High-throughput, backscatter network, RFID, rate adaptation.**

I. Introduction

THERE is a long-standing vision of ultra-low power ubiquitous sensor networks where many tiny sensors are wirelessly connected and can perform continuous sensing tasks without human intervention, e.g., Smart Dust [1]. Backscatter networks are one of the most promising candidates to realize this goal as backscatter nodes -like RFID tags- can capture power from propagation radio waves, making battery-free networks possible. Thanks to the advances of energy efficiency scaling for microelectromechanical systems, a wide range of applications that previously are only supported by battery-assisted sensors become available for backscatter networks, such as temperature or light intensity sensing [2], acoustic signal capturing [3], and even video surveillance [4]. While backscatter networks have seen the future of increasing sensing data coming in, backscatter communication that supports continuous and high-throughput transmission is not quite ready yet. Recently there have been several attempts that focus on revamping the traditional backscatter protocols for more efficient transmission [5]–[7]. Yet incompatibility with industry standards, e.g., ISO 18000-6C (C1G2) specification, and requirements of customized hardware hinder wide adoption of those proposals. As such, we aim to design a high-throughput protocol that is fully compatible with C1G2 using Commercial Off-The-Shelf (COTS) devices, which can benefit tons of currently deployed backscatter devices. To achieve this, however, there are several key challenges:

- *Ineffective Rate Selection*: Prior work of rate selection for backscatter networks only focuses on the uplink that is for transmitting sensor data [8], [9], leaving the impact of downlink rates largely uninvestigated. Actually, the downlink is indispensable and implicitly involved in the uplink transmission because any uplink has a downlink as its *predecessor*, which means if the downlink fails due to incorrect rate settings, the uplink would be discontinued. This is the unique characteristic of the backscatter link that a downlink and an uplink are sequentially combined as a backscatter link. Therefore, if the downlink rate is left unattended, even the optimal setting for the uplink may not bring overall throughput gain.
- *Probing Overhead*: In backscatter networks, all transmissions are scheduled by the reader through an ALOHA-like MAC protocol because nodes cannot sense each other. The performance of channel probing would severely degrade due to MAC collisions when the node population increases [8]. Although CARA [9] proposes an estimation algorithm to compensate such collisions, the probing process still needs to follow the above MAC scheduling, prolonging the probing time. In addition, the probing trigger, which is necessary for deciding when to probe, could exacerbate the issue. For example, Blink [8] requires measurements of at least 10 channels for its trigger, and CARA needs to probe at least 5 channels.

Manuscript received July 24, 2019; revised November 20, 2019 and February 20, 2020; accepted June 11, 2020; approved by IEEE/ACM Transactions on Networking Editor X.-Y. Li. Date of publication July 1, 2020; date of current version October 15, 2020. This work was supported in part by the NSFC under Grant 61932017 and Grant 61971390, in part by the Fundamental Research Funds for the Central Universities under Grant WK2150110013, and in part by the Canada NSERC Discovery Grant. *(Corresponding author: Jiangchuan Liu.)*

Si Chen, Jia Zhao, and Jiangchuan Liu are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: sca228@sfu.ca; zhaojiaz@sfu.ca; jcliu@sfu.ca).

Wei Gong is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: weigong@ustc.edu.cn).

Digital Object Identifier 10.1109/TNET.2020.3002876

- *Limited Visibility for Channel Estimation*: While it is common that PHY hints for channel estimation, e.g., bit error rate (BER), are not available for most of the COTS wireless devices, it becomes even worse when we deal with COTS readers; even the packet level loss rate is very difficult to obtain because COTS readers only report the number of successful reads in a time interval. Previous solutions either use an extra monitoring device, like USRP, to sniffer messages transferred in the air, or log commands from the reader into tags' EPC memory using Computational RFIDs (CRFID). Yet these methods not only introduce more cost due to additional hardware but also are inapplicable to situations where only COTS devices are available.

To address the above issues, we propose a high-throughput Rate Adaptation framework for Backscatter networks, RAB. It is fast and efficient while being compatible with the C1G2 protocol and existing commercial RFID readers. To do so, it primarily makes three fundamental optimizations over the current standard. First, our work provides insights that both the uplink and downlink affect the overall throughput significantly, which motivates us to adapt rates for both in contrast to prior work that only focuses on the uplink [5], [8], [9]. Second, we describe a novel channel estimation method that uses filter-based probing to effectively reduce errors brought by MAC-layer collisions and estimates the loss rate by leveraging the link timing features of the C1G2 protocol. Third, we present a correlation-based channel hopping and an accurate mobility detection approach that uses PHY hints to determine when to trigger channel estimation, considerably saving channel-probing overhead. Fourth, we design an effective interference detector based on rate mapping clusters and a robust rate-selection scheme to deal with rate distortion brought by interference sources.

We build a prototype of RAB using a Thingmagic reader and 100 Alien Higg3 tags. We compare RAB with Blink and CARA and results show that across 120 traces with different mobility, channel, and network-size conditions, RAB achieves overall throughput gains of 2.6× over Blink and 2× over CARA on average. This gain comes from two sources: First, RAB reduces probing cost significantly by 9.1× compared to Blink, and by 4.8× compared to CARA; Second, for data transmission, our rate selection scheme achieves throughput gains of 1.8× over Blink and 1.6× over CARA.

Contributions: We present RAB, a novel rate adaptation for backscatter networks that for the first time investigates the impact of downlink to the overall rate selection. As a result, RAB improves throughput based on both uplink and downlink variations. A complete robust link layer design is demonstrated throughput extensive experiments.

II. Backscatter Primer

Backscatter System: A backscatter system usually is composed of a reader and one or more backscatter nodes,[1] e.g., RFID tags. The reader initiates the communication by transmitting carrier waves, which serves two purposes. First, the tag can capture energy from the radios waves and power itself for computation and communication. Second, the tag backscatters information bits by modulating the same carrier waves. While many of the principles are generally applicable to all RFID devices, here we focus on the UHF RFID devices whose behaviors are defined in the C1G2 protocol [10].

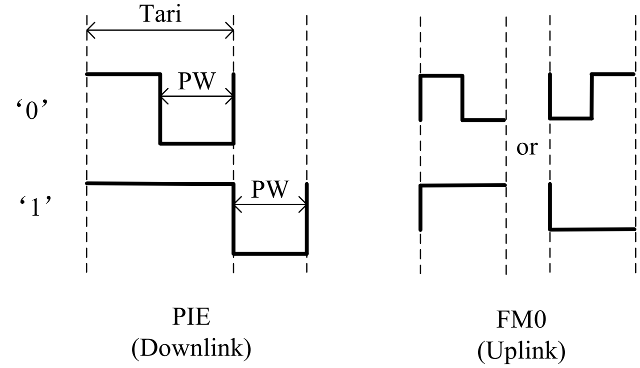


Fig. 1. Examples of downlink and uplink symbols. The downlink rate, ranging from 40 to 160 kbps, is controlled primarily by the length of Tari; The uplink rate, ranging from 5 to 640 kbps, mainly depends on encoding schemes (FM), Miller2/4/8) and backscatter link frequencies.

Backscatter Link: While the reader is usually assumed powerful, the tag is restricted in terms of computation, communication, and hardware capabilities since it can only capture limited power from radio waves. Therefore, the asymmetry exists almost everywhere in backscatter systems including backscatter links. For example, the tag typical has a dipole antenna with a gain of 2.1 dBi and a sensitivity of -13 dBm, while the reader is with a circularly polarized antenna that has a gain of 9 dBi and a sensitivity of -80 dBm. Accordingly, the downlink symbols are amplitude-modulated Pulse Interval Encoding (PIE) symbols, which are easy to decode because an analogy comparator is enough. As shown in Figure 1, downlink symbol '0' is composed of a power-on interval and a power-off interval of equal length. The total length of symbol '0' defines Tari (Type A Reference Interval) and PW (pulse width) is half of Tari. A symbol '1' differs from '0' only in the power-on interval length; The total duration of '1' should be more than 1.5Tari and less than 2Tari. The C1G2 protocol specifies the typical values of Tari: 6.25, 12.5, and 25 μs, which correspond to downlink rates of 160, 80, and 40 kbps.[2] In contrast, the uplink data rate is configured by setting BLF (Backscatter Link Frequency) and different encoding schemes (FM0, M2/4/8). For example, if the uplink is set at a BLF of 250 kHz using Miller2, its data rate is 250/2 = 125 kbps. Note that both rates of uplink and downlink are controlled by the reader.

C1G2 Protocol: The C1G2 protocol specifies how the reader interrogates tags through several rounds of handshaking. We briefly describe its data reading as follows.[3] As shown in Figure 2, basically the reading process includes two phases: ID transfer and Data transfer. First, the reader starts by transmitting a *QUERY* command that contains a *Q* parameter,

[1] We use sensors and tags interchangeably in this paper.

[2] These are maximum rates assumed all symbol-0s.

[3] For more details please refer to [10].

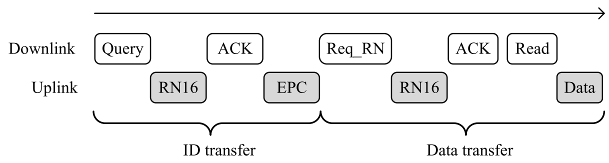


Fig. 2. Reading data from a tag following the C1G2 protocol. The reading process includes an ID transfer phase and a Data transfer phase, each of which has a handshaking through several different commands.

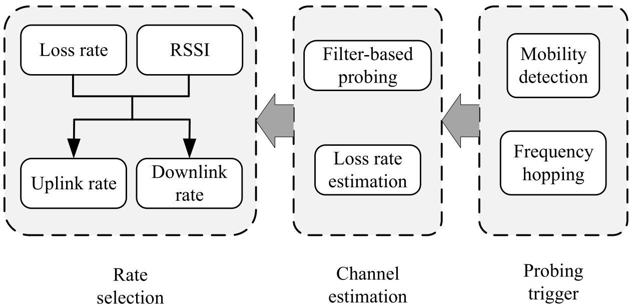


Fig. 3. The framework of our rate adaption scheme including three modules: rate selection, channel estimation, and probing trigger.

which specifies how many slots are included in a query round. Then the tag would choose a random number in [0,2^Q-1) as its slot counter. If this counter is equal to 0, the tag replies a 16-bit random number (*RN16*); otherwise, the counter decreases 1 after each *QUERY*/*QUERYREP*. On receiving the RN16, the reader sends an *ACK* that contains the decoded RN16 to the tag. If the tag confirms the reader-decoded RN16 is correct, it backscatters an identifier, *EPC* (typically 96 bits). This is the end of the ID transfer phase. If the reader needs data from the tag, it starts another round of handshaking through *REQ_RN*, *RN16*, and *ACK* messages. If this round of handshaking goes well, the tag would reply with the memory data upon receiving a valid *READ* command.

Our focus in this paper is to choose optimal rates for both the uplink and downlink that can maximize the overall throughput while conforming to the C1G2 protocol. Optimizations from other aspects, such as rateless coding, energy efficiency, or the fairness of MAC, are out of this paper's scope and thus are not considered.

III. Overview

Figure 3 presents the framework of RAB. The cornerstone of RAB is our observation that we should adapt data rates for both the downlink and uplink to maximize throughput. While common wisdom says that the uplink rate should be properly chosen to improve the throughput of the backscatter link, we argue that the downlink rate should be treated in the same way as there is a tradeoff in setting the downlink rate. Our experiments show that too slow downlink rates could lose the chance to increase throughput when the channel is good, which motivates us to increase the downlink rate. At the same time, we also observe that too aggressive downlink rates can bring down the throughput even to 0 when a bad channel is present because of the well-known sharp transition between low and high loss rates [11] due to multipath fading. By using a rate mapping algorithm, we choose the optimal rates for both the uplink and downlink using overall loss rates and RSSIs that capture multipath fading and path loss, respectively.

While RSSIs are the standard output of most readers, loss rate measurements are not readily available. To measure the loss rate accurately, we introduce a filter-based probing scheme that avoids the potential MAC collisions of multiple tags and thus is able to achieve fast probing regardless of the tag population. To do so, we leverage the built-in *SELECT* command provided by the C1G2 protocol, making our probing lightweight and suitable for point-to-point measuring. In addition, we design a link timing based loss-rate estimation to overcome the invisibility brought by the programming interfaces of COTS readers. Link timing is another unique characteristic of backscatter communication, which ensures the compatibility of devices from different manufacturers. By using such link timing structure, we can accurately approximate how many queries have been sent and thus derive the loss rate.

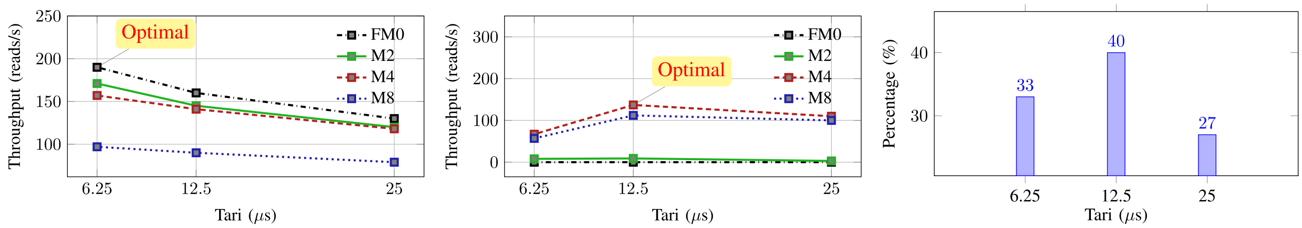
The final module is to answer a question: when to probe. We design a reliable probing trigger to further reduce the probing cost by combing a PHY-assisted mobility detection and a correlation-based channel hopping. In our mobility detection, we mainly make use of a PHY-hint, *phase*, which is widely used in many localization schemes and supported by all COTS readers and the LLRP standard [12]. Differing from [8], [9], it is lightweight and does not need measurements from multiple channels. Channel hopping is another time window for probing. We present a fast channel hopping that is based on the observation that good/bad channels tend to get together instead of being randomly distributed in the spectrum. Therefore, our strategy is that staying away from the probed bad channel and sticking around the good channel.

IV. Rate Selection

A. Backscatter Link Characteristics

As discussed before, a backscatter link consists of a downlink that is Reader-to-Tag and an uplink that is Tag-to-Reader. Prior work mainly focuses on adapting appropriate rates for the uplink for two reasons. First, the path loss fading of an uplink is more severe than its corresponding downlink because, while power decays with the square of distance for the downlink, it decays with the fourth power of distance for the uplink. Second, the uplink is supposed to transfer more important data, like sensing information, while the downlink is more viewed as a way to disseminate parameters/commands. However, a key point that is largely ignored is that if there is anything wrong with the downlink, e.g., decoding errors, the corresponding uplink would be discontinued, leading to handshaking failures.

From previous sections, we know that the downlink rate can be set by adjusting the value of Tari. To examine the impact of different Tari values on the throughput, we keep a tag at a fixed place and BLF = 250 kHz. Then we vary different encoding schemes for the uplink link. The results are shown in Figure 4a. This is a link with good channel quality where faster rates have better throughput. The optimal rates in this case are Tari = 6.25 for the downlink and FM0 for the uplink. Therefore in the case of good channels, we would



(a) A good channel (b) A bad channel (c) Distribution of optimal downlink rates

Fig. 4. To examine the impact of data rates of both the uplink and downlink, we measure throughput with various settings. (a) is an example of a good channel, which favors the fastest uplink rate (FM0) and downlink rate (Tari = 6.25); (b) is an example of a bad channel. Specifically, both FM0 and M2 encoding settings do not work, and the performance of Tari 6.25 is even worse than that of Tari 12.5, which suggests Tari 6.25 is an aggressive choice. (c) is the distribution of optimal Tari values across 100 random locations, showing that there is no single Tari value that is dominating.

miss the chance to increase throughput if a conservative Tari is chosen. For example, with M2 for the uplink, the throughput of Tari = 6.25 is 171 reads/s, but it drops to 120 reads/s with Tari = 25. This observation motivates us to use the fastest rate for maximizing throughput. However, this is not always the case. As we move the tag to an 1-meter away location, we observe different behaviors. As shown in Figure 4b, this time the link is experiencing some difficulties because the throughput of both FM0 and M2 encoding schemes is almost 0. In this case, the optimal rates become that Tari = 12.5 for the downlink and M4 for the uplink. This case tells us that too aggressive rates would not benefit but hurt overall throughput in the case of not good channels. In addition, we measure links at 100 random locations and plot the distribution of optimal Tari values in Figure 4c, which shows that there is no single Tari value that is dominating. To summarize, the above observations suggest that the optimal Tari should be carefully chosen to maximize the throughput based on the quality of channels.

B. Rate Mapping

To find the optimal rates for the uplink and downlink, we adopt a classification-based approach that takes loss rates and RSSIs as input. Although RSSIs are inaccurate in measuring backscatter signal strength due to self-interference [8], they are still useful in indicating path loss. At the same time, the overall loss rate entails multipath fading for both the uplink and downlink. This feature is very important because our hypothesis is that multipath fading is the main reason that the aggressive rate, Tari = 6.25, would not always be the optimal rate for the downlink where path loss is less of a problem.

Our rate selection map is built as in Figure 5. The intuition behind this mapping is that when the loss rate increases, more complex encoding schemes should be introduced for resisting channel errors; when the RSSI decreases, the lower-throughput uplink is used to combat path loss. In addition, the impact of both the uplink and downlink under multipath fading is accounted into the loss rate. Therefore, this mapping essentially is able to deliver accurate and fast rate selection. While classes in Figure 5 are only for illustration, the real sizes and types of classes are empirically learned through a training set collected in indoor environments. After all the classes are established (class center and distance), we map a new pair of measured loss rate and RSSI to the closest class.

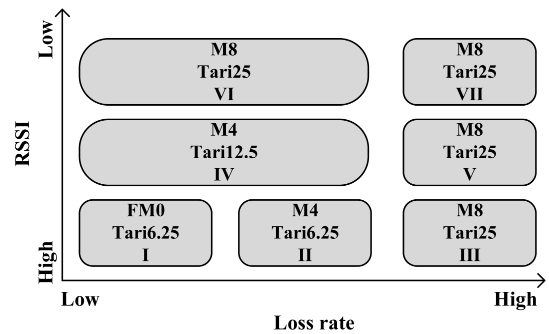


Fig. 5. Optimal rate map of the uplink and downlink. When RSSIs decrease, we choose the downlink with lower throughput. When loss rates increase, we use slower encoding schemes of the uplink to combat the interference. Note that BLF is not considered here for simplicity.

V. Channel Estimation

For rate selection, we assume that the loss rate is known. However, it is not readily available in practice. In this section, we show how to efficiently probe and estimate the loss rate.

A. Filter-Based Probing

Previous work of backscatter channel probing is neither accurate nor efficient. The inefficiency of Blink and CARA comes from the C1G2 MAC that is designed for tags that cannot sense each other because probing packets still need to follow the same MAC. There have been many solutions on how to overcome such inefficiency [5], [13]. While those efforts achieve significant efficiency by overhauling the C1G2 MAC, they are overkill for just channel probing. Furthermore, those solutions bring inevitable incompatibility with the C1G2 protocol and thus lose interoperability with many COTS tags.

Our solution for this is that we make use of the built-in *SELECT* command of the C1G2 protocol to create a filter for probing. The *SELECT* command is designed for choosing a tag population for inventory and access. One or more tags are selected by the reader according to user-specified criteria, which is analogous to selecting records from a database. In a *SELECT* command, the reader can specify which *Memory Bank* to match, the associated starting address and length, and a *MASK*. There are four types of memory banks: Reserved, EPC, TID, and User memory. For example, if we know a tag's ID in advance, then we can easily make it selected by simply sending a *SELECT* command specifying the memory bank as EPC, starting address as 0, length as 96, and *MASK* as the wanted tag's ID. This way, only the tag that matches the mask would reply. Note that this method requires the ID information before probing. As our goal is to maximize the throughput for reading sensor data, we should know which sensor we would like to collect data from in advance. Even sometimes we may not know the sensor's ID beforehand, as shown in Figure 2, the data transfer phase is always preceded by an ID transfer phase. Therefore, knowing the ID of a sensor before transferring the data is not a problem for us. For the rest of the paper we assume the IDs of tags are known before reading sensor data.

Now by using the *SELECT* command, we enable a point-to-point probing style that avoids MAC collisions completely. Usually, a *SELECT* command is about 45-bit long (excluding the *MASK*), which incurs some extra cost. However, such cost is considerably less than the waste due to the inefficient MAC. To better understand the performance improvement of our selective probing, we can examine the time complexity for different methods. It is easy to see that the time complexity of probing for RAB is $\mathcal{O}(n)$, where n is # of tags. For Blink and CARA, besides the C1G2 standard, they haven't mentioned any other settings or optimizations, so we assume they follow the standard frame slotted aloha model. During probing, if collisions happen, that would be counted towards packet loss because commercial readers cannot distinguish a packet loss due to bad channel or colliding. Hence, the minimal probing criteria is to probe each tag for at least once while making collisions as few as possible. Although both Blink and CARA do not specified the frame length setting for probing, by following this criteria and frame slotted aloha model, we can formulate this probing problem as the famous "birthday problem". From literature [14], [15], we know that when the frame length is $\mathcal{O}(n^2)$, collisions can be avoided with high probability, i.e., with a high probability, $h(t_i) \neq h(t_j)$ for all $i \neq j$, where $h(x)$ is the hashed frame slot index, t_x denotes the x-th tag. Note that the time complexity of this probing problem cannot be simply deduced from the well-known maximum throughput for the frame slotted aloha model that when both the frame length and number of tags are n, $\frac{1}{e} * n$ singletons can be achieved. Even simply increasing the frame length and tag population to en, the achieved $\frac{1}{e} * en = n$ singletons cannot guarantee that n tags of interest are correctly probed because collisions still exist and actually what we need to probe are en tags. Hence, multiple frames are needed even with $\frac{1}{e}$-efficiency for each frame and the total time cost would still be $\mathcal{O}(n^2)$ according to [14], [15]. In addition, such a multi-round probing scheme has a big disadvantage; it is very difficult to probe the same tag for multiple times due to randomly chosen slots in each frame. So for future comparison, we use a single frame of $\mathcal{O}(n^2)$ for Blink and CARA. While optimizations of this $\mathcal{O}(n^2)$ complexity are possible for different application needs, it is out of the scope of this paper. So we leave this discussion for future work since we already provide a solution with $\mathcal{O}(n)$ complexity.

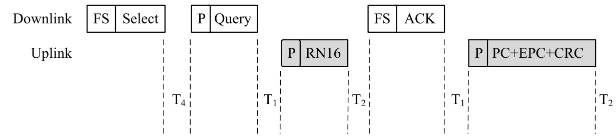


Fig. 6. Link timings of a probe. The C1G2 protocol has strict timing requirements for each message, giving us opportunities to estimate loss rates. P denotes either an uplink or downlink Preamble. FS denotes the Frame-Sync symbol.

B. Loss Rate Estimation

After probing, the next step is to estimate the loss rate of the link. Usually the loss rate can be estimated through the following

$$r_l = \frac{p_{\mathrm{rec}}}{p_{\mathrm{prb}}},$$

where r_l is the estimated loss rate, p_{prb} is the number of packets probed in a time interval, and p_{rec} is the number of packets received in that time interval. While p_{rec} is easy to obtain for all kinds of RFID devices, things are different for p_{prb}. For USRP-based readers, p_{prb} is not a problem as they are fully programmable. COTS readers, however, do not offer the way to obtain how many packets are sent or measure the loss rate. In other words, they are more like a black box and all we know is the probing time interval. Therefore, we need to estimate how many probes/queries sent in a given period of time. The time cost of a probing process can be modeled as follows

$$t_p = p_{\mathrm{prb}} * t_{\mathrm{prb}} + p_{\mathrm{rec}} * t_{\mathrm{rec}} + t_d,$$

where t_p is the total probing cost for a tag, t_{prb} is the time cost for a single probing packet, t_{rec} is the time cost for a single received packet, t_d is the composite built-in protocol delay. The unknowns are p_{prb} and t_d.

To estimate p_{prb}, our first step is to take into account of the data rate and the amount of data to be sent over both the uplink and downlink. Then we need to find certain delays built in the protocol, as shown in Figure 6. The first specified timing limitation is T_4, which is the time that the reader has to wait before issuing another command. The length of T_4 is 2RTcal, where $\mathrm{RTCal} = 0_{\mathrm{length}} + 1_{\mathrm{length}}$. After the *QUERY* command, the tag needs to wait for T_1, of which the nominal value is MAX(RTCal, $10\mathrm{T}_{\mathrm{pri}}$), where $\mathrm{T}_{\mathrm{pri}} = 1/\mathrm{BLF}$. If there is a reply from the tag, the reader must acknowledge it within T_2, ranging from [$3\mathrm{T}_{\mathrm{pri}}$, $20\mathrm{T}_{\mathrm{pri}}$]. T_1 and T_2 also apply to the *ACK* and *EPC* messages.

If we set Tari = 6.25 and FM0 encoding, a probe would take about 2.5 ms, corresponding to 400 probes/second. However, in the field study, our measured result is around 250. This is because there is a hardware-dependant command delay between two probes. Besides this uncertain hardware-dependent delay, we model all uncertain parameters in the protocol into a linear system, including T_1, T_2, T_4, and 1_{length}. To build the linear system, we make multiple measurements across different settings and use the *constrained least square* method to estimate unknowns.

Note that while many prior efforts try to solve this loss estimation problem, they all need extra hardware. For example, Flit [13] logs all the message counts into EPC using CRFIDs; [16] uses an extra USRP-based monitor. In contrast, we observe an opportunity to use the precise timing structures that are specified in the C1G2 protocol and thus make is compatible with commercial readers.

VI. Probing Trigger

The probing trigger decides when to probe the channel, which is very important because too often probing poses unnecessary overhead and too rare probing would lose the chance to adapt rates. Our probing trigger includes two indicators: mobility detection and channel hopping.

A. Mobility Detection

When a sensor moves to another location, its channel inevitably changes. At this time, the natural thought is that a reader may need to choose the optimal rate for this new position to maximize the throughput. While many localization schemes have been proposed for RFID devices, they either require a number of antennas [17], or are not fast and lightweight enough for channel estimation purposes [18]. Blink [8] uses link signatures to detect mobility, yet it requires measurements from at least 10 channels because RSSIs are the only sources. Such multiple-channel detection introduces too much overhead. To address this issue, we propose a more effective mobility detection using both RSSIs and phases.

The solution is to use phase, a PHY-hint, which is supported in COTS readers as specified in the LLRP standard. For every successful read, the reader outputs a phase reading and an RSSI value, making it virtually zero-overhead. The reported phase is an effective way to measure the distance between the reader and tag, R. The relationship between such distance and measured phase, θ, is as follows [18],

$$\theta = 2\pi\frac{2R}{\lambda} + \theta_D + \theta_R + \theta_M + N\pi,$$

where $\lambda =$ is the wavelength, θ_D, θ_R, θ_M, are phase errors brought by tag and antenna diversity, reflection characteristics, and multipath, respectively, N is the integer ambiguity as the measured phase is with period π. Therefore the distance between two locations is approximated as

$$\Delta R \approx \frac{\lambda}{4\pi}\Delta\theta.$$

To set up a threshold that detects mobility, we conduct an empirical study. From our field experiments, we observe that when the tag is stationary, the phase measurement is highly concentrated. Specifically, the variance is only 2.2°, and the gap between the min value and max value is only 19° (0.33 radians), which only corresponds to 0.8 cm. Therefore, we set up a threshold $\theta_{th} = 0.33$.

Note that to ensure that N is the same for two consecutive phases, the phase rotation between the two should be less than π. This requirement is equal to that when the reading rate is 50 reads/s, it can handle moving objects at velocity up to 4 m/s, which is fairly enough for indoor applications. When the reading rate is below this threshold, it could make false negative alarms. To reduce this alarm, we use RSSIs as a second metric and set its threshold at $\mathrm{RSSI}_{th} = 1$, which is the granularity of RSSIs from COTS readers. Therefore, our mobility detection works as follows. First, we check whether the phase difference is greater than θ_{th}, if so, we label it as a positive location change; otherwise, we check whether the RSSI difference is greater than RSSI_{th}, if so, it is positive, otherwise negative.

Note that environmental mobility, e.g., human/metal objects moving nearby, could be misidentified as location changes because link characteristics, e.g., RSSIs and phases, are easily affected by multipath. In fact, such misidentification is beneficial to our system because it is the channel change that causes misidentification and thus makes probing necessary.

Recently work on Tagwatch [19] introduces a rate-adaptive reading system that first identifies mobile tags and then exclusively read those tags. While both this work and ours have mobility detection modules, there are at least two fundamental differences. First, Tagwatch aims to improve reading rates at the application layer, e.g., tracking, while we intend to boost reading rates at the link layer. According to the popular OSI model for networking [20], the link layer, which lies between the physical and MAC layers, emphasizes adapting physical lay modulation parameters to varying channels. We follow the same principle and try to find the best modulation (Tari values and encoding schemes) for both the downlink and uplink. Nevertheless, Tagwatch aims at an upper application layer and does not investigate physical layer parameters. Second, Tagwatch's mobility detection works at a coarse time scale; it detects mobility at the scale of "3-5s" (See Section 7 of [19]). Time scale here means how fast a mobility detection module responds. On the contrary, ours works at the order of tens of microsecond. Note that such a time-scale difference does not come from the reading rate, but the solutions and design goals. In particular, Tagwatch employs a GMM model to approximate mobility for tracking purposes where time-requirement is not too stringent while RAB needs a faster approaches that can choose the best rate for the physical layer, which requires to work within channel coherence time, e.g., 100 ms. In short, RAB and Tagwatch perform mobility detection for different purposes and working at different network layers; thus they are complementary to each other and can work together to bring better system performance.

Knowing the motion statuses of backscatter nodes can further help rate adaptation. For example, if real-time location information is available, the reader may tend to choose lower rates when the node is far away. The moving direction and speed of nodes are also very helpful when the reader wants to know if the node is moving towards or away from some "dead zones" where the slowest rate should be adopted to combat severe channel conditions. Yet, deriving such motion information accurately and timely is a challenge and most existing solutions on the application layer are too heavy for rate adaptation on the link layer. For example, Tagoram

TABLE I
THROUGHPUT COMPARISON OF WITH AND WITHOUT INTERFERENCE AT P1 THAT IS 30 cm FROM THE INTERFERING SOURCE

	(a) FM0	(a) M2	(a) M4	(a) M8	(b) FM0	(b) M2	(b) M4	(b) M8
Tari6.25	240	228	226	123	50	46	38	20
Tari12.5	231	215	202	116	41	37	26	16
Tari25	186	169	147	104	34	29	18	13

(a) w/o interference, RSSI=-50 dBm (b) w/ interference, RSSI=-68 dBm

TABLE II
THROUGHPUT COMPARISON OF WITH AND WITHOUT INTERFERENCE AT P2 THAT IS 5 m FROM THE INTERFERING SOURCE

	(a) FM0	(a) M2	(a) M4	(a) M8	(b) FM0	(b) M2	(b) M4	(b) M8
Tari6.25	37	60	152	79	10	26	27	29
Tari12.5	42	116	134	91	8	16	53	33
Tari25	31	32	105	92	7	14	36	28

(a) w/o interference, RSSI=-60 dBm (b) w/ interference, RSSI=-70 dBm

TABLE III
APPLYING THE LEARNED MAP TO DIFFERENT SCENARIOS ACROSS TIME AND PLACES

	Accuracy (%)	relative to optimal throughput (%)
Testbed - 1st day	93.4	96.4
Testbed - 2nd day	94.5	98.1
Testbed - 3rd day	92.5	93.1
Classroom	83.2	90.2
Library	76.5	86.3
Lounge	77.9	85.7
Tennis Field	70	81.2

can achieve centimeter-level accuracy but requires too much computation overhead and long delays [18].

B. Channel Hopping

Our second trigger is based on channel hopping, which is mandatory as defined in the C1G2 protocol that the reader can only stay on a channel in a time window. The quality of channel may change due to hopping so that it is the chance the reader needs to adapt rates. Prior work, such as *selection* in [8], needs to probe all the channels to choose top ones, incurring substantial unnecessary overhead. In contrast, our hopping scheme is inspired by CARA [9], which is to use channel correlation to largely reduce probing overhead. As we share the same observation with CARA that good or bad channels are clustered by channel indexes, the main difference is how we fit this idea into our probing framework. Specifically, when the current channel is good, we choose to probe the next channel that is within h_g-hop of the current one; if the probed channel one is good, we stay, otherwise, we will switch to another one that is far away from the probed one, say h_b-hop distance. The channel gap is empirically set at $h_g = 3$ and $h_b = 5$. To decide a channel is good or bad, we use a very conservative threshold 5 reads/s. The rationale of this setting is the observation that the transition between high and low loss rates is sharp [11].

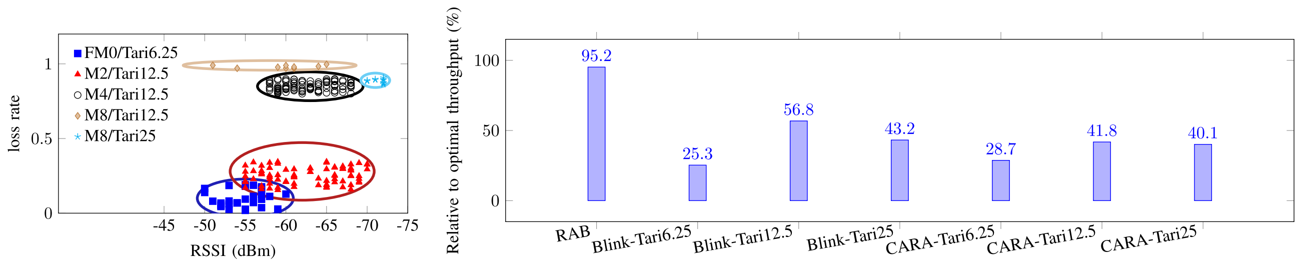
Note that the *fast switching* method [8] may seem similar to ours at first glance. First, the choice of the next channel in [8] is random and thus non-directional, whereas our hopping direction is guided by the channel correlation. Second, it needs to measure burstiness of the channel, incurring on extra burdens.

VII. COUNTERMEASURES AGAINST INTERFERENCE

Previously, we assume no interference for rate adaptation; however, the channel quality is susceptible to wireless interference, distorting our rate mapping relationship. The interference sources could be other unscheduled RFID readers and many other wireless devices operating on the 900 MHz band, which has a very narrow bandwidth and harbors both amateur and ISM frequencies. Typical devices using 900 MHz include wireless LAN point-to-point bridges, remote control of broadcast televisions, baby monitors, cordless phones, hobbyist radios, two-way radio talkie, etc. We perform controlled experiments to examine how interference impacts our rate mapping. As shown in Table II, when the interference source is 30 cm away, the maximal throughput drops from 240 reads/s without interference to 50 reads/s with interference. Note that the optimal rate remains the same in this case. Nevertheless, this may not hold for other scenarios. We move the interference source 5 m away as shown in Table III and observe that the optimal rate changes from M4/Tari6.25 to M4/Tari12.5. Therefore, we conclude that while interference can significantly degrade the throughput, the optimal rate changes indefinitely, which motivates us to design countermeasures against interference.

First, we need to detect the existence of interference. Our detection scheme is based on the observation that interference distorts the rate mapping relationship. Hence, as shown in Figure 7a, if the measured (RSSI, loss rate) pair falls out of all classes, we classify it as the interfered pair. The rationale for this is two-fold. When there is interference, it usually requires higher RSSIs to achieve the same loss rate or the loss rate increases for the same RSSI, both resulting in pairs outside classes.

After interference detection, if no interference is found, we perform rate mapping as described in the previous sections. If some interfered source is spotted, we employ throughput-based probing to choose the best rate. Because how the best rate may change under interferences is uncertain, our throughput-based probing is essentially exhaustive search. Our throughput-based probing starts with the current best rate. When there are four successive failures, we probe the next level. Also, we have a timer, called probing interval. If the rate has been staying at the same level for a probing interval, there is a forced-probe for the next higher rate. As such,



(a) Empirically rate map.

(b) Impact of downlink rates with different schemes

Fig. 7. We learn an empirical rate map from over 200 samples as in (a), which can be used to guide the rate selection for measured RSSI and loss-rate pairs; then we compare RAB's rate selection against BLINK and CARA, showing that RAB has significant improvement thanks to the optimal rate selection of downlink rates.

Algorithm 1 Countermeasures Against Interference

```
Run the interference detection module
if interference detected then
    state = interfered
    while the probing interval ends do
        throughput-based probing for the best rate
    end while
    probe the next higher rate
else
    state = interference-free
    Rate mapping
end if
```

the rate wouldn't be trapped at a low rate. The pseudo-code of interference countermeasure for rate adaptation is included in Algorithm 1.

VIII. Implementation

In this section, we present how we conduct evaluation.

Reader: We mainly use a Thingmagic M6e reader for implementation, which is fully compatible with the C1G2 protocol. Same as [8], the COTS reader has three limitations due to API constraints: First, the data rate can only be set up at the beginning of a query round; Second, the channel switching is not lightweight and takes about 30 ms; Third, the minimum probing time is 30 ms. We hope these factors will be addressed in future readers. Currently, we only use trace-driven studies to examine the aspects that are bounded by the above limitations, such as channel switching.

Tag: Although we have tested many tags from different vendors, such as Impinj, NXP, we do not observe significant performance differences. Thus we choose a representative, the Alien Higgs 3 tag, AZ-9640. One of the main reasons that we extensively use this tag is that it has the largest user memory, which is 512 bits, among tags in the same price range. As the content of sensor data does not affect our protocol at all, we write 512 random bits into the user memory of each test tag in advance.

Parameter: The Thingmagic M6e provides two BLF options, 640 kHz and 250 kHz, but only FM0 and Tari 6.25 are allowed with 640 kHz. Thus we mainly use 250 kHz for BLF on this reader, which allows Tari 6.25, 12.5, 25 and FM0/M2/4/8 on this frequency. For probing, we set up $Q = 1$ to avoid MAC collisions and a filter of which the memory bank is EPC, the starting address is 32, the length is 96, and the mask is the target tag's ID. The rates of probing packet are fixed at the slowest: M8 and Tari 25. The reader power is fixed at 30 dBm.

Competition: We compare RAB with two state-of-the-art schemes, Blink [8] and CARA [9]. To ensure a fair competition, rate adaptation schemes from other wireless networks, e.g., SampleRate [21], are not included as no clear standards or publications have specified how to adapt them to backscatter networks, because a backscatter link is two-way not one-way for other wireless networks.

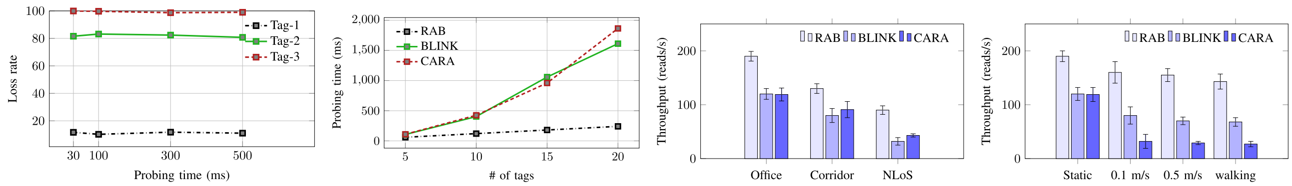
Default Experimental Settings: By default, the tag no. is 5; the backscatter link frequency is 250 kHz; the Q of the ALOHA protocol is 1 for selective probing; each reading period is 3 seconds; the length of tag data is 512 bits.

Due to the design of the C1G2 protocol, every tag-data reading needs to transmit EPC (96 bits) first, which introduces unnecessary delay and affects the overall reading performance. To avoid such a limitation, we evaluate our method using reads/s instead of the amount of tag-data traffic. To make our proposal more suitable for transferring bulk tag-data e.g., sound and images, one of the most important future work include designing burst read modes that can build a connection first and then enter into flow-based transmission without reidentification. Such a design may need to take the MAC-layer redesign into consideration as well due to the fairness concern.

IX. Evaluation

A. Rate Selection

To begin with, we investigate how our rate selection scheme works. As Figure 5 only shows the intuition how rates would adapt to different locations, the actual boundaries of different classes could be irregular. Figure 7a is the empirical rate map we learn from 230 randomly sampled locations in our testbed of size 4m×5m. At each location, we measure all possible combinations of downlink and uplink rates. As expected, we observe that not every class is on the map and the boundaries are not regular. In addition, the trend of different classes does go with our prediction that when the RSSI decreases, the lower throughput of the downlink is favored; when the loss rate increases, a slower encoding scheme should be used.



(a) Loss rate VS probing time

(b) Comparison of probing overhead with for RAB, Blink, and CARA.

(c) Impact of probing on throughputs of different places

(d) Impact of probing on throughputs of static and mobile tags

Fig. 8. We examine our probing scheme in detail. (a) shows that a time interval of 30 ms is enough to accurately estimate loss rates; (b) shows that the probing costs of Blink and CARA are way larger than that of RAB; (c) demonstrates that RAB achieves better throughput for various scenarios; (d) shows our lightweight probing benefits the throughput in both static and mobile scenarios.

Note that our classifier has some errors. For example, some points of FM0/Tari6.25 and M2/Tari12.5 are mixed, because the throughput of both is similar.

To further check the impact of downlink rates, we compare it with Blink and CARA. Since both Blink and CARA do not consider the downlink rate, we make three variants for them, each of which has a distinct Tari. The results are plotted in Figure 7b. Not surprisingly RAB outperforms all the variants of Blink and CARA because a single fixed Tari cannot bring too much gain across different location and channel conditions. One interesting thing to note is that the fastest downlink rate, Tari 6.25, performs even worse than other Tari values. It is mainly because that the too aggressive rate hurts the downlink and makes uplink and overall throughput suffered.

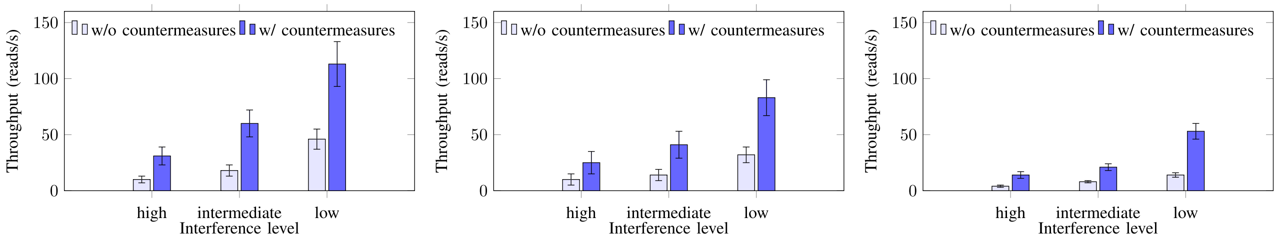
To verify the effectiveness of our rate map, we apply it to various scenarios that are with different dates and places. The results are shown in Table III. First, we test this rate map for three consecutive days in our testbed and obtain testing data of 200 samples for each day. We achieve more than 90% rate selection accuracy and more than 90% of the optimal throughput for three days, which shows the robustness of our scheme against time. Then, we apply the map at three different places including classroom, library, and lounge. The rate selection accuracy decreases a bit due to the different background of the place, yet the achieved throughput is still more than 85% of the optimal one. This is because the boundary errors in the empirical rate map make the rate selection accuracy degraded, but the similar performance of boundary points keeps the overall throughput not affected too much.

B. Probing Cost

Next, we examine the impact of our probing scheme. First, we need to determine how long should we probe. Figure 8a shows the probing results across different time intervals for 3 different tags. We observe that the accuracy of probing is not sensitive to the time interval for low and high loss rates. Therefore, we set the probing interval at 30 ms. Note that 30 ms is the minimal time window that is allowed on COTS readers.

Furthermore, we compare our probing cost against Blink and CARA with different tag populations. To avoid the negative effect of 30 ms minimal window that severely degrades the probing performance of Blink and CARA, this comparison is done with traces. Figure 8b demonstrates that the probing cost of Blink and CARA grows quadratically with the number of tags while that of RAB increases linearly. Specifically, the probing costs of Blink and CARA are 1612 ms and 1864 ms, corresponding to 6.7× and 7.8× more than that of RAB when there are 20 tags. This is primarily due to the filter-based probing paradigm that probes tags sequentially while Blink and CARA need more time to deal with MAC collisions.

In addition, to investigate how our probing benefits the overall throughput, we compare RAB against state-of-the-art schemes under complex scenarios. First, we examine how RAB performs under different multipath environments, including offices with normal multipath, corridors with severe multipath, and NLoS scenarios with tags obstructed by a wooden door. To eliminate the impact of MAC collisions and channel hopping, we only use one tag and one channel here. From Figure 8c, we observe that RAB is significantly better than Blink and CARA in all three cases. This is mainly because it uses a rate mapping scheme to select the best rate for both the downlink and uplink, while previous systems only rely on the uplink. As expected, RAB and other systems experience throughput drop when multipath becomes more severe. Nevertheless, RAB's degradation is less than Blink and CARA, showing strong resilience to multipath. Next, we intend to examine how RAB behaves under dynamic channel conditions. In particular, to examine the performance under different speeds, we employ an iRobot Create programmable robot, which has two powered wheels and a third passive caster wheel maintains balance. According to the official SDK, the maximum velocity can be set is 0.5 m/s. So we conduct tests under static, 0.1 and 0.5 m/s for the iRobot, and a person attached with a tag of around 0.5 m/s scenarios. Figure 8d shows that the throughput of RAB is considerably better than those of Blink and CARA for both static and mobile scenarios. Furthermore, while there is no much difference between Blink and CARA in the static setting, CARA suffers more degradation than Blink does in the mobile scenario because CARA is not mobility-aware. Besides, RAB's performance is quit stable across different speeds, which can be attributed to its mobility-awareness. In addition, the case of a person with a tag performs slightly worse than the iRobot at the same speed. It is primarily because a human usually absorbs more RF energy than the robot, leading to lower backscattered signal strength.



(a) 5 tags (b) 10 tags (c) 20 tags

Fig. 9. Impact of interferences and countermeasures for different tag populations.

TABLE IV

QUERY ESTIMATION ACROSS DIFFERENT RATES. WE OBSERVE THAT THE RELATIVE ERRORS OF QUERY ESTIMATION FOR RAB ARE WITHIN 5%

	query measured	query predicted	relative error (%)
FM0/Tari6.25	248.8	258.3	3.8
Miller2/Tari6.25	244.6	256.4	4.8
Miller4/Tari6.25	235.7	224.4	4.7
Miller8/Tari6.25	127.1	130.9	3
FM0/Tari12.5	246.2	255.8	3.9
Miller2/Tari12.5	245.1	241.5	1.5
Miller4/Tari12.5	209.8	214.4	2.2
Miller8/Tari12.5	122.0	123.8	1.4
FM0/Tari25	244.6	233.8	4.4
Miller2/Tari25	243.8	241.1	1.1
Miller4/Tari25	175.6	182.9	4.1
Miller8/Tari25	106.3	105.6	0.6

C. Loss Rate Estimation

Now we look to check link timing based loss rate estimation. As the number of successful reads is known from the reader output, we only need to examine the accuracy of query estimation. For the ground truth, we use a USRP-based monitor at a very close distance, 10 cm, to capture messages between the reader and the tag. The results in Table IV show that our estimation achieves less than 5% errors all the time and thus are quite robust across a range of different rate settings. Such errors do not affect the rate selection as shown in Figure 7a. Note that while prior methods can also obtain loss-rate estimates, they require either a USRP monitor or CRFID tags [13]. In contrast, our method is accurate and does not need any extra hardware because we make use of the link timing feature of backscatter communication.

D. Interference Countermeasures

To investigate effectiveness of our interference countermeasures and co-existence with other wireless devices, we first examine the interference detection accuracy and the impact on the single-tag throughput. We test three different interfering sources: ImpinJ reader R420, Amateur radio Alinco DJ-G29T, and WLAN bridge Nanobridge NBM9. As shown in Table V, our detection accuracy are all above 75% for interference strengths ranging from 33.9 to 0 dBm for all tested devices.

TABLE V

INTERFERENCE DETECTION ACCURACY AND IMPACT ON THROUGHPUT. WE OBSERVE THAT THE STRONGER INTERFERENCE, THE LOWER THROUGHPUT FOR DIFFERENT INTERFERENCE SOURCES

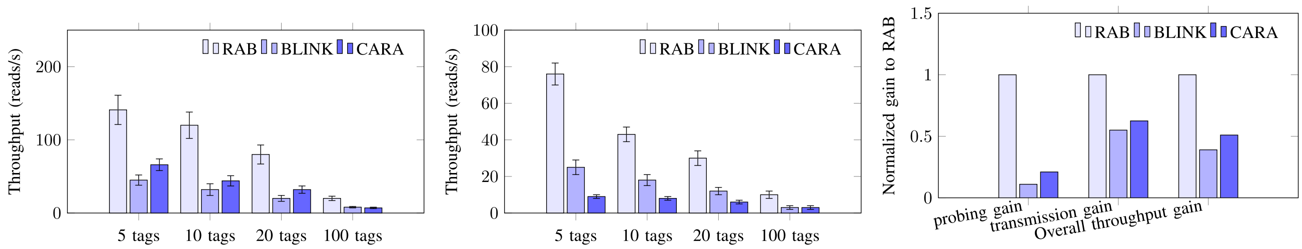
Interference strength	Detection ratio (%)	Single-tag throughput
30 dBm (ImpinJ reader)	92.3	37 ± 3
20 dBm (ImpinJ reader)	87.5	83 ± 5
10 dBm (ImpinJ reader)	81.9	121 ± 8
0 dBm (ImpinJ reader)	78.1	167 ± 10
33.9 dBm (Amateur radio)	93.7	38 ± 2
28 dBm (WLAN Bridge)	93.1	40 ± 4

Particularly, for the ImpinJ reader, when the interference strength is 30 dBm, the accuracy is as high as 92.3%. The detection ratio becomes less accurate as the interference strength is decreasing. The main reason is as the interference level is low, the interference becomes more indistinguishable from normal signals and thus hard to discover. Similarly, when the interfering signal becomes stronger, the single-tag throughput would go lower. For the amateur radio and WLAN bridge, the achieved accuracy is still more than 90%. In conclusion, our interference detection is robust for all those different devices.

After detecting interference, we further conduct a bunch of experiments to compare the performance with and without our countermeasures. As shown in Figure 9, the comparison is done across different levels of interference and different tag populations. With interference countermeasures, high throughput gains are observed for all scenarios. Specifically, the throughput with countermeasures is 2.5× and 3.1× better than cases at high and low level without countermeasures when there are 5 tags. Similar observations can be made when the number of tags increases to 10 and 15, shown in Figure 9b and 9b. Such performance gains are mainly due to our throughput-based probing scheme and the introduction of probing timer.

E. Overall Performance

We now look at the overall performance of the whole framework and compare it with state-of-the-art systems. First, we study the static case where all tags are placed randomly. Figure 10a shows that when there are 5 tags, the throughput of RAB is 3.1× and 2.1× better than Blink and CARA, respectively. The same trend can be observed when the number of tags increases. As expected, all schemes degrade with the



(a) Static scenario (b) Mobile scenario (c) Performance gains

Fig. 10. Overall performance comparison under static and mobile scenarios with different tag populations.

increasing number of tags because of more coordination time needed.

When it turns to the mobile case in Figure 10b, all of the three systems are affected by mobility differently, but RAB is still the best across different tag populations. Particularly, when the number of tags is 20, RAB achieves 2.5× and 5× throughput gains over Blink and CARA. CARA is the worst due to its lack of mobility detection module.

Then we conduct over 120 tests across different mobility, channel, and network-size conditions. Due to equipment constraints, currently we are unable to find a number of robots to carry tags so we invite 20 volunteers. We let each volunteer carry 1 tag when there are less than 20 tags. When the tag population is 100, each one takes 5 tags. For mobility, we ask volunteers to randomly walk no faster than 1 m/s. For channels, we collect the data across 1-week at two difference places. The overall gains and its breakdown on average are reported in Figure 10c. RAB achieves overall throughput gains of 2.6× over Blink and 2× over CARA. We break down this gain and find that RAB reduces probing cost by 8.2× and 4.3× over Blink and CARA. The majority of this probing gain comes from the filter-based probing design as it successfully avoids MAC collisions while being compatible with the C1G2 protocol. Meanwhile, regarding data transmission, RAB is 1.8× and 1.6× better than Blink and CARA. This transmission gain is mainly brought by the downlink-aware rate selection scheme while all prior systems leave the downlink unattended.

X. Related Work

Backscatter Communication Efficiency: Backscatter communication optimizations can be roughly classified into two categories: C1G2-compatible and C1G2-incompatible. Buzz [5] introduces a rateless coding for backscatter nodes, which achieves lossless transmission. Flit [13] designs a new MAC that enables burst transferring bulk data, significantly reducing wasted time by the C1G2 MAC. Laissez-Faire [22] and BiGroup [23] propose to decode parallel transmissions by analyzing signals in the both time and IQ domains, which can work at moderate and high SNR scenarios. Those C1G2-incompatible optimizations achieve substantial performance gain but fall short of accommodating billions of deployed RFID readers and nodes. Some C1G2-compatible improvements have been proposed recently. Blink [8] makes use of unique backscatter link signatures to detect mobility and adapt rates. CARA [9] observes the opportunity that throughput can be improved by channel-aware rate selection. Unlike both that focus on the uplink rate selection, we observe that the downlink rate could greatly affect the overall throughput as well. In addition, our filter-based probing tries to efficiently estimate channels and avoid collision problems that are not well considered before.

Rate Adaptation: Rate adaptation has been widely researched in active-radio based wireless networks, like 802.11. BER [24], SNR [25], [26], and loss rate [27] are the most commonly used metrics. While our work shares the same idea that chooses the optimal rate that maximizes the network throughput by estimating the channel quality. Those methods have limited applicability to backscatter systems, especially for the C1G2 protocol. For example, the limited visibility of current COTS readers makes even loss rates hard to observe. To solve this, we use the link timing features specified by the C1G2 protocol to approximate the loss rate. In addition, we accurately deduce mobility hints using RSSI and phase measurements together.

New Backscatter Paradigms: Several novel backscatter systems where nodes are powered by various sources have been proposed, e.g., WiFi-backscatter [28]–[30], Bluetooth-backscatter [6], FM-backscatter [7]. LoRa backscatter is proposed to significantly increase the backscatter operation range to about 500 m using commodity LoRa hardware [31]. Long-range WiFi-based backscatter communication that is compatible with commodity WiFi device uses code translation to piggyback the sensor data on the ongoing WiFi communication [32], which also extends to Bluetooth and ZigBee [33]. Those systems largely extend the operating range of traditional readers and see a bright future of interconnecting more and more wireless devices. Yet, their interpretability with C1G2 is worth further investigation.

XI. Conclusion and Future Work

We have presented RAB, a protocol that is to optimize throughput within the C1G2 standard from many aspects, including downlink-aware rate selection, filter-based probing, lightweight probing triggers, and robust interference countermeasures. Our prototype has shown that considerably throughput gains have been achieved over state-of-the-art schemes. With more and more backscatter sensors have been invented, we believe RAB can benefit a wide range of Internet-of-Things applications.

References

[1] V. Hsu, J. M. Kahn, and K. S. Pister, "Wireless communications for smart dust," Electron. Res. Lab., College Eng., Univ. California, Oakland, CA, USA, 1998.

[2] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy, and A. Mamishev, "A wirelessly-powered platform for sensing and computation," in *Proc. UbiComp*, 2006, pp. 495–506.

[3] Y. Zhao and J. R. Smith, "A battery-free RFID-based indoor acoustic localization platform," in *Proc. IEEE Int. Conf. (RFID)*, Apr. 2013, pp. 110–117.

[4] S. Naderiparizi, A. N. Parks, Z. Kapetanovic, B. Ransford, and J. R. Smith, "WISPCam: A battery-free RFID camera," in *Proc. IEEE Int. Conf. RFID (RFID)*, Apr. 2015, pp. 166–173.

[5] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2012, pp. 1–12.

[6] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith, "Inter-technology backscatter: Towards Internet connectivity for implanted devices," in *Proc. ACM SIGCOMM*, 2016, pp. 356–369.

[7] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, "FM-backscatter: Enabling connected cities and smart fabrics," in *Proc. USENIX NSDI*, 2017, pp. 243–258.

[8] P. Zhang, J. Gummeson, and D. Ganesan, "Blink: A high throughput link layer for backscatter communication," in *Proc. ACM MobiSys*, 2012, pp. 99–112.

[9] W. Gong, H. Liu, K. Liu, Q. Ma, and Y. Liu, "Exploiting channel diversity for rate adaptation in backscatter communication networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2016, pp. 1–9.

[10] (2017). *EPC C1G2 Standard*. [Online]. Available: http://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1

[11] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2003, pp. 1–13.

[12] (2017). *Low Level Reader Protocol*. [Online]. Available: http://www.gs1.org/epcrfid/epc-rfid-llrp/1-1-0

[13] J. Gummeson, P. Zhang, and D. Ganesan, "Flit: A bulk transmission protocol for RFID-scale sensors," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2012, pp. 71–84.

[14] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[15] K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla, "On synopses for distinct-value estimation under multiset operations," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2007, pp. 199–210.

[16] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2008, pp. 223–234.

[17] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM SIGCOMM*, 2014, pp. 1–4.

[18] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2014, pp. 237–248.

[19] Q. Lin, L. Yang, H. Jia, C. Duan, and Y. Liu, "Revisiting reading rate with mobility: Rate-adaptive reading in COTS RFID systems," in *Proc. 13th Int. Conf. Emerg. Netw. EXp. Technol.*, Nov. 2017, pp. 199–211.

[20] J. F. Kurose, *Computer Networking: A Top-down Approach Featuring the Internet, 3/E*. London, U.K.: Pearson, 2005.

[21] J. C. Bicket, "Bit-rate selection in wireless network," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.

[22] P. Hu, P. Zhang, and D. Ganesan, "Laissez-faire: Fully asymmetric backscatter communication," in *Proc. ACM SIGCOMM*, 2015, pp. 255–267.

[23] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for COTS RFID tags," in *Proc. ACM MobiCom*, 2015, pp. 500–511.

[24] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 3–14, Aug. 2009.

[25] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and MAC protocols," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2009, pp. 193–204.

[26] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, "AccuRate: Constellation based rate estimation in wireless networks," in *Proc. USENIX NSDI*, 2010, pp. 175–190.

[27] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2006, pp. 146–157.

[28] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi backscatter: Internet connectivity for RF-powered devices," in *Proc. ACM Conf. SIGCOMM (SIGCOMM)*, 2014, pp. 607–618.

[29] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive Wi-Fi: Bringing low power to Wi-Fi transmissions," in *Proc. USENIX NSDI*, 2016, pp. 151–164.

[30] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, "BackFi: High throughput WiFi backscatter," in *Proc. ACM SIGCOMM*, 2015, pp. 283–296.

[31] V. Talla, M. Hessar, B. Kellogg, A. Najafi, R. J. Smith, and S. Gollakota, "Lora backscatter: Enabling the vision of ubiquitous connectivity," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 105:1–105:24, Sep. 2017.

[32] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "HitchHike: Practical backscatter using commodity WiFi," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. (CD-ROM)*, Nov. 2016, pp. 259–271.

[33] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "FreeRider: Backscatter communication using commodity radios," in *Proc. 13th Int. Conf. Emerg. Netw. EXp. Technol.*, Nov. 2017, pp. 389–401.

Si Chen (Student Member, IEEE) received the bachelor's degree from the China University of Geosciences and the master's degree from Simon Fraser University, where she is currently pursuing the Ph.D. degree with the School of Computing Science. Her recent research interests include wireless networks and big data.

Wei Gong (Member, IEEE) received the B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, the M.S. degree from the School of Software, Tsinghua University, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University.

His research interests include backscatter communication, distributed computing, and the Internet-of-Things applications.

Jia Zhao (Graduate Student Member, IEEE) received the M.S. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China. He is currently pursuing the Ph.D. degree with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. His research interests include networking, multimedia communications, cloud computing, and transport protocols.

Jiangchuan Liu (Fellow, IEEE) received the B.Eng. degree *(cum laude)* from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is currently a University Professor with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. He is an NSERC E. W. R. Steacie Memorial Fellow. He was a co-recipient of the inaugural Test of Time Paper Award of the IEEE INFOCOM in 2015, the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award in 2013, and the ACM Multimedia Best Paper Award in 2012. He has served on the Editorial Board of the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and the IEEE INTERNET OF THINGS JOURNAL. He is a Steering Committee Member of the IEEE TRANSACTIONS ON MOBILE COMPUTING.