BLINK: A High Throughput Link Layer for Backscatter Communication

Pengyu Zhang, Jeremy Gummeson, Deepak Ganesan
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{pyzhang, gummeson, dganesan}@cs.umass.edu

ABSTRACT

Backscatter communication offers an ultra-low power alternative to active radios in urban sensing deployments — communication is powered by a reader, thereby making it virtually "free". While backscatter communication has largely been used for extremely small amounts of data transfer (e.g. a 12 byte EPC identifier from an RFID tag), sensors need to use backscatter for continuous and high-volume sensor data transfer. To address this need, we describe a novel link layer that exploits unique characteristics of backscatter communication to optimize throughput. Our system offers several optimizations including 1) understanding of multi-path self-interference characteristics and link metrics that capture these characteristics, 2) design of novel mobility-aware probing techniques that use backscatter link signatures to determine when to probe the channel, 3) bitrate selection algorithms that use link metrics to determine the optimal bitrate, and 4) channel selection mechanism that optimize throughput while remaining compliant within FCC regulations. Our results show upto 3× increase in goodput over other mechanisms across a wide range of channel conditions, scales, and mobility scenarios.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design, Experiment, Measurement, Performance

Keywords

Backscatter communication, Mobility detection, Rate adaptation, Channel switching

MobiSys'12, June 25–29, 2012, Low Wood Bay, Lake District, UK.

1. INTRODUCTION

An emerging class of applications require small form-factor, ultra-low power sensors that are ubiquitously deployed in ways presently difficult to achieve with conventional wireless sensor technologies. These devices may be deployed in areas where small form-factor is critical (e.g. on-body sensors for gesture and activity recognition), maintenance is difficult or impossible (e.g. embedded monitoring of bridges and planes), or on a host of small everyday objects (Internet of Things).

An acute problem for this class of sensor devices is operating under extremely low energy budgets. The sizes of batteries or capacitors have to be tiny, necessitating new designs that can operate at orders of magnitude lower power than typical mote-class sensor platforms. While micro-controllers have become increasingly efficient in terms of their active and sleep-mode power consumption, active radios such as 802.15.4 still consume too much power. There is therefore a dire need for an ultra-low power communication mechanism for this class of sensors.

One paradigm that promises to dramatically reduce energy cost is backscatter communication. A backscatter basestation provides a carrier wave to the sensor node, which can decode these transmissions using a simple analog comparator circuit. To transmit data, a sensor toggles the state of a transistor to detune its antenna and "reflects" the carrier wave back to the reader with its own information bits. Since a sensor does not need to actively transmit a radio signal, this can lead to extremely low-power communication. In addition, backscatter requires low-complexity circuits which can be implemented at low-cost. Despite these benefits, widespread use of backscatter communication for sensors has been stymied by two drawbacks: a) it is restricted to short-range, single-hop communication, and b) RFID readers are bulky and difficult to deploy in a dense manner.

Recent developments, however, have mitigated these drawbacks and renewed the case for wider use of backscatter communication on sensors. The first development is dramatic increases in communication range by enabling tags to take advantage of alternate energy sources, for example tiny solar panels [15, 5]. The second development is miniaturization in RFID reader hardware leading to small portable readers that can be attached to a mobile phone [2]. Thus, backscatter communication offers a considerably more energy-efficient and increasingly practical alternative to active radio circuits on existing sensor systems.

Despite the possibilities, we have limited understanding of how a backscatter link layer should be optimized for data

transfer from sensor devices. Our focus in this work is on two central elements in such a link layer — bit-rate adaptation and channel selection. While these topics have been widely studied in systems that employ active radios (e.g. 802.11 [9, 21, 23, 27, 28] and 802.15.4 [24]), the applicability of prior results to our problem is limited. Unlike a conventional active radio link, a backscatter link comprises both the forward link from reader to sensor and the backward link from the sensor to reader, since the sensor is essentially "reflecting" the carrier wave from the reader while modulating with data. This difference results in different path loss for the forward vs backward link, and unique multipath self-interference behavior. In addition, the highly asymmetric hardware capabilities of the reader and the sensor tag imply that modulation and coding schemes for the forward and reverse link are very different.

In this paper, we explore the design of a high-throughput **link layer** for sensors that use backscatter communication. Our work provides novel insights into self-interference in backscatter communication and shows that a combination of RSSI and packetloss metrics are required to capture link characteristics. We use these metrics to design a novel link layer with three fundamental contributions. First, we design a mobility-aware link layer that reduces channel probing overhead by comparing link signatures over time to detect sensor mobility. Second, we present a classifier that uses RSSI and packetloss information to select the optimal bitrate for data transfer. Third, we design channel selection mechanism that optimizes the use of channels for communication while remaining within FCC specifications. Our results show that:

- We achieve less than 10% false positive rate and less than 1% false negative rate for detecting sensor tag mobility across a range of bitrates.
- Our bitrate classifier selects the optimal bitrate with over 80% accuracy under most conditions, and achieves over 88% of the optimal goodput even under heavy multipath scenarios.
- We show that channel selection and switching improve goodput by over 2× in comparison with default schemes used by readers.
- BLINK scales well to large numbers of static and mobile tags. In a deployment with 20 static tags, BLINK is 1.6× better than the default scheme used by Impinj readers (AutoSet) and 1.3× better than a backscatter-optimized version of SampleRate. In a deployment with 10 mobile tags, BLINK is 2.4× better than AutoSet and 2× better than SampleRate.

2. BACKGROUND

This section provides an overview of the physical layer defined by EPC Gen 2, a dominant standard for backscatter based communication. Our focus is on mechanisms that are applicable to commercial readers, since they are readily available and in widespread use.

2.1 RF Harvesting

Backscatter communication is designed to both provide power to a sensor tag as well as to enable communication. As shown in Figure 1, the RFID reader provides a carrier wave, which can be rectified by the sensor to produce DC voltage. This voltage is boosted to an appropriate level by a charge pump at the sensor, and accumulated in a small storage capacitor until the voltage reaches an appropriate threshold before any computation (or sensing) can begin. Once voltage is sufficient to power the device, it can begin to receive and transmit data, both of which are done by modulating the same carrier wave. The RF harvester at the sensor includes an analog comparator circuit that can decode reader transmissions from the reader. To transmit data, a sensor toggles the state of a transistor to detune its antenna and reflect the carrier wave back to the reader with its own information bits. The Intel WISP [6] and UMass Moo [1] are examples of sensor platforms that rely on backscatter communication.

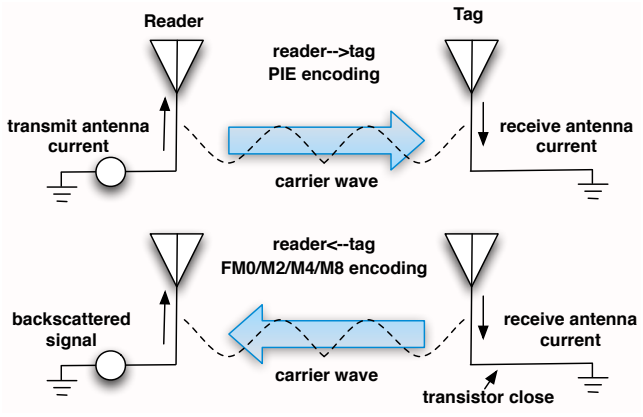


Figure 1: Backscatter signaling at PHY [3]

2.2 Forward vs Backward link

The forward and backward links in backscatter communication differ in several ways. First, the path loss is very different for the two links. The signal to noise ratio (SNR) for typical backscatter communication decays with the square of distance for the forward link and to the fourth power of distance for the backscatter link. Second, the encoding schemes for the links are different — reader to sensor communications use pulse-interval encoding (PIE), which allows easy decoding, whereas sensor to reader communication uses more complex encodings (FM0, Miller2, Miller4, Miller8). Third, the antenna sensitivity at the sensor and reader are vastly different. A typical RFID reader (e.g. Impinj [4]) uses a mono-static antenna for sending and receiving data, which has a sensitivity of -80 dBm. In contrast, an RFID-scale sensor (e.g. the Intel WISP [6]) uses a simple dipole antenna for data transfer, which is significantly less sensitive than the reader antenna. These factors contribute to different link qualities in the two directions. The forward link uses weaker encoding and is received by a less sensitive antenna, but has lower path loss. The backward link uses stronger encoding and is received by a highly sensitive antenna, but has much higher path loss.

Our focus in this paper is on optimizing the backward link from the sensor tag to the reader. We make this decision for two reasons: 1) the forward link offers almost no choices - only one encoding scheme (PIE) is supported and no baud rate option is clearly proposed, whereas sensor to reader

communication uses more complex encodings (FM0, Miller2, Miller4, Miller8) and several baud rates (32 kbps to 640 kbps) since the reader has more computational resources for decoding, and 2) the backward link is crucial since it has greater path loss and unique multipath self-interference behavior, and is therefore very sensitive to sensor placement relative to the reader (discussed in greater detail in §4.1).

2.3 Backscatter Channels

RFID readers use frequency hopping to avoid interference across readers when reading sensors in the same area. A typical UHF reader hops between 50 channels in the 902MHz $\sim$ 928MHz ISM band. FCC regulations specify that a reader can have a maximum channel dwell time of 0.4 seconds in any ten second period to reduce interference in a channel [11]. Commercial reader implementations address this by spending an equal amount of time in each of the 50 channels (0.2 secs) and hopping in sequence from the first to last channel. As we show in §4.4, this is inefficient, and more intelligent choice of channels while remaining within FCC specifications can improve throughput.

3. SYSTEM OVERVIEW

Figure 2 shows the BLINK link layer architecture. At the core of BLINK are link metrics that capture path loss and multipath fading characteristics of a backscatter link. We show that backscatter communication has unique characteristics: RSSI is a better measure of path loss and packet loss rates are a better measure of multipath fading. These metrics are used by a mobility detector that detects changes in the mobility patterns of a sensor tag and triggers bitrate and channel adaptation. The central idea of the mobility detector is to compare link signatures over time to identify whether a sensor has moved from one location to another, or whether it is in continuous motion.

When mobility is detected, the rate adaptation module needs to choose a new bitrate that best suits the channel characteristics. This is done by using a classifier that maps from the link metrics (RSSI and packetloss vectors) to the appropriate bitrate. An additional challenge in rate adaptation is that the link metrics need to be obtained at the lowest bitrate to ensure that sensors are not missed, but obtaining these metrics across 50 channels is exceedingly slow and reduces goodput. The rate adaptation module therefore uses a fast probe technique that exploits loss patterns on backscatter channels, as well as knowledge of the feature needed for the classifier to optimize probing duration.

Mobility also triggers channel selection; this module is responsible for selecting channels that maximize throughput. We take advantage of flexibility in FCC regulations that allow the use of fewer than 50 channels while ensuring per-channel dwell times are under 0.4s. This module measures channel characteristics such as the burstiness and sharpness of transitions between good to bad channels, and uses this information to decide whether to select the best channels a priori, or whether to dynamically switch between them to take advantage of bursts of good throughput.

4. BLINK DESIGN

This section describes the key design elements of BLINK. We first present the link metrics that are the foundation of our link layer, and then discuss how each module in our system takes advantage of the link metric to detect mobility, select optimal bitrate, and select the best channels to use.

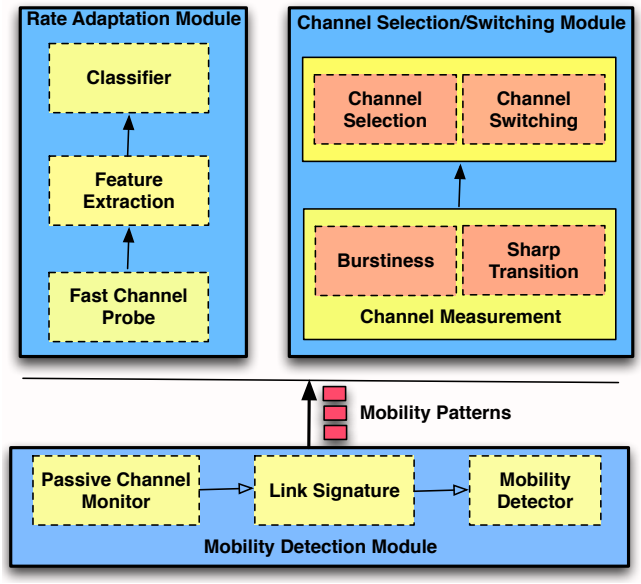


Figure 2: System overview.

4.1 Backscatter Link Metrics

Commercial RFID readers expose two link metrics: a) the RSSI value for each query response from a sensor tag, and b) the aggregate per-channel loss rate for each dwell time interval. In a traditional wireless networks that use active radios such as 802.11 or 802.15.4, RSSI and loss rate are strongly correlated. As a consequence, most link-layer metrics rely on either fine-grained RSSI information or coarse-grained packet loss rates (e.g. [21, 23]). However, a unique feature of backscatter communication is that packet loss and RSSI provide complementary information about path-loss and self-interference, and therefore need to be used in conjunction. To understand why, let us look at the nature of multipath interference in backscatter systems.

Backscatter multipath: Multi-path phenomena is often caused by the transmitted signal being reflected by objects like walls and buildings. As a result, the receiver will detect multiple copies of the same signal which traverse different paths. The summation of these signals distorts the original shape of transmitted signal resulting in pockets of constructive interference where RSSI is high and destructive interference where RSSI is low. In addition, the summation of signals is sensitive to frequency and leads to frequency selective signal distortion.

Multipath effects are a bit different for backscatter communication. Consider the case shown in Figure 3 where the reader sends out a signal S1 and the sensor tag backscatters with signal S2 on the same carrier wave. In addition to the backscattered signal from the sensor, a wall-reflected multipath signal, $S1^{'}$, is also received at the reader. Thus, the backscatter signal received by the reader is the summation of two signals: wall-reflected multipath signal $S1^{'}$ and sensor-backscattered signal S2. $S1^{'}$ could be seen as carrier wave modulated by information from the reader and S2 is carrier wave modulated by information from the sensor. As a re-

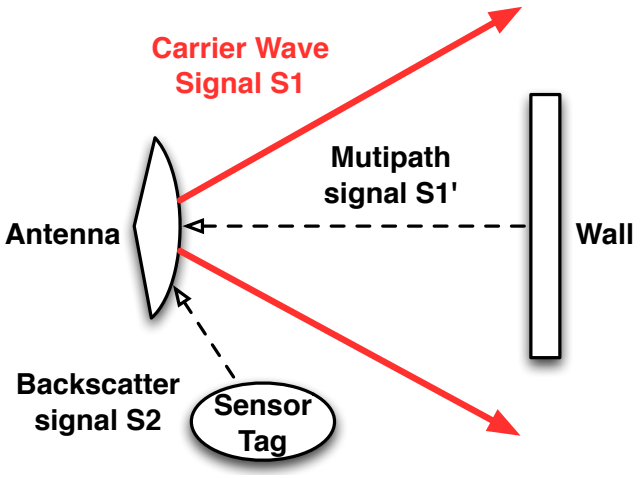


Figure 3: Multipath self-interference when tag is placed at the edge of reader beamformer direction.

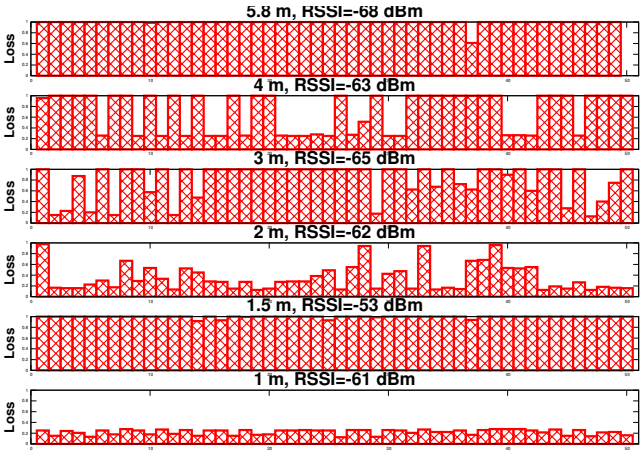


Figure 4: Empirical results about multipath self-interference.

sult, even if constructive interference happens between the two signals , it may not be decoded correctly at the reader since $S1^{'}$ and S2 have different information bits. In other words, RSSI may be high due to constructive interference but packet loss would be high as well.

To contrast against a traditional communication scenario, assume that the sensor tag in Figure 3 is equipped with an active radio. In this case, its transmission signal, S2, to the base-station might interfere with a multipath version of the signal $S2^{'}$ caused by reflection from the wall. Unlike backscatter multipath, copies of the same signal (S2 and $S2^{'}$) interfere with each other, and techniques such as equalization can handle the offset between the signals.

This phenomena is exacerbated by RFID placement relative to the reader's antenna beamformer. When a sensor tag is placed within the beamformer direction of the reader antenna, then the signal strength of the backscattered signal (S2) is typically high enough that it can be decoded despite multipath effects. However, if the sensor is placed outside the primary beamformer direction, then the signal that it receives is weak and the backscattered signal (S2) may not be strong enough to overcome the interference from $S1^{'}$.

Empirical evidence: To provide empirical evidence of the above behavior, we use a passive tag to measure the packet loss rate on each channel for a tag placed at the edge of the antenna beamformer. Figure 4 shows RSSI and corresponding packet loss rate across the 50 channels when the tag is placed at different locations. When the tag is 1 meter from reader, it is within the beam of the reader antenna. As a result, the reader obtains high RSSI and low loss rate. Moving the tag to 1.5m results in a sharp improvement in RSSI, but packet losses become severe. This point clearly shows the effect of multi-path self interference — RSSI is high, which would suggest excellent channel quality, but packet loss rate is high as well due to multipath. The multipath effect reduces when the tag moves a bit more to 2m. Beyond this point, we do not observe self-interference and link quality degrades predictably with distance.

Implication: Why does this distinction between RSSI and packet loss matter? The result shows that we need to consider both RSSI and packet loss to optimize communication on backscatter channels. Together, they capture the effect of path loss and multipath self-interference.

4.2 Mobility Detection

The mobility detection module generates triggers whenever it believes that the sensor tag's location or mobility pattern has changed. Detecting the mobility pattern of a sensor is important for a reader to decide how to interact with the sensor. When a sensor has moved, a reader may need to change the encoding, baudrate, or channels to maximize throughput [22]. In this section, we introduce a zero-overhead approach to determine mobility behavior of sensor tags.

At the core of mobility detection is the notion of a backscatter link signature. The link signature is defined as the distance between the RSSI vectors and lossrate vectors across successive scans of all the channels. Here, the RSSI vector comprises a vector of the RSSI values on each of the N channels that an RFID reader uses to communicate with a sensor, and the lossrate vector is the loss rate across the same channels. We denote the RSSI vector for location A to be $(a_1, a_2, ..., a_N)$ and the packet loss vector to be $(a_1^*, a_2^*, ..., a_N^*)$

We use a simple euclidean distance metric to measure the distance between the vectors for successive scans. The distance d between successive RSSI vectors, $(a_1, a_2, ..., a_N)$ and $(b_1, b_2, ..., b_N)$ is defined as:

$$d = \frac{\sqrt{\sum_{i=1}^{N}(a_i - b_i)^2}}{N}$$

The distance between two packet loss vectors d^* can also be calculated in the above manner. Given the two distances, the mobility detector identifies if a sensor is static or mobile as follows:

$$\text{Output} = \begin{cases} \text{Static} & \text{if } d < d_T \quad \& \quad d^* < d_T^* \\ \text{Mobile} & \text{if } d \geq d_T \quad \| \quad d^* \geq d_T^* \end{cases}$$

where d_T and d_T^* are empirically measured thresholds (see §5). The rationale for generating mobility triggers when either distance was larger than the threshold was because

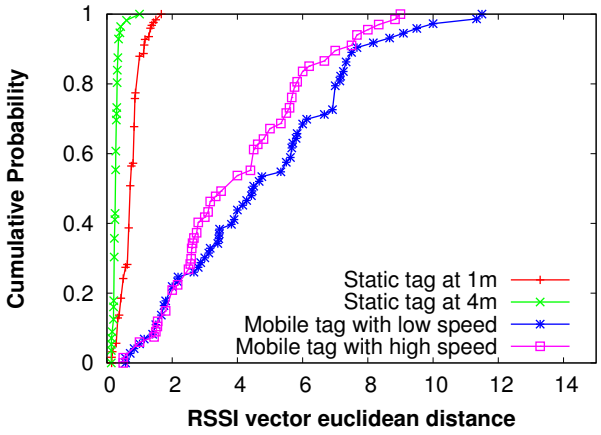


Figure 5: RSSI euclidean distance when tag is stationary and mobile

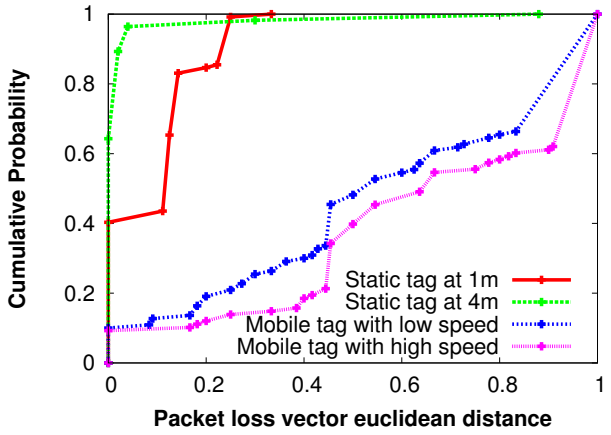


Figure 6: Packet loss euclidean distance when tag is stationary and mobile

we were willing to sacrifice a few false alarms, but wanted to ensure that mobility scenarios were not missed.

Note that changes in the environment such as movement of metal objects in the vicinity of the sensor could be misinterpreted by the mobility detector as a change in location of the sensor since the link signature might have changed. These triggers are, in fact, useful since environmental changes can impact link characteristics and therefore require adjusting communication parameters between the reader and the sensor.

Empirical evidence: To provide an empirical comparison of link signatures for stationary and mobile sensor tags, we use a passive tag to measure link signatures in two cases. The first is when the tag is stationary and placed at two spots that are at different distances from the reader. We use an Impinj reader and the fastest bitrate, FM0/640, for capturing link signatures. The second case is when a tag is mobile at different speeds. To ensure repeatable mobility experiments, we use a LEGO toy train on an oval track which was mounted with a tag. The train was run at different speeds to measure the link signatures under mobility. In each of these experiments, we get a sequence of link signatures, one for each consecutive pair of reader scans.

The distance between RSSI vectors are shown in Figure 5 and the distance between packet loss vectors are shown in Figure 6. The graphs clearly show that it is straightforward to distinguish between the mobile and stationary case since they have vastly different signatures. Even while considering the cases within one of these groups, there is sufficient difference in the signature that we can detect changes in location, or mobility speeds. In §5, we provide a more in-depth breakdown to show that this is indeed possible.

4.3 Rate adaptation

The rate adaptation module exploits the link metrics (RSSI and packet loss vectors) that it obtains from the probes to determine the bitrate that would achieve highest goodput for communication with a reader. As shown in Table 1, an EPC Gen 2 reader specifies six bitrates that are a result of different encoding/baudrate combinations. Our goal is to select the best of these options.

Bitrate (symbol/s)	Throughput (kbps)
FM0/640	640
FM0/160	160
Miller4/640	160
Miller4/256	64
FM0/40	40
Miller8/256	32

Table 1: Ranked encoding and baud rate combinations based on throughput.

4.3.1 RSSI/Packet loss → Encoding/Baudrate

RSSI and packet loss rate are two dominant factors that affect the performance of backscatter system. This intuition for how these link metrics impact bitrate is shown in the RSSI-lossrate map in Figure 7. Consider the first row where RSSI goes from high to low and packet loss is low. As we showed earlier, RSSI in backscatter communication is primarily impacted by pathloss and not multipath effects. Thus, as RSSI reduces, the maximum throughput of the channel reduces and a bitrate that leads to a lower throughout should be chosen. The optimal choice of bitrate follows the order shown in Table 1 i.e. order of reducing throughput.

Now lets look at the case where packet loss goes from good to bad and RSSI is fixed at high. We know that packet loss rate shows the effect of multipath self-interference. Therefore, for a fixed RSSI value, increase in loss rate should lead to a choice of more advanced encoding schemes. Using a more advanced encoding would mean that more symbols are employed to transmit the same information, which in turn would mean more tolerance for signal distortion in the time domain. This would reduce the negative impact of multipath interference. This is shown in the first column in Figure 7 — loss rates increase, the choice of encoding shifts from FM0, which employs one high-low pulse to encode 1 bit of information, to Miller-4, which uses four high-low combinations, and finally Miller-8, which uses eight.

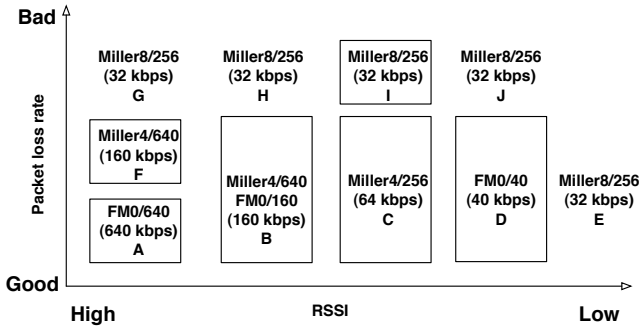


Figure 7: The link metrics map of encoding and baud rate for bit rate adaptation. When RSSI decreases, we should choose encoding/baud rate with lower throughput. When packet loss rate increases, we should increase the complexity of encoding to overcome interference.

4.3.2 *Classifier design*

The RSSI/packet loss to bitrate map shown in Figure 7 provides the intuition for the operation of a classifier, but doesn't provide the boundaries between classes. We learn these boundaries using training data collected from various indoor settings, and use a classifier to map from the current link metrics to the appropriate bitrate.

The classifier uses two features, the sorted list of RSSI across the channels and the sorted list of packet loss across the channels. We use the sorted list since we find that the shape of RSSI and packet loss is a better indicator of the optimal bitrate than using specific channel information. Given the RSSI and loss vectors, the classifier chooses the closest bitrate cluster among those shown in Figure 7. In other words, it chooses cluster i using the following expression:

$$\forall i \in \{A, B, ..., I\} \quad i : \min(|R_i - R|_{l_2} + |L_i - L|_{l_2})$$

where R and L are the current sorted RSSI vector and loss rate vectors, and R_i and L_i are the empirically-determined centroids of the cluster i from Figure 7.

4.3.3 *Mobility-aware channel probing*

An important question when designing a rate adaptation algorithm is the cost of probing in-order to obtain the link metrics across different channels. Unlike the mobility detector, which can passively monitor the RSSI and loss rate metrics at any bitrate that has been currently chosen by upper layers, the channel probe needs to use the simplest encoding and slowest baud rate. This is because of two reasons: 1) using a complex encoding and fast baudrate can mask channel errors, making it harder to estimate the channel accurately, and 2) higher bitrates reduce communication range, and can lead to misses of corner and far-away sensor tags. To give an estimate of the time taken for a channel probe, consider an Impinj reader at the slow FM0/40 setting. By default, the reader spends 100ms in each channel to transmit 7 queries. Since there are 50 channels, the total time for a channel probe is 5 seconds. This is clearly a large number, and results in significant loss of goodput and responsiveness. This section presents a fast probing mechanism that uses two key ideas: a) one-query channel probing, and b) random channel probing.

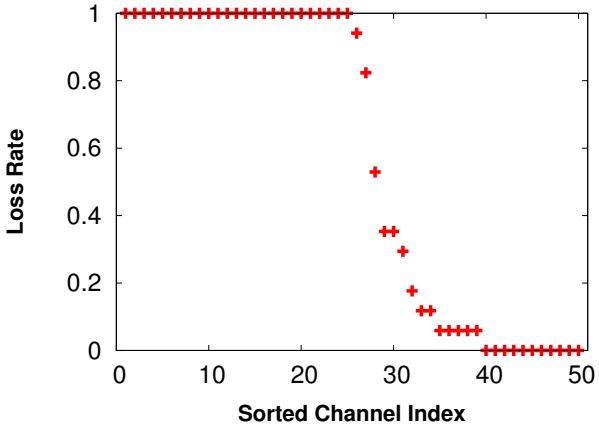


Figure 8: Short transition between "good" and "bad" channels. We plot sorted packet loss rate on 50 channels. Only 10% channels have loss rate between 80% and 20%. 90% channels are either consistently good (zero loss) or consistently bad (over 90% loss).

One-Query Channel Probing: How many queries in each channel are required to estimate its loss rate? The more queries it takes, the more expensive a channel probe. The key idea in this approach is that one query per channel may be enough since backscatter links have a sharp phase transition behavior (from "good" to "bad" channel quality).

To validate our approach, we look at the loss rate patterns across channels on the Impinj reader. Figure 8 shows the packet reception rates sorted from the highest to lowest across the 50 channels. We observe that there is a fairly sharp transition and that channels tend to be either consistently good (zero loss) or consistently bad (over 90% loss). Only 10% of the channels have loss rates between 20% and 80%. This behavior is typical of wireless channels — studies have shown that the transition between low to high loss rates is sharp across distance [31], and we observe that this is true across frequency as well.

The sharp transition makes probing more efficient — a successful packet on a channel is likely be followed by several continuous successful packets, and vice-versa. As result, we can use a single query probe in each channel to estimate channel loss rate. This simple change reduces the overall cost of probing by 7× from 5 seconds to 0.7 seconds.

Random channel probe: A single query per channel reduces probe time but 0.7 seconds is still a long duration, so we look at reducing this even further. This approach exploits the fact that the features used by the classifier are the *shape* of the RSSI and loss rate vectors. Thus, it is not essential to probe all the 50 channels, only to probe as many channels as are needed to accurately estimate the RSSI and loss vector shape. We exploit this idea to perform a random sampling of a subset of the 50 channels (in our case, 10), which reduces the channel probe time by another 5×.

4.4 Channel selection/switching module

The RFID reader hops among 50 channels in the $902MHz \sim 928MHz$ range for communicating with sensor tags. While FCC regulations prevent the reader from dwelling on a single channel for too long, there is some flexibility in how chan-

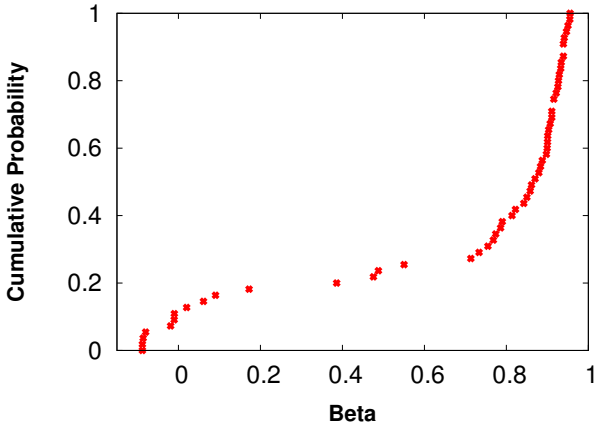


Figure 9: Distribution of β values for the intermediate links. Most links are highly bursty ($\beta > 0.8$) making the case for a fast switching algorithm

nels are chosen. FCC allows 0.4s channel usage within 10s, therefore, the minimal number of channels can be used is 25 rather than 50. When 25 channels are chosen, an RFID reader can stay on each channel for the maximum-allowed 0.4s duration.

Our channel selection module takes advantage of two characteristics of backscatter channels. The first is the sharp transition between "good" and "bad" channels as shown in Figure 8. Since a large number of channels tend to be consistently good or bad, a simple channel selection approach would be to select only the 25 channels that have least packet loss rate, and dwell in each for the maximum allowed duration, 0.4s.

The second characteristic of channels that we take advantage of is burstiness. Wireless channels are known to exhibit bursty loss patterns [24], and this behavior impacts performance of wireless protocols [7]. Channel burstiness has been shown to be mainly caused by RSSI and interference, but can also result from hardware or protocol choices, for example, an RFID reader might scan the field by changing beamformer orientation, resulting in bursty loss characteristics. Figure 9 plots the β metric defined in [24] for channels where we observe intermediate loss rates (i.e. neither 100% nor 0%) across several placements, where $\beta = 1$ means that link is bursty while $\beta = 0$ tells us that independent packet loss is observed. We clearly see that channels with intermediate loss rates tend to be bursty, with success and losses occurring in clusters.

The bursty nature of channels presents an alternate approach to selecting channels. Instead of choosing the top 25 channels, we can use *fast switching* to switch across channels. If a query transmission in a channel fails, the reader can immediately switch to another next channel. This mechanism also helps us deal with occasional loss bursts that we observe due to external interference, and lets us take advantage of intermediate links during bursts of good activity. To ensure that we meet FCC regulations, we keep a counter on the amount of time spent per channel and mark the channel as "used" when it completes the 0.4s dwell time quota.

The transition characteristics and burstiness can depend on the deployment, external interference, and also on the choice of reader hardware. To handle such variability, we use an online approach to measure the sharpness of transitions (fraction of intermediate channels) and the extent of burstiness (beta value). If the channel is not sufficiently bursty, we pick the top 25 channels but if there is high burstiness and sharp transitions, we use the fast switching approach.

5. IMPLEMENTATION

In this section, we describe key implementation details that are not covered in previous sections. Our implementation is entirely done on the reader-side and requires no modifications to the tags.

Gen 2 Reader: Our mobility-aware bitrate adaptation and channel selection protocol is designed to operate on commercial readers that support the EPC Gen 2 protocol. However, despite being compliant with EPC Gen 2, commercial readers often do not expose all the parameters that we need to tune bitrate and channels. On the Impinj reader, we experienced two limitations: a) changing the bitrate was only possible at the beginning of a round, and it took $\tilde{3}$0ms to finish the configuration, which sacrificed some goodput, and b) the reader does not expose hooks that would allow us to select specific channels or change the order of switching between channels. To evaluate aspects of our link layer such as channel selection and switching that are not possible to implement on the Impinj reader, we use trace driven simulations using traces captured by a commercial reader.

USRP Reader: One option that we considered during our implementation is the use of a USRP software radio reader developed by Buettner et al [13] to evaluate our link layer. Clearly, the USRP reader gives us much greater flexibility in terms of evaluating our techniques, but, it also has several hardware limitations that make it hard to fully evaluate BLINK. For example, the USRP has a maximum range of 2.5 meters, making it difficult to observe the multipath interference behavior. In addition, compared to the commercial reader, the achievable goodput with a sensor tag was lower, transitions between good and bad channels were much less sharp, and burstiness was less evident. To avoid hardware artifacts from influencing our results, we use the commercial Impinj reader for almost all our experiments, and sparingly use the USRP reader in cases where the commercial reader does not expose necessary hooks.

Baseline: We compare BLINK against two baseline schemes: a) the default Impinj reader configuration, called "AutoSet", and b) a backscatter-optimized version of SampleRate [9], a widely used WiFi-based rate adaptation algorithm. AutoSet is a rate adaptation algorithm used by default on the Impinj Speedway RFID reader [12]. Although six bit rate configurations are available on the Impinj reader, AutoSet only utilizes three of these — FM0/320 kbps, Miller4/68 kbps, and Miller8/20 kbps. Among these, the Impinj reader uses Miller4/68 when the tag has good connectivity to the reader (close range), and FM0/320 when the tag has bad connectivity (long range). Miller8/20 kbps is used for the very last query to pick up any stragglers that have very poor connectivity to the reader.

We also compare BLINK with SampleRate [9] a commonly used rate adaptation algorithm for WiFi communication. SampleRate maintains a ranked list of bit-rates, based on the average per-packet transmission times observed at each rate. The highest rate on this list is used to transmit data,

except every tenth data packet, which is transmitted at one of the other bit-rates and used as a channel probe. SampleRate stops using a bit rate if it experiences four successive packet losses.

A direct implementation of SampleRate on the Impinj reader turned out to be inefficient since the Impinj reader incurs significant overhead for switching across bit-rates. For example, a single query packet at FM0/640 takes about 6ms whereas switching bit-rates incurs a latency of 30ms. As a result, SampleRate with default parameters performs poorly and can only achieve around 10 reads/s goodput even at close ranges. We therefore optimized SampleRate parameters, and found that it performs best when probes are done for 0.5 secs after every 5 secs of transmission at the best rate. We use these parameters for SampleRate in our evaluation.

Thresholds for link signatures: Our mobility detection protocol uses two distance thresholds, d_T and d_T^*, to determine if a sensor tag is mobile. We now describe how we set these thresholds. Figure 10(a) shows the link signatures of a sensor tag in two cases: a) a stationary tag placed at several different locations in a room, and b) a mobile tag that is placed on a toy train moving along a oval train track at different speeds.

First, we look at the distances for the static case (tag at a single location) vs the mobile case. Figure 10(a) shows a significant difference between the static and mobile cases, and the choice of thresholds ($d_T = 0.3$ and $d_T^* = 0.15$) is relatively straightforward. Second, we peer more deeply into the static cases to see whether we can distinguish between a tag placed at one location vs another. Figure 10(b) shows the RSSI and packet loss distances between one of the locations (referred to as A) and several other locations where the tag was placed. Again, its clear that there is a substantial difference across locations, and the same threshold that we used for the mobile case works in this case as well.

Note that by using the same threshold in the two cases, we are not distinguishing between a tag that has moved to another location vs a tag that is in continuous motion. While a simple extension of looking at a window of distances could address this issue, distinguishing the cases is not important for us since we only care about generating appropriate triggers for rate adaptation and channel selection.

6. EVALUATION

In this section, we evaluate the implementation of our link layer using commercial Impinj readers and passive Alien RFID tags. The evaluation consists of four parts: 1) benchmarking the accuracy of our mobility detection algorithm, 2) validating the link metrics to bitrate map, and evaluating the benefit of our classifier-based bitrate selection algorithm, 3) demonstrating the goodput benefit of using channel selection/switching, and 4) evaluating the overall performance of our high throughput backscatter link layer.

6.1 Mobility detection

The mobility detection module enables a reader to be aware of tag mobility patterns. In this section, we evaluate our mobility detection algorithm in two steps: 1) we benchmark accuracy when we use RSSI and packet loss distance exclusively, and a combination of both to detect mobility, and 2) we compare accuracy when we detect mobility under different choices of bitrate (since a reader can be communicating with tags at different bitrates).

	RSSI	LossRate	RSSI+LossRate
False positive	1.01%	8.04%	9.05%
False negative	5.63%	3.03%	0.87%

Table 2: False positive/negative rate when different channel features are utilized. FM0/640 is used to obtain link signature.

	False positive	False negative
FM0/160	0%	1.67%
Miller4/640	4.08%	10%
Miller4/256	0%	2.38%
Miller8/256	0%	0%
FM0/40	10%	3.92%

Table 3: False positive/negative rate when different bitrates are used to obtain link signature.

Detection accuracy under different link metrics: To evaluate mobility detection accuracy, we attach an RFID tag to a toy train that moves along a 1m × 2.5m oval track. The train follows a stop and move pattern; it moves along the oval track for two mins, stops for three mins, and follows this pattern eight times.

We evaluate the accuracy of mobility detection by measuring the false positive and false negative rate, where the null hypothesis is defined as the tag not being mobile. Given this hypothesis, the false positive rate can be defined as the ratio of being notified mobile when tag is actually stationary, and false negative rate is the ratio of being identified as stationary when tag is in fact mobile.

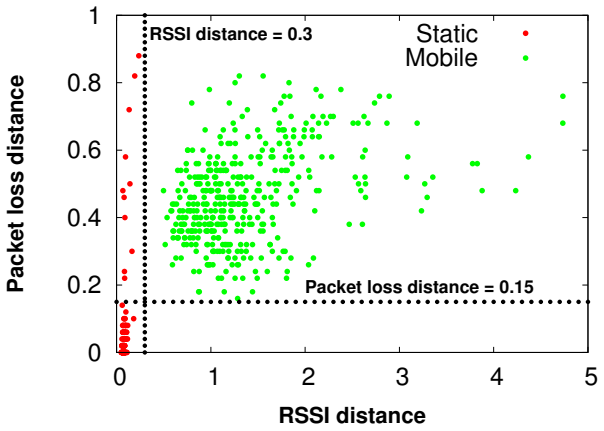
Table 2 shows that by combining RSSI and lossrate features, we can reduce false negatives to under 1%, which is 3× lower than the rate when only loss rate were used, and 5× lower than the rate if only RSSI were used. We lose a bit on false positives since when either RSSI or lossrate distances suggest mobility, we consider the sensor to have moved. This causes more false alarms but our goal was to ensure that actual mobility scenarios were not missed, which is shown to be the case.

Detection accuracy under different bitrates: Since the reader could be operating at different bitrates at different times, one question is how the accuracy of mobility detection is impacted by bitrate. To understand this, we replicate the above experiment, but do this across all bitrates. To reduce the time to run the trace, we use two stops for each. We then measure the false positive/negative rate at different bitrates.

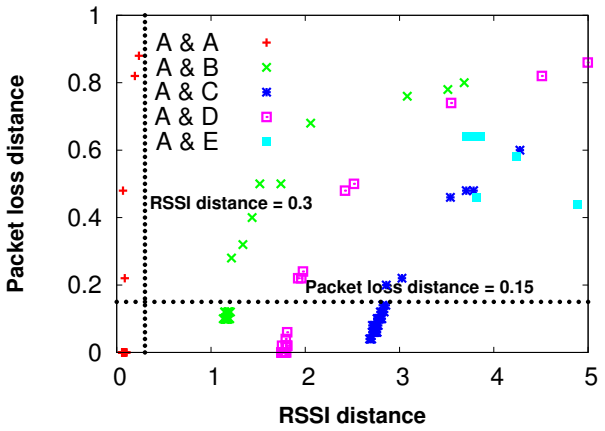
The results are shown in Table 3. We observe that both false positive and negative rates are lower than 10% for all other five bitrates. The results show that no matter which bitrate is selected for obtaining the link signature, our algorithm can achieve similar mobility detection accuracy.

6.2 Rate adaptation

We now turn to the rate adaptation algorithm, which uses a classifier to select the optimal bitrate. We evaluate our rate adaptation algorithm in four steps: 1) we verify the correctness of link metrics to bitrate map, 2) we benchmark the accuracy of the classifier in choosing the best bitrate when full channel probing is done, 3) we evaluate the accuracy



(a) Link signature of static and mobile tag.



(b) Link signature for tag move from location A (1m from reader) to B(4m), C(7m), D(10m), E(12m).

Figure 10: Link signature of static and mobile tag.

of the classifier when fast probing is done, and 4) we compare the goodput achieved by the rate adaptation algorithm against the optimal goodput as well as the one achieved by fast probing.

Correctness of link metrics to bitrate map: We now provide micro-benchmarks that validate the link metrics to bitrate map that we presented in §4.3. We look at two cases — the tag is placed within the beamformer direction, and the tag is placed at the edge of the beamformer direction (i.e. placed close to reader but far from the beamformer direction). In each case, we look at how the optimal bitrate as well as the second-based bitrate changes as the tag is moved away from the reader.

Figure 11(a) shows the results for the case when the tag is placed within the beamformer direction. Multipath self-interference is less likely in this setting, and therefore the optimal bitrate follows a more predictable pattern. As the tag moves away from the reader, the optimal bitrate goes from Block A to B to D, roughly following the order in the lowest row of the map in Figure 7.

Figure 11(b) shows the case when the tag is placed at the edge of the beamformer. As discussed in §4.1, we expect multipath self-interference to be more severe in this case. The results here are more unpredictable. When the tag is 1 meter from reader, the highest bitrate (Block A) is chosen. When the tag moves just a bit further to 1.5m, the optimal bitrate moves from Block A to Block F (as predicted by the first column of Figure 7). This point is clearly caused by multi-path self-interference as described in Figure 3 — RSSI is high, which would suggest excellent channel quality, but packet loss is high as well. The multipath effects reduce when the tag moves a bit more to 2m, and the optimal bitrate changes to Block B. After that, channel quality progressively degrades and the optimal selection moves to Block C and finally to Block I.

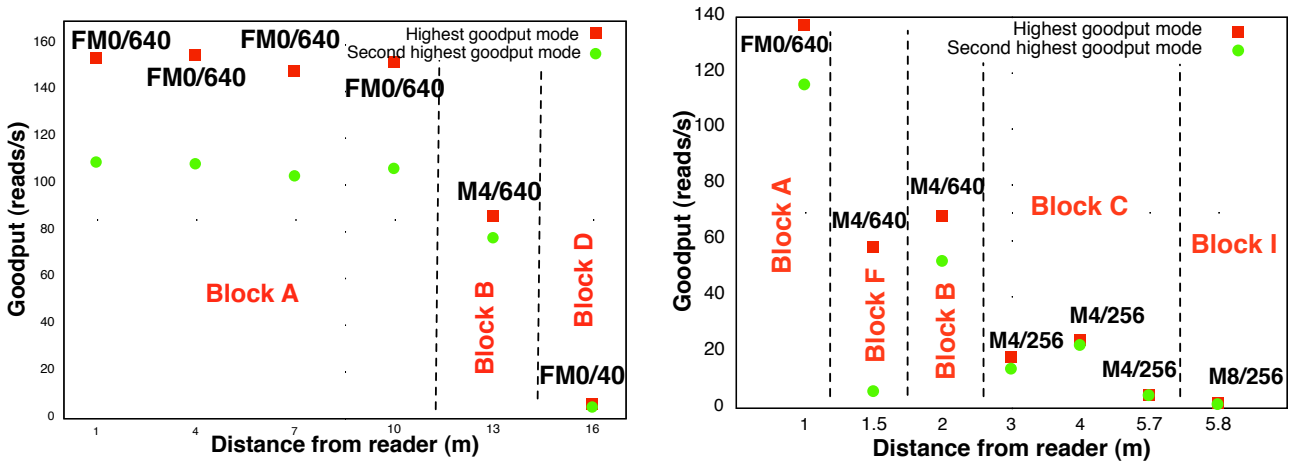
The two figures also show the goodput obtained by the second-highest bitrate (green circles). There are cases where the second-highest bitrate has considerably lower throughput than the optimal one, where errors in classification will be expensive. But there are also several cases where the difference is small, and classifier errors will not lead to significant reduction in goodput.

Bitrate boundaries: While Figure 7 provides an intuition for how the combination of link metrics can be used to predict bitrate, it is an idealized model and the actual boundaries between the different clusters in real-world data are likely to be less regular. Understanding these block boundaries can provide a better idea for how well a classifier would work and where classification errors might occur.

Figure 12 shows the empirically measured optimal goodput mode and their corresponding RSSI and lossrate across several reader-tag distances, placements and times of day. Each circle shows the case when a particular bitrate was selected as the optimal bitrate, and maps to one of the blocks in Figure 7. We did not observe points for some of the clusters, and do not plot them in the figure. As can be expected, the boundaries between clusters are not as regular as shown in the idealized bitrate map, however similar trends can be observed. Also, it can be seen that the classifier might have errors in boundary regions — for example, some points in Block A fall into the Circle B (red points in the green circle), so these might lead to mis-classification. However, we also notice that in these boundary regions, the goodput of both choice A or B is similar, hence both are good choices.

Classifier Accuracy: We turn to evaluating the accuracy of our classifier. To train the classifier, we use 158 sample points from a room over a day, where each sample involves placing the tag at a random location relative the reader, and measuring the optimal bitrate, the RSSI across channels and the packet loss across channels. After the training process, the classifier has an empirically measured RSSI/packet loss to bitrate map.

Table 4 describe the three settings that we use for testing, each of which stresses the classifier in a different way. Group 1 is a dataset from the same room as the training set, but on a different day, Group 2 is a dataset on a corridor with different multipath propagation characteristics, and Group 3 is a dataset in a corridor where we expected substantial multipath since there were a significant number of metal objects including servers nearby. For each test set, we compare



(a) Top two goodput modes when tag is placed within reader beamformer direction

(b) Top two goodput modes when tag is placed at the edge of reader beamformer direction

Figure 11: Top two goodput modes

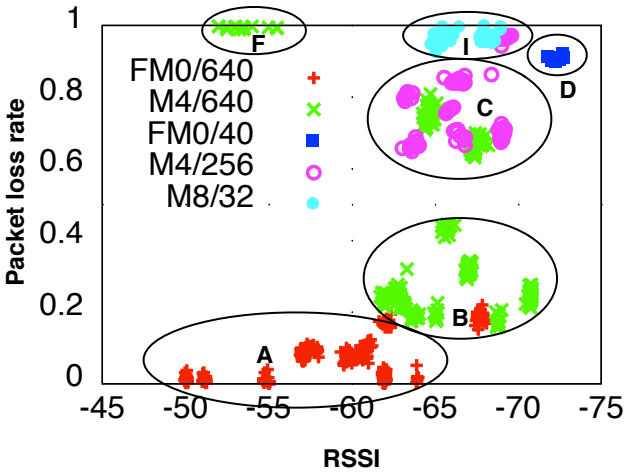


Figure 12: Empirical measured link metric map. It shows the optimal bitrate under different RSSI and packetloss.

the selected bitrate given by classifier to the ground truth optimal bitrate to evaluate the accuracy of the classifier.

The accuracy of classifier in the three groups of experiments is shown in Table 5. We observe that for the first and second groups, the classifier picks up the optimal goodput mode with over 83% accuracy and achieves over 89% of the optimal goodput. For the third group, the accuracy of selecting the optimal bitrate drops dramatically to around 50%, which seems poor. However, a large number of these cases fall into the boundaries between clusters, hence despite low accuracy, the classifier achieves 88% of the optimal goodput.

To summarize, these results show that a) our classifier can achieve approximately 89% of the goodput achieved by optimal bitrate selection under varying conditions, and b) the classifier can be trained in one region and used in several other regions with vastly different propagation and multipath effects without significant effect on goodput.

	Group 1	Group 2	Group 3
Training data	room/day 1	room/day 1	room/day 1
Testing data	room/day 2	corridor1/day 3	corridor2/day 4
Training size	158	158	158
Test size	347	161	162

Table 4: Three groups of experiments for verifying the accuracy of classifier.

Training data	Testing data	Accuracy	% optimal goodput
room/day1	room/day2	83.95%	98.18%
room/day1	corridor2/day3	86.47%	89.75%
room/day1	corridor2/day4	51.38%	88.95%

Table 5: The accuracy for classifier to choose optimal goodput mode. It also shows the percentage of goodput achieved compared with optimal goodput.

Comparison against reader bitrate control: In this evaluation, we ask two questions: a) how does BLINK's selected bit-rate compare to the optimal bit-rate, and b) how does our classifier perform if it only uses RSSI or loss-rate as feature. We use data from all four settings in Table 4 in this evaluation.

Figure 13 shows a CDF of goodput from an optimal scheme that always selects the best bit-rate, the goodput achieved by BLINK, and the goodput obtained by our classifier when utilizing only RSSI/loss rate as feature. We find that BLINK largely follows the optimal case in terms of goodput, and picks a sub-optimal rate only 18% of the time. The loss-rate based classifier is the next best mechanism and picks a sub-optimal bit-rate 30% of the time, although the difference in goodput is only about 3%. The RSSI-based classifier performs the worst, and picks a sub-optimal bitrate 62% of the time, and has on average 15% lower throughput than BLINK. This shows that using a combination of loss-rate and RSSI performs better than using only one of these metrics, and that BLINK performs close to an optimal scheme.

Effect of fast channel probing on bitrate: The results so far assumed that the classifier was given the full channel

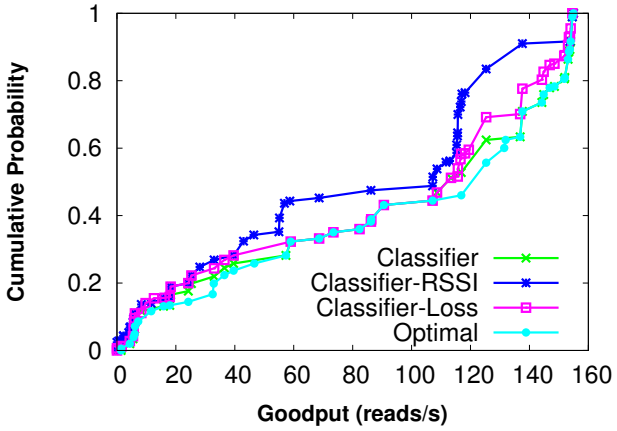


Figure 13: The CDF goodput of optimal mode, mode chosen by classifier, mode chosen by classifier with only RSSI/loss rate feature and default Impinj algorithm.

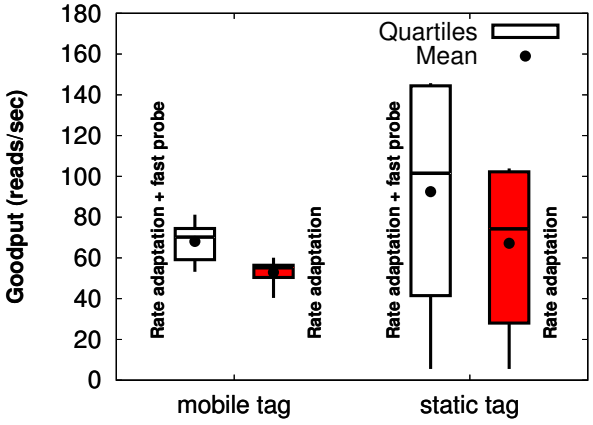


Figure 14: The goodput of rate adaptation and rate adaptation under fast probing for 1 static and mobile tag.

	Accuracy	% optimal goodput
Regular probe	85.45%	97.08%
One Query probe	88.72%	98.06%
10 random probe	73.44%	92.75%
One Query probe + 10 random probe	72.34%	91.48%

Table 6: The accuracy for classifier to choose optimal goodput mode under fast channel probing. It also shows the percentage of optimal goodput retained.

probe *i.e.* full information across 50 channels. As described in §4.3.3, this is very time consuming, and fast channel probing can reduce channel probe time by 35×. So, we ask: how does fast probing impact classification accuracy? Our results are based on a dataset collected in the high-multipath corridor (same environment as corridor2/day4), since it represents the worst case scenario. We use a trace-driven simulation to identify how different schemes would perform.

Table 6 summarizes our results. The second row shows that using one query per channel rather than the default seven queries does as well, or in fact a bit better in terms of classifier accuracy and fraction of optimal goodput. The third row shows that cutting down on the number of channels to sample from 50 to 10 reduces accuracy to 73% but still achieves 93% of the goodput of the optimal scheme. Finally, combining the two metrics has a similar effect. This result shows that on readers where the probe time is very long, a combination of one query per channel and random subset of channels can achieve 35× reduction in probe times without significantly impacting goodput.

Effect of fast channel probing: We now look at the benefits of fast channel probing on performance both for a static tag and mobile tag. In the mobile case, we attach a tag to a LEGO toy train which moves along a 1m×2.5m oval track. The mobility detector triggers rate adaptation roughly every five seconds in this experiment. Figure 14 shows the goodput achieved by the two schemes. Fast channel probing increases goodput by 29% for one mobile tag and 38% for one static tag.

6.3 Channel selection/switching

We now turn to an evaluation of the goodput achieved by the channel selection/switching algorithm. Even though FCC regulations and EPC Gen 2 provides flexibility for channel selection, commercial readers do not provide user-level control over these parameters. As a result, we evaluate the algorithm in two ways. First, we use a trace-driven simulation, where the trace is a repeated 50-channel scan of a tag by an Impinj reader. We stitch the dwell times for each channel together to form a single continuous trace on top of which we emulate the channel selection algorithm. While this trace does not entirely reflect reality, it can give a sufficient picture of the performance of our algorithm. Second, we implement the channel selection algorithm on the USRP software radio reader. The downside of this study is that the USRP reader behaves quite differently from the commercial reader — packet losses are much higher, loss transitions are less steep and burstiness is less evident. Despite this, the USRP reader evaluation provides a baseline for how our techniques would perform when the channel behaves differently from expected.

We evaluate both channel selection and channel switching — the former assumes that lossrate information is available for all channels, whereas the latter switches dynamically and doesn't require prior knowledge of which are good or bad channels. Figure 15 compares the performance of channel selection and switching against the default scheme used by commercial readers.

The results on trace-driven simulations show that channel switching is 2.16× better than the default scheme and channel selection is marginally worse and is 2.06× better than the default. This shows that switching is effective due to burstiness and sharp transitions. For the USRP reader, the results are a bit different — switching is 1.12× better than default and selection is 1.31× better.

6.4 Overall system performance

We now look at the performance of an integrated system on a commercial reader. Since channel selection or switching cannot be enabled on these readers, we do not include the module in our combined evaluation. Our goals are two-

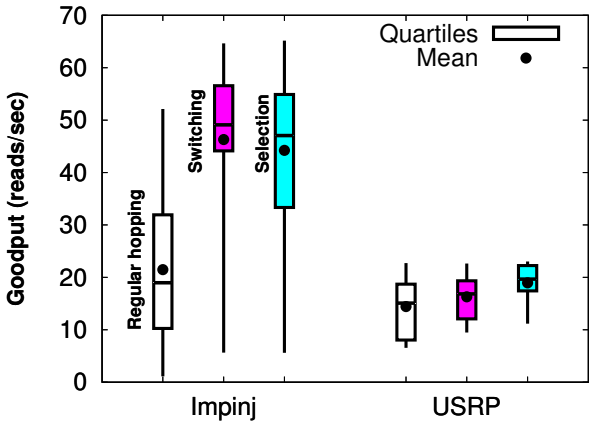


Figure 15: The goodput achieved with channel selection and switching.

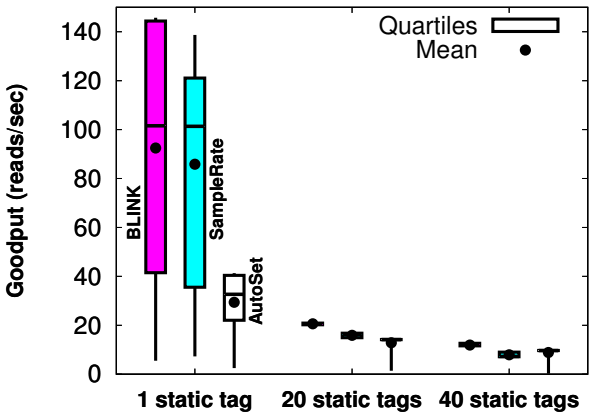


Figure 16: The goodput of AutoSet, SampleRate, and BLINK for 1, 20 and 40 static tags.

fold: a) to provide a comparison between BLINK, AutoSet, and SampleRate, and b) to understand the benefits of our schemes at scale when there are large numbers of static or mobile tags.

First, we look at the static case, where we evaluate performance as we increase the number of static tags placed in front of a reader from 1 to 40. These tags are placed at random locations. To obtain statistically significant results, the results are aggregated over several runs and several placements. Figure 16 shows that for one static tag, BLINK is 3.15× better than AutoSet. SampleRate achieves 2.92× better goodput than AutoSet which is just slightly lower than BLINK. When the tag population scales to 20, BLINK and SampleRate are 1.6× and 1.24× better than AutoSet respectively. When the tag population is further increased to 40, the goodput increase is 1.34× for BLINK but SampleRate degrades and has a throughput of only 0.9× that of AutoSet. This is likely because of the increase in MAC layer collisions leading to higher variability in packet loss rates and consequently poor decision-making by SampleRate. Thus, BLINK has significant benefits as deployment density increases even under static settings.

Second, we look at the mobile case. We place one, five, and ten RFID tags at fixed locations on a person who moves continuously in a 15m×15m room. The movement pattern involves walking towards and away from the reader, and walking in a circle along the edges of the room. We repeat the same mobility trace at roughly the same speed to obtain a fair comparison among BLINK, SampleRate and AutoSet. Figure 17 show the summary statistics across 10 mobility traces for each scheme. We see that for one mobile tag, BLINK is 2.29× better than AutoSet, and SampleRate is 1.77× than AutoSet. However, as the number of mobile tags grow, the benefits of BLINK increases. When 5 mobile tags are employed, the gains of BLINK and SampleRate over AutoSet are 1.96× and 1.42× respectively. When tag population scales to 10, BLINK is 2.44× better than AutoSet, whereas SampleRate is only 1.21× better than AutoSet. These results clearly demonstrate the benefits of BLINK across a wide range of scenarios and scales for a realistic number of RFID sensors.

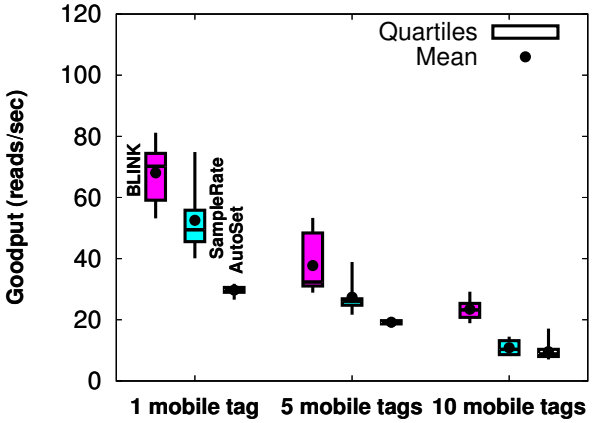


Figure 17: The goodput of AutoSet, SampleRate, and BLINK for 1, 5, and 10 mobile tags.

7. RELATED WORK

There has been substantial prior work on wireless channel characteristics and link layers. Our primary contribution is to design a link layer that is optimized to the unique characteristics of backscatter communication.

Backscatter system: Much of the work on backscatter communication is specific to optimizing communication from EPC Gen 2 tags, for example, better tag density estimation [26], better search protocols to reduce inventorying time [19], better tag collision avoidance [20], more accurate tag identification [29], better recovery from tag collisions [8]. Some work has also looked at how physical layer parameters account for the major propagation characteristics of the backscatter channel [14]. Other work has looked at more efficient use of harvested energy, for example Dewdrop [10] balances task demands with available energy, and our prior work uses hybrid energy harvesting to increase communication range to many tens of feet [15]. None of these tackle link-layer adaptation mechanisms to improve goodput.

ECP Gen 2 optimization: Recent work with Computational RFID devices such as the Intel WISP includes Buet-

tner et al [11], who explore how the RFID PHY layer interacts with EPC Gen 2 and identify physical layer parameters that degrade overall performance and reliability. [11] also suggests adapting the link layer to improve goodput by using FM0/640 kbps to read tags until only low SNR tags are left (because they can't be read at the high rate), and then changing to Miller-8/256 kbps to read the rest of the tags. Our work addresses backscatter bit-rate selection in a much more rigorous manner.

In an effort complementary to this paper, we have designed Flit, a high-throughput and low-power MAC layer that addresses inefficiencies of EPC Gen 2 for bulk data transfer from sensors [16]. BLINK is complementary to this effort and differs in two ways: 1) Flit involves optimizations at the protocol level of the EPC Gen 2 MAC, whereas BLINK is entirely at the link layer and agnostic of how the protocol works, b) Flit involves optimizations on the tag-side and therefore requires programmable RFID tags (e.g. WISP) whereas BLINK is entirely at the reader-side and can operate with any commercial RFID tag.

Link signatures: Recent work has also explored the problem of location distinction using link signatures [30]. The main idea is to form a signature of the channel impulse response from physical layer data at a receiver, and use this to detect location changes. This is tested for WiFi software radios where PHY information is available. At a high level, our mobility detector uses similar ideas, but we differ in our focus on backscatter communication and link metrics that are relevant to capturing the channel signature for RFIDs. Link signatures have also been used in localization — for example, SkyLoc [25] exploits GSM signal strength fingerprinting for identifying the current floor of a user in tall multi-floor buildings. While BLINK exploits radio fingerprints as well, it does so for location distinction, which is simpler and does not involve an extensive training phase.

Estimating channel quality: Link layer mechanisms for active radios rely on different link metrics including SNR, loss rate and PHY-layer hints. SNR is used by AccuRate [23], which uses it to select the appropriate constellation map, FARA [21], which uses per-frequency SNR to evaluate the depth of fading in each channel of 802.11, and [17], which proposes an effective SNR metric to predict the highest link bit rate. Packet loss rates are widely used, for example, [28] uses short-term loss ratio to opportunistically guide its rate change decisions. More recently, PHY-layer hints have been used, for example SoftRate [27] uses "SoftPHY" or confidence values conveyed by PHY to learn the bit error rate (BER). Our work addresses unique aspects of backscatter communication where we find RSSI and packet loss rate are needed in conjunction to capture path loss and multipath fading characteristics.

Channel selection: [24] investigates β-factor metric to measure intermediate link burstiness and proposes opportune transmissions to improve bursty link packet reception ratio. In contrast, our work exploits burstiness for fast channel switching in search for better transmission opportunities. Our work also differs from CSMA-based techniques for channel selection (e.g. [18]) since we deal with backscatter-specific considerations.

8. CONCLUSION

Backscatter communication offers an attractive alternative to existing active-radio based sensor systems. However, backscatter has predominantly been used for communication with RFID tags which have limited data to transfer, and there has been limited prior work on understanding how backscatter can be used for high-throughput communication between readers and sensors. Our work bridges the gap by providing an in-depth exploration of the physical layer behavior of backscatter links, what metrics are needed to capture this behavior, and how bitrate adaptation and channel selection can exploit these channel characteristics to improve throughput while remaining within FCC guidelines for the UHF RFID frequency range. Results show that our link layer, BLINK, provides substantial throughput benefits for a range of channel conditions, scales, and mobile conditions.

9. ACKNOWLEDGEMENT

We thank Prof. Lakshminarayanan Subramanian for shepherding and the anonymous reviewers for their valuable comments. This research was supported by NSF grants 0855128, 0916577 and 0923313.

10. REFERENCES

[1] A Batteryless Programmable RFID-Scale Sensor Device. `http://spqr.cs.umass.edu/moo/`.

[2] How to NFC. `http://developer.android.com/videos/index.html#v=49L7z3rxz4Q`.

[3] RFID Basics: Backscatter Radio Links and Link Budgets. `http://rfidtribe.com/index.php?option=com_content&view=article&id=423&Itemid=102`.

[4] The Impinj UHF Gen 2 Speedway RFID Reader. `http://www.impinj.com`.

[5] Ultracapacitor Offers 75-Foot Read Range for Passive Tags. `http://www.rfidjournal.com/article/articleview/8565`.

[6] WISP: Wireless Identification and Sensing Platform. `http://seattle.intel-research.net/wisp/`.

[7] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11 b mesh network. In *ACM SIGCOMM*, 2004.

[8] C. Angerer, R. Langwieser, and M. Rupp. Rfid reader receivers for physical layer collision recovery. *IEEE Transactions on Communications*, 2010.

[9] J. Bicket. *Bit-rate selection in wireless networks*. PhD thesis, Massachusetts Institute of Technology, 2005.

[10] M. Buettner, B. Greenstein, and D. Wetherall. Dewdrop: an energy-aware runtime for computational rfid. In *USENIX NSDI*, 2011.

[11] M. Buettner and D. Wetherall. An empirical study of uhf rfid performance. In *ACM MobiCom*, 2008.

[12] M. Buettner and D. Wetherall. A gen 2 rfid monitor based on the usrp. *ACM SIGCOMM CCR*, 2010.

[13] M. Buettner and D. Wetherall. A software radio-based uhf rfid reader for phy/mac experimentation. In *2011 IEEE International Conference on RFID*, pages 134–141, 2011.

[14] J. Griffin and G. Durgin. Complete link budgets for backscatter-radio and rfid systems. *Antennas and Propagation Magazine, IEEE*, 51(2):11–25, 2009.

[15] J. Gummeson, S. Clark, K. Fu, and D. Ganesan. On the limits of effective hybrid micro-energy harvesting on mobile crfid sensors. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 195–208. ACM, 2010.

[16] J. Gummeson, P. Zhang, and D. Ganesan. Flit: A bulk transmission protocol for rfid-scale sensors. In *MobiSys*, 2012.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. *ACM SIGCOMM*, 2010.

[18] N. Jain, S. Das, and A. Nasipuri. A multichannel csma mac protocol with receiver-based channel selection for multihop wireless networks. In *IEEE ICCCN*, 2001.

[19] C. Law, K. Lee, and K. Siu. Efficient memoryless protocol for tag identification. In *Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 75–84. ACM, 2000.

[20] V. Namboodiri and L. Gao. Energy-aware tag anticollision protocols for rfid systems. *Mobile Computing, IEEE Transactions on*, 9(1):44–59, 2010.

[21] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. Frequency-aware rate adaptation and mac protocols. In *ACM MobiCom*, 2009.

[22] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden. Improving wireless network performance using sensor hints. In *USENIX NSDI*, 2011.

[23] S. Sen, N. Santhapuri, R. Choudhury, and S. Nelakuditi. Accurate: Constellation based rate estimation in wireless networks. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, pages 12–12. USENIX Association, 2010.

[24] K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *ACM SenSys*, 2008.

[25] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara. The skyloc floor localization system. In *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*, pages 125–134. IEEE, 2007.

[26] H. Vogt. Efficient object identification with passive rfid tags. *Pervasive Computing*, pages 98–113, 2002.

[27] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM CCR*, 2009.

[28] S. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM Mobicom*, 2006.

[29] D. Zanetti et al. Physical-layer identification of uhf rfid tags. In *ACM MobiCom*, 2010.

[30] J. Zhang, M. Firooz, N. Patwari, and S. Kasera. Advancing wireless link signatures for location distinction. In *ACM MobiCom*, 2008.

[31] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, 2003.