Parallel Backscatter in the Wild: When Burstiness and Randomness Play with You

Meng Jin
Northwest University
School of Software and BNRist, Tsinghua University
mengj@mail.tsinghua.edu.cn

Yuan He
School of Software and BNRist, Tsinghua University
heyuan@tsinghua.edu.cn

Xin Meng
Northwest University
mengxin@stumail.nwu.edu.cn

Dingyi Fang
Northwest University
dyf@nwu.edu.cn

Xiaojiang Chen
Northwest University
xjchen@nwu.edu.cn

ABSTRACT

Parallel backscatter is a promising technique for high throughput, low power communications. The existing approaches of parallel backscatter are based on a common assumption, i.e. the states of the collided signals are distinguishable from each other in either the time domain or the IQ (the In-phase and Quadrature) domain. We in this paper disclose the superclustering phenomenon, which invalidates that assumption and seriously affects the decoding performance. Then we propose an interstellar travelling model to capture the bursty Gaussian process of a collided signal. Based on this model, we design Hubble, a reliable signal processing approach to support parallel backscatter in the wild. Hubble addresses several technical challenges: (i) a novel scheme based on Pearson's Chi-Square test to extract the collided signals' combined states, (ii) a Markov driven method to capture the law of signal state transitions, and (iii) error correction schemes to guarantee the reliability of parallel decoding. Theoretically, Hubble is able to decode all the backscattered data, as long as the signals are detectable by the receiver. The experiment results demonstrate that the median throughput of Hubble is 11.7× higher than that of the state-of-the-art approach.

MobiCom '18, October 29-November 2, 2018, New Delhi, India

ACM ISBN 978-1-4503-5903-0/18/10…$15.00
https://doi.org/10.1145/3241539.3241544

CCS CONCEPTS

• **Networks** → **Wireless access networks**; *Network architectures*;

KEYWORDS

Backscatter; Wireless; Parallel Transmission

ACM Reference Format:
Meng Jin, Yuan He, Xin Meng, Dingyi Fang, and Xiaojiang Chen. 2018. Parallel Backscatter in the Wild: When Burstiness and Randomness Play with You. In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18), October 29-November 2, 2018, New Delhi, India.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3241539.3241544

1 INTRODUCTION

Backscatter, as a battery-free communication technology, enables Internet of Things (IoT) devices to sense and transmit data at ultra-low cost, hence becoming attractive to a broad range of applications [10, 12, 13, 16–20]. As the IoT technology proliferates in areas like logistics, warehouses, manufacture, retail, and supply chains, the deployment scope and scale of backscatter devices (e.g. RFID and WISP tags) have explosively grown in recent years. How to efficiently gather data via backscatter becomes a crucial problem. Under this circumstance, parallel backscatter is proposed. When transmissions from multiple tags come in parallel, the aggregate throughput is expected to be much higher than that of the conventional approaches [4, 7, 21–23].

It is however a non-trivial task to parallelize backscatter in practice. The parallel backscatter signals will generally collide with each other at the receiver (e.g. the reader antenna), making it hard to recover the data from each tag. If taking the low-power communication and environmental influence into account, the collided signals may be noisy and variational, which further increases the difficulty of parallel decoding.

Backscatter generally adopts ON-OFF keying modulation to encode data [7], which accordingly generates two signal states. At the core of parallel backscatter is a process to identify the state of every collided signal. The existing works to tackle this problem [5, 6, 8, 11] mainly exploit the features of signal states and state transitions in the time and IQ (the In-phase and Quadrature) domains to accomplish parallel decoding. *A common assumption behind is that the state of the collided signals are distinguishable.*

When we implement the existing approaches with backscatter tags in the wild, we observe a huge gap between the theory and the practice. The reason lies in the follow aspects:

First, backscatter signals are noisy by nature. Viewed in the IQ domain, the collided signals at the same state form a cluster rather than fall at a single point. The radius of the cluster is generally determined by the noise level. So the signal clusters will expand in noisy environments.

Second, the distances among the cluster centers in the IQ domain are generally determined by the received signal strengths (RSS) at the reader. In reality, weak RSS will lead to short inter-cluster distances. When expanded clusters get closer to each other, they are likely to overlap, as is so-called the *Superclustering Phenomenon*. Superclustering confuses the collided signals as well as the clusters, which obstructs parallel backscatter in practical scenarios.

Last but not least, the number of clusters exponentially increases with the degree of parallelism. When more tags join the parallel backscatter, the chance of superclustering dramatically increases, which further decreases the decoding rate (the percentage of packets successfully decoded).

In order to the address the above problem, we propose Hubble, a reliable signal processing approach to support parallel backscatter in the wild. The design of Hubble stems from the following key insight: the spatiotemporal distribution of the collided signals follows a *bursty Gaussian process*, which is the combination of the burstiness in the time domain and the gaussian property in the IQ domain. Based on this insight, we propose an interstellar travelling model, which accurately characterizes the underlying but deterministic signal state transitions with random noise. By exploiting the rich processing capacity at the backscatter receiver, one can use this model to extract the signal states as well as trace the state transitions. Our contributions in this work are three-folded:

- Through extensive observations and experiments, we disclose the superclustering phenomenon and its negative impact, which exists in almost all the parallel backscatter scenarios. In the process of exploring the root causes of superclustering, we find that the spatiotemporal distribution of the collided signals follows the bursty Gaussian process and propose the interstellar travelling model to characterize this process.
- We propose the Hubble approach, which addresses several unique technical challenges in parallel decoding: (i) a novel scheme based on Pearson's Chi-Square test, which contrasts the signal's temporal burstiness to the noise's randomness, helps to extract the collided signals' combined states. (ii) a Markov driven method is designed to capture the law of signal state transitions. Using this method, one can continuously trace the underlying state of each signal. (iii) error correction schemes are designed to further guarantee the reliability of parallel decoding.
- We implement Hubble and evaluate its performance across different scenarios. Hubble significantly enhances the practical usability of parallel backscatter. Theoretically, Hubble is able to decode all the backscattered data, as long as the signals are detectable by the receiver. The experiment results demonstrate that the median throughput of Hubble is 11.7× higher than the state-of-the-art approachs.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the background knowledge and the motivation of this work. In Section 4 we elaborate on the design of Hubble. After discussion on several important issues in Section 5, we present the evaluation results in Sections 6. Section 7 concludes this work.

2 RELATED WORK

As we mentioned in the previous section, the central task of parallel backscatter to identify the state of every collided signal, so that the backscatter data can be decoded. Depending on the schemes to distinguish different signal states, the existing works can be classified into three categories.

Decoding based on IQ domain information. Early works on parallel decoding [1, 2, 9, 14, 15] assume that channel coefficients of the tags are stable and add up linearly at the receiver when collision occurs. Therefore, the collided signal can be separated and decoded according to their locations on the IQ domain. In practice, however, channel coefficients keep changing due to the dynamic environment and tag movement. Under such conditions, the above methods have to re-estimate the channel coefficients frequently to deal with channel dynamics, which means apparently high overhead.

Decoding based on time and IQ domain information. Recent works propose to decode the collided signals by simultaneously exploiting the time and IQ domain information [5, 6, 11]. The underlying assumption is that different tags start their transmissions with different delays and flip their states with predictable intervals during transmissions. Therefore, signal transition caused by different tags can be

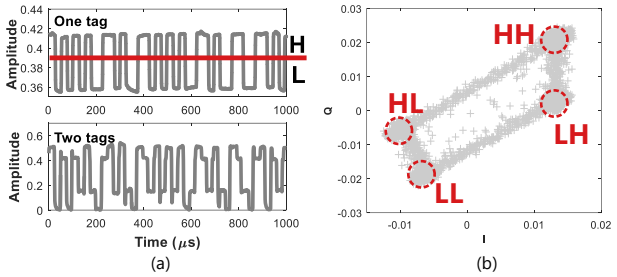

Figure 1: Signals from backscatter tags: (a) time domain signals from a single tag (top) and two concurrent tags (bottom). (b) IQ domain signals.

identified and separated based on their timings. By further introducing the IQ domain information, the signal transmitted by each individual tag is decoded.

The advantage of those works is that they don't rely on knowledge of channel coefficients or linear dependency among collision states. The limitation is that they require precise timing, i.e. low drifting rate (less than 200 ppm is tolerable [6]) of the built-in clock of tags, to ensure flipping of different tags occur at predictable time points. The reality is that the COTS (Commercial off the shelf) tags exhibit high drift rates, which are between 40,000 ppm and 68,000 ppm [6, 8]. The intrinsic clock drifts cause degradation of decoding rate in practice.

Decoding based on the law of state transitions. A recent work FlipTracer [8] propose to identify states of signal with a so-called one-flip-graph. FlipTracer is designed based on the finding that transitions between signals' combined states follow identical and stable probabilities. By tracing the computed one-flip-graph, Fliptrace is able to identify all the signal states in the IQ domain, while doesn't rely on signal's stability in either the time or the IQ domain. However, Fliptracer still implicitly relies on the same assumption as the other existing works. That is, the state of the collided signals are *distinguishable* from each other in either the time domain or the IQ domain.

According to the above discussion, we now better understand how the existing works improved the practical usability of parallel backscatter. At the same time, we observe that in addition to the limited degree of parallelism, another factor that limits the applicability of parallel decoding is the requirement on signal quality. Specifically, due to the superclustering phenomenon, the existing proposals only work in the ideal conditions, where the backscatter signals are relatively strong. That means the tags have to be located very close to the reader without obstacles. Imagine the real IoT applications with backscatter devices, the aforementioned superclusering phenomenon is likely to happen in almost all the scenarios. This fact is however ignored by the existing approaches. How to achieve reliable decoding of parallel backscatter in the wild? This is a problem with great significance in both research and application fields.

3 BACKGROUND AND MOTIVATION

This section starts with brief introduction to the background knowledge. Then we exemplify and discuss the challenges of parallel backscatter in the wild. In the third subsection, we present important observations on the signals' burstiness vs. randomness, which motivate the design of Hubble.

3.1 Primer

A backscatter tag transmits its data through ON-OFF keying modulation. The signal, which has two states, i.e., high (H) and low (L), can be decoded using a magnitude threshold, as shown in Figure 1(a-top). In parallel transmission cases, signals from N tags collide at the reader side, creating 2^N energy levels, each representing a specific combination of the tags' states (denoted by $S = [s_1, s_2, ..., s_N]$, where $s_i = H$ or L indicates the state of tag i). As an example, Figure 1(a-bottom) shows a collided signal of two tags. We can see that the gap between different energy levels is not stable. Decoding solely based on the energy profile is infeasible.

A typical scheme to address the above problem is to acquire richer information of the collided signals from the IQ domain [5, 6, 8, 11]. As shown in Figure 1(b), due to different phases and signal strengths, the collided signals form four clusters in the IQ domain. If the signal clusters are clearly separated from each other, we can associate each cluster to a specific combined state using the existing approaches [5, 6, 8, 11]. Then, tracing signals' state transitions among the clusters tells the transmitted bits of each tag. The above is the basic mechanism of parallel decoding.

3.2 Challenges

The above mentioned parallel decoding method is however brittle in practice. The fundamental problem is that signal clusters in the IQ domain are not always distinguishable from each other. As we have explained in the previous section, the noise level in the environment determines the radius of a cluster, while the RSS of the signals determines inter-cluster distances. In practical deployments of backscatter systems, the noise level is often comparable with the signal strength. Consequently, the signals' clustering results may be far from the desired state.

As an example, Figure 2(a) shows the collided signals from three tags that are located only 1 feet away from the reader. We can observe 8 clearly separated clusters in this case. When we move one of the tags to 3 feet away from the reader, the decrease in the tag's RSS leads to the occurrence of overlapped clusters, as shown in Figure 2(b). In this case, multiple clusters overlap with each other, forming a *supercluster*.

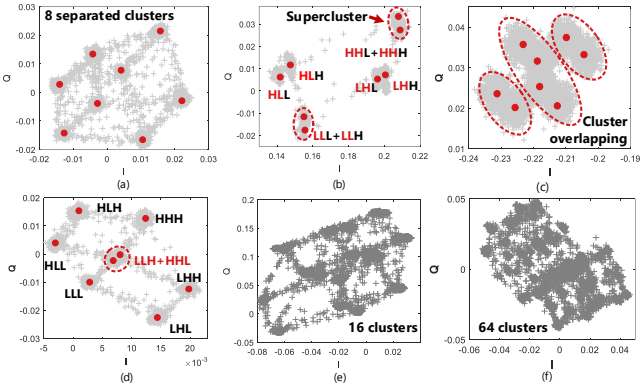

Figure 2: Examples of the collided signals on IQ domain: (a) three strong tags; (b) coexistence of strong and weak tags. (c) three weak tags; (d) coincidental overlapping; (e) four strong tags; (f) six strong tags.

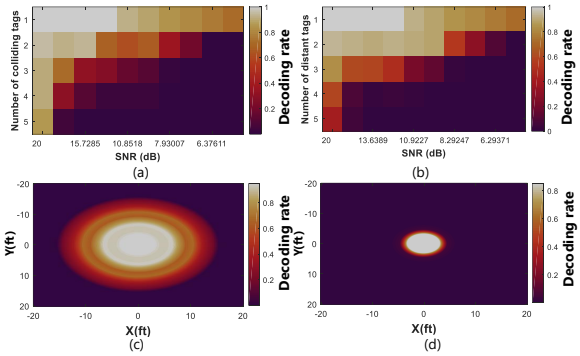

Figure 3: (a)-(b) Performance of FlipTracer and BiGroup under different SNRs; (c)-(d) Distribution of FlipTracer's decoding rate when 2 and 5 tags transmit in parallel.

The receiver even cannot figure out how many signal states should be associated with a supercluster, not to mention tracing state transitions. Figure 2(c) shows that when all the three tags are located 3 feet away from the reader, almost every cluster is merged into a supercluster. None of signal's combined states can be easily extracted from the IQ domain.

It is worth noticing that weak signals are not the only cause of superclustering. Even when the signals are all strong, superclusters also exist with a certain chance. As shown in Figure 2(d)), the colliding signals may coincidentally generate overlapping clusters. This phenomenon becomes much more serious when more tags transmit concurrently, as shown in Figures 2(e) and (f), where the degree of parallelism is 4 and 6, respectively.

With regard to the superclustering phenomenon, we observe a huge gap between the theory and the practice, based on our implementation of two existing approaches FlipTracer [8] and BiGroup [11] with backscatter tags in the wild. Specifically, we carry out a group of experiments under different settings of the tag-reader distance and the degree of parallelism. All the tags are positioned at the same distance from the reader. We use SNR (signal-to-noise ratio) rather than the real distance as the X-axis because SNR is a general metric across different contexts. Clearly, the reader-perceived SNR decreases as the tag-reader distance increases. Figures 3(a) and (b) respectively show the decoding rates (the percentage of packets successfully decoded) of the two approaches. We can see that the theoretical results claimed by the existing approaches are achievable only when the tags are very close to the reader and the degree of parallelism is small. The decoding rates seriously degrade when either the distance or the degree of parallelism increases.

Based on this finding and the signal attenuation model, we calculate the distribution of FlipTracer's decoding rate when 2 and 5 tags transmit in parallel (the result of BiGroup is similar and omitted here to save space). Figures 3(c) and (d) conceptually plot the results in a 40×40 2D area. Suppose the reader is at the center, tags deployed anywhere in the area should be within the communication range of the reader. Surprisingly and reluctantly, we find that the existing approaches achieve decent performance only in a very small portion of the entire area. That means the practical usability of the existing approaches is far restricted in terms of space and degree of parallelism.

Given the above observation and analysis, it is not hard to understand that superclusering is a critical problem which seriously affects the performance of parallel backscatter. One may conjecture that a density-based clustering algorithm (e.g. the one adopted by FlipTracer [8]) may help to resolve the superclustering problem. But that is not true in the case of parallel backscatter. Figures 4(a)-(c) plot the density distributions of signals of three difference cases. We can see that when cluster overlap with each other, the overlapped area has a comparable density with that of the central cluster areas, because the overlapped area gathers signals from multiple clusters. In fact, only utilizing the IQ domain information is no longer effective in identifying different clusters.

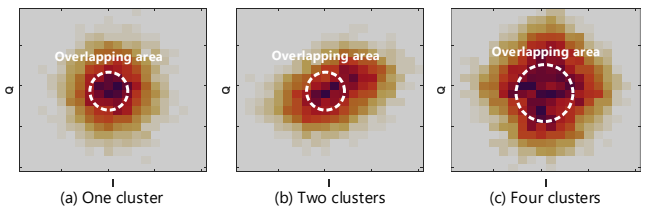

Figure 4: Density distribution of collided signal.

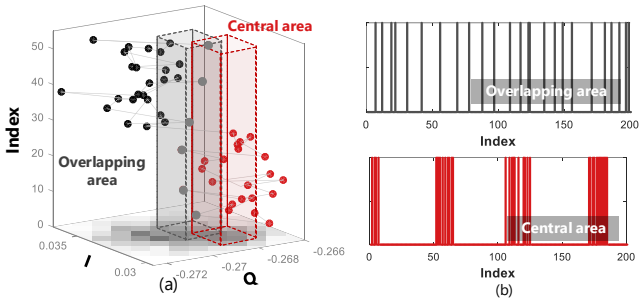

Figure 5: (a) The bursty Gaussian behavior of signal; (b) temporal characteristic of signals that confined in overlapping area and central area.

3.3 Randomness v.s. Burstiness

How to distinguish the overlapping area from the central cluster area? Observing the signal's temporal characteristics in the IQ domain provides a new angle to tackle this problem.

Specifically, since the reader can oversample the received signals, one state of the collided signal generally correspond to many samples at the reader side. Embodied in the sample sequence, the collided signal will always continuously dwell on a state for a certain period of time, before transiting to another state. That is called the *temporal burstiness* of a collided signal. Due to such burstiness, during any short period of time, the spatial distribution of signals in the IQ domain tends to be concentrated in the corresponding cluster. At the same time, the noise, which is generally supposed to be a Gaussian factor, will make the signals randomly deviate from the cluster. Figure 5 plots this interesting process in a 3D space, where the Z-axis denotes time (measured by the index of signal samples).

We have two important findings from the above observations and analysis:

- Although the central cluster area and the overlapped area have similar signal densities, the arrival of the signal samples in these two kinds of areas exhibit apparently different levels of randomness/burstiness (as shown in Figures 5 (b)). In other words, the signals falling into the central cluster area is a deterministic bursty event, while the signals falling into the overlapping area is a random event. This key finding illuminates a way to effectively find the centers of all the clusters, even when they overlap with each other.
- Due to the predefined data rate of communication and oversampling, the spatiotemporal distribution of collided signals follows a probabilistic pattern. Such a traceable pattern (like the expected dwell time at a state and the transition probability between different states), combined with locations of the signal samples in the IQ domain, provide fine-grained basis for accurately tracing the underlying state transitions.

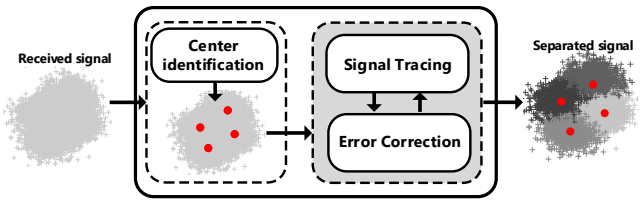

Figure 6: Workflow of Hubble.

4 HUBBLE DESIGN

Hubble is designed to accurately extract the underlying signal state sequence from the seriously jumbled collision signal based on the bursty Gaussian behavior of the signal. Figure 6 summarizes the sketch of Hubble, which is composed of three components. Specifically, Hubble first identifies the locations of the cluster centers (which represents different signal states) in the IQ domain, using the *center identification* component. After that, Hubble begins to trace signal's state transitions, during which the *signal tracing* and *error correction* components are alternated periodically to guarantee high reliability.

In the following of this section, we first propose the interstellar travelling model to capture the bursty Gaussian behavior of the signal, followed by the design details of the three main components of Hubble.

4.1 The interstellar travelling model

In this section, we propose our interstellar travelling model to capture the bursty Gaussian behavior of the signal. We start by briefly introducing the standardized coding (FM0 or Miller) used in backscatter transmission. Here we take FM0 coding for example (Miller coding exhibits similar characteristic with FM0). FM0 flips signal state at every bit boundary, and the bit 0 has an additional mid-bit state flipping. Considering that the reader oversamples the received signal, signal will dwell on a state before transiting to the other state, showing strong burstiness. Suppose the sampling rate of the reader is M_R, and the bitrate of the tag is B_T, then signal will dwell on one state for either $\frac{M_R}{B_T}$ (when transmitting bit 0) or $\frac{M_R}{2 \cdot B_T}$ (when transmitting bit 1) samples' length.

Then let's look at the scenario when N_T tags transmit in parallel. In this scenario, the collided signal has $K = 2^{N_T}$ states, denoted by $S_1, ..., S_K$. The dwell time of signal on a state indeed follows the exponential distribution.

Specifically, the exponential distribution describes the time interval between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a stable average rate. There are two main

properties of the collided signal that make the dwell time follow the exponential distribution: First, the average interval between successive signal flipping (i.e., the dwell time of the signal) is solely determined by the number of concurrent tags and the bitrate of each tag, which are relatively stable over time. Second, different tags have different response delays (which is caused by the differences in tags' power charging rate and strength of incident radio power), and the bit duration of the tags is neither identical among tags nor stable over time (due to the high drifting rate of the clock). This makes the dwell time of the collided signal highly random and unpredictable. In conclusion, the dwell time is random but averaged at a stable value, which makes it follow the exponential distribution.

Based on the above discussion, the probability that a signal will dwell on a state for t_d samples can be estimated by

$$P_d(t_d) = \frac{1}{T_d} \cdot e^{-\frac{1}{T_d} \cdot t_d} \tag{1}$$

where T_d is the average dwell time of the signal on a certain state. If a packet contains similar numbers of 0 and 1 bits, we have $T_d = \frac{2M_R}{3N_T \cdot B_{ave}}$. Here, B_{ave} is the average bitrate of the colliding tags.

We further conduct a set of experiments to verify our assumption assumption on the exponential distribution. In the experiment, we let 5 tags transmit concurrently, and the reader samples the signal at 25MHz. Bitrates of the 5 tags change from 50 Kbps to 250 Kbps. Figure 7 shows the observed distribution of t_d and the fitted exponential distribution under different B_{ave}. We find that the dwell time of the signal properly fits the exponential distribution.

Due to the time-domain burstiness of the collision signal described above, we can estimate the probability that the i-th signal sample is on state S_k based on $P_d(t_d)$ and the state of the previous sample:

$$P_T(i, S_k|t_d) = \begin{cases} P_d(t_d), & s_{i-1} = S_k \\ (1 - P_d(t_d))P_{trans}(S_q, S_k), & s_{i-1} = S_q \end{cases} \tag{2}$$

Here $P_{trans}(S_q, S_k)$ is the transition probability between states S_q and S_k ($S_q \neq S_k$), and s_i denotes the state of sample i. Equation (2) indicates that for the i-th signal sample, the probability that it is on a certain state S_k depends on: i) the state of the $(i-1)$-th signal sample (i.e., whether the $(i-1)$-th sample is on state S_k); ii) the probability that the signal will stay on a certain state given that it has stayed for t_d samples; and iii) the transition probabilities among different states.

Equation (2) captures signal's burstiness in time domain. Now we explore signal's Gaussian property in the IQ domain. Specifically, due to the noise, received signal samples on the same state S_k are dispersed and scattered around a centroid position (denoted by (I_k, Q_k)) in the IQ domain. Assume that the noise follows the Gaussian distribution, then the

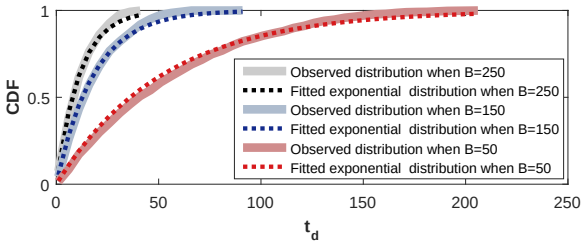

Figure 7: Distribution of signal's dwell time.

probability that a signal sample on state S_k is located at a location (I, Q) can be estimated by:

$$P_{IQ}((I, Q)|S_k) = \pi(I, Q, I_k, Q_k, \Sigma) \tag{3}$$

where $\pi(\bullet)$ is the probability density of the Gaussian distribution. Σ is the covariance matrix, which is related to the noise level.

Combining P_T and P_{IQ} provides an estimation of the probability that the $i-th$ signal sample is located at (I, Q):

$$P_{BG}((I, Q)|i) = \sum_{k=1}^{K} P_{IQ}((I, Q)|S_k) \cdot P_T(k|i, t_d) \tag{4}$$

P_{BG} captures the bursty Gaussian behavior of the signal, which is observed in Figure 5. In the following, we exploit this model to extract the signal states and trace the state transitions of the signal.

4.2 Extraction of combined states

As discussed in Section 3.2, signal clusters in the IQ domain are not always distinguishable from each other. Therefore, to trace signal's transitions among different states, the first challenge we meet is to find the signal clusters, which represent different signal states. We divide this process into two phases. Specifically, Hubble first coarsely clusters the samples using the density-based clustering method, i.e., DBSCAN [3]. Very likely, some overlapped clusters may be grouped into a supercluster in this phase. Then, in the second step, Hubble decomposes those superclusters one by one in the descending order of their cardinality (the number of samples assigned into the cluster), until a cluster contains only one center is detected.

In this section, we focus on the second phase, exploring how to reliably identify signal centers in a supercluster by leveraging the temporal characteristics of the signal samples.

4.2.1 Temporal distribution of the signal. For identifying the centers in a supercluster, the most thorny problem we meet is how to distinguish between the central area of a cluster and an overlapping area. Experimental results in Section 3.3 tell that although these two areas exhibit similar IQ domain representation (i.e., density), the occurrence of the signal samples in these two areas exhibit different levels of

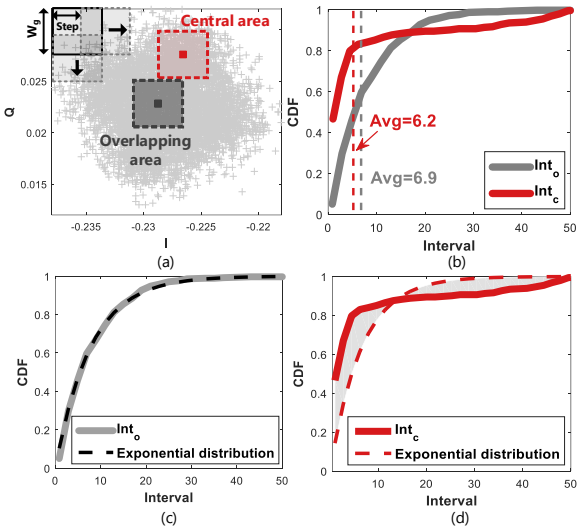

Figure 8: Signal movement in the overlapping area and the central area: (a) collided signal; (b) comparing the distribution of $\mathbf{Int_c}$ and $\mathbf{Int_o}$; (c) distribution of $\mathbf{Int_o}$; (d) distribution of $\mathbf{Int_c}$.

burstiness/randomness. As we have discussed in Section 4.1 (in Equation (4)), for a collided signal, the probability that the i-th sample occurs at a location (I, Q), namely $P_{BG}((I, Q)|i)$, is determined by two factors: i) the state of sample i (which is captured by P_T); and ii) the signal's IQ domain distribution (i.e., the noise level, which is captured by P_{IQ}).

On one hand, for a location (I, Q) which is close to a cluster center that represents a state S_k, $P_{IQ}((I, Q)|S_k)$ is high only when the signal is at state S_k (i.e., $s_i = S_k$), and is particularly low otherwise. That's to say, the sample's occurrence probability at (I, Q) is highly determined by $P_T(i, k)$, which captures the probability that sample i is at state S_k. The expression of P_T shown in Equation (2) tells that a signal's state exhibits strong burstiness in time domain. This indicates that $P_{BG}((I, Q)|i)$ exhibits strong burstiness if (I, Q) is close to a cluster center.

On the other hand, if (I, Q) is in an overlapping area, $P_{IQ}((I, Q)|S_k)$ exhibits no significant skewness for different S_k. Therefore, in this case, $P_{BG}((I, Q)|i)$ is not so sensitive to the state of the sample i (i.e., $P_T(i, k)$), exhibiting less burstiness and higher randomness.

We use the supercluster shown in Figure 4(c) as an example to verify the above assumption. Specifically, we extract the signal samples in a central area and an overlapped area of the supercluster (as shown in Figure 8(a)), and denote these two groups of samples by $\mathbf{S_c}$ and $\mathbf{S_o}$, respectively. Then we calculate the occurrence interval of samples in these two areas (denoted as $\mathbf{Int_c}$ and $\mathbf{Int_o}$). Figure 8(b) compares the distribution of $\mathbf{Int_c}$ and $\mathbf{Int_o}$.

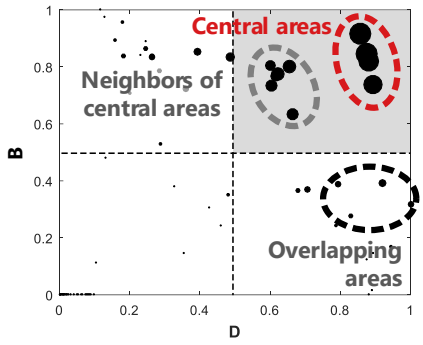

Figure 9: Center identification based on $D(g)$ and $B(g)$.

We can see that, although the average values of $\mathbf{Int_c}$ and $\mathbf{Int_o}$ are similar (which indicates similar densities of the two areas), their distributions are quite different. Specifically, the distribution of $\mathbf{Int_o}$ properly fits the exponential distribution, as shown in Figure 8(c). This is caused by the signal's random occurrence in the overlapping area. However, distribution of $\mathbf{Int_o}$ deviates seriously from the exponential distribution, as shown in Figure 8(d). This is due to the signal's bursty occurrence in the central area.

4.2.2 Identification in principle. The above observation motivates us to combine the distribution characteristics of the signal in both time and IQ domains as a metric to assess which areas are central areas. Specifically, we search in the supercluster with a moving grid (as shown in Figure 8(a)), where the size of the grid is W and the moving step is $\frac{W}{2}$. W can be adapted according to the noise level. For each location g, we denote the samples in the grid by $\mathbf{S_g}$, and the occurrence intervals between those samples by $\mathbf{Int_g}$. We calculate two metrics of each location g: i) the number of samples in g (i.e., the sampling density, denoted by $D(g)$); and ii) burstiness of samples' occurrence in g (denoted by $B(g)$).

Here, burstiness is quantified through Pearson's Chi-Square statistical test, by computing the deviation of the observed distribution of $\mathbf{Int_g}$ from the exponential distribution. A higher deviation level indicates a higher probability that g is a central area. Specifically, suppose there are n possible values in $\mathbf{Int_g}$, and the observed occurrence probability of a certain value Int_j $(1 \le j \le n)$ is denote by O_j. If we denote the mean of the values in $\mathbf{Int_g}$ by Int_{ave}, then the value

$$B(g) = \chi^2 = \sum_{j=0}^{n-1} \frac{(O_j - f(Int_j))^2}{E_j}. \quad (5)$$

is a measurement of the deviation level, which further quantifies the burstiness of samples in g. Here, $f(\bullet)$ is the probability density function of the exponential distribution.

Figure 9 shows the normalized density and burstiness of each grid g. We can see that only the central areas exhibit both high density and high burstiness, which makes them

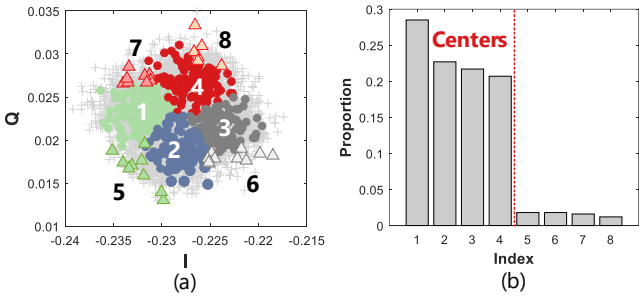

Figure 10: Results of the probing process: (a) the result observed at the IQ domain; (b) proportion of the training samples that assigned to different candidates.

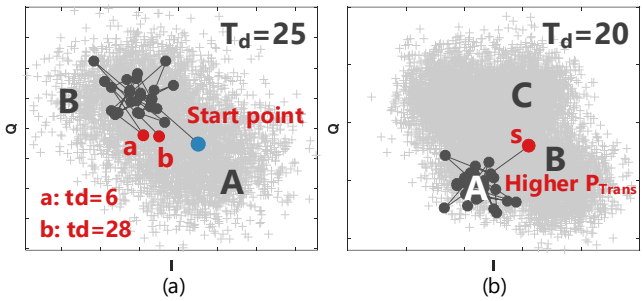

Figure 11: Assigning a sample that located in the overlapped area: (a) two-cluster-overlapping case; (b) four-cluster-overlapped case.

distinct from others. Then we propose a metric to quantify how likely a grid g is a central area:

$$P_c(g) = \beta \cdot B(g) \cdot D(g). \tag{6}$$

where β serves as a normalization constant.

4.2.3 Identification in practice. A challenge we meet in practice is that the reader doesn't know the number of clusters. A naive method for center identification is to use a threshold: the grids whose P_c exceed the threshold will be identified as cluster centers. However, this is not a general solution because we cannot find a fixed threshold for all the collision cases. In the design of Hubble, instead of judging directly based on P_c, we propose to recognize the cluster centers through a *probing process*. Specifically, considering that central areas usually exhibit both higher density and higher burstiness than other areas, we consider the centroids of the grids, whose $(D(g), B(g))$ is at the top right corner of Figure 9 (e.g., $D(g) > 0.5$ and $B(g) > 0.5$ in our implementation), as the candidates of cluster centers. Then we treat the first 5% samples as training samples and assign them to the candidates using the method proposed in Section 4.3 (where P_c serves as the weight of each candidate). The result of the probing process can indeed help to recognize the cluster centers.

Here, we explain why most samples are assigned correctly to the real cluster centers in the probing process. We divides the candidate points that are not the cluster centers to two groups, denoted as *halo points* (which locates adjacent to the cluster centers), and *overlapping points* (which locates on the overlapping area). For a halo point, since its location is similar to the corresponding cluster center, its P_{IQ} will be very similar with that of the cluster center for each sample i. Consider that P_c of a halo point is lower than that of a cluster center, more samples will be assigned to the cluster center. For a mid point, since they are typically located on the overlapping area, where the samples exhibit poor temporal consistency. This leads to low P_T for assigning a sample to the the mid points.

As an example, Figure 10(a) shows the probing result of the case in Figure 8. We find that most samples are assigned correctly. Therefore, if we denote the number of training samples that are assigned into g as $|g|$. A hint for selecting the cluster centers is provided by sorting $|g|$ in the decreasing order, as shown in Figure 10(b). Here, the anomalously decrease in the 5-th rank helps us to identify the top 4 candidates as the cluster centers.

4.3 Tracing of state transition

Now we have K cluster centers, each representing a signal state. The next task is to identify the underlying state of each sample i. Indeed, the state of sample i can be inferred based on two clues: i) the location of sample i; and ii) the signal's underlying transition pattern. We use two examples (as shown in Figures 11(a) and (b)) to illustrate why this would work.

In Figure 11(a), two clusters (A and B) overlap with each other, and we cannot identify the states of the samples located in the overlapping area (e.g., a and b). However, if we find that the average dwell time T_d is 25 samples in this case, and the signal has stayed at A for only 6 samples when a arrives, and for 28 samples when b arrives. Then we can easily infer that the signal signal is likely to flip when sample b arrives, and thus sample a probably belongs to cluster A, while sample b probably belongs to cluster B.

In some rare cases, assigning only based on the dwell time is not enough. As shown in Figure 11(b) where four clusters overlap with each other. Sample s is located in the overlapping area of B and C. Although we know that the signal is leaving cluster A based on the dwell time, but we cannot determine where the signal is transiting to. Fortunately, as discussed in [8], signal has distinct transition probability between different pairs of states. Therefore, we can solve this dilemma by comparing the transition probabilities between states A and B and states A and C.

Based on the above discussion, instead of assigning the samples in isolation, we should treat each sample in the

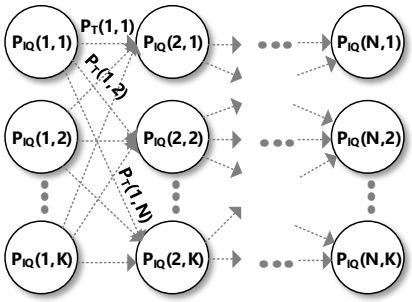

Figure 12: The markov model used for signal tracing.

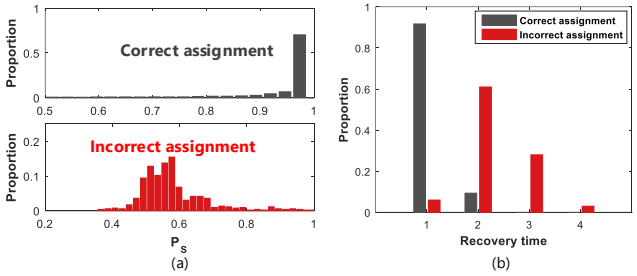

Figure 13: Error detection: (a) distribution of $P(i,k)$; (b) distribution of recovery time.

context of its predecessors in the time series, and identify the state of each sample based on a joint consideration of sample's location, probable dwell time at a state, and the transition probabilities between states. We propose a Markov-based model to solve the above problem, as shown in Figure 12. Specifically, we treat the combined state of the samples as hidden states of the markov model. Then P_{IQ}, which captures signal's gaussian property in the IQ domain, acts as the *emission probability* of markov model, and P_T, which captures the temporal burstiness of the signal acts as the *transition probability* of the markov model. Then the likehood for assigning sample i to state S_k is given by:

$$P(i, S_k) = P_w(S_k) \cdot P_{IQ}((I, Q)|S_k) \cdot P_T(i, k|t_d), \tag{7}$$

Here, $P_w(k)$ is the weight of each cluster, given by $\frac{P_c(k)}{\sum_{k=1}^{K} P_c(k)}$.

Given $P(i, S_k)$, solving the following optimization problem assigns the N received samples to the K states in a way that maximizes the likelihood of the assignments.

$$\arg\max_{\mathrm{C}} \sum_{i=1}^{N} P(i, k). \tag{8}$$

This problem can be solved by using Dynamic Programming in $O(KN)$ operations.

Model initialize. Initially, P_{IQ} and P_T are unknown, and we roughly initialize these two parameters as follows:

Initializing P_{IQ}: Equation (3) tells that P_{IQ} is related to two parameters: i) the location of cluster k, which is given in the cluster identification component; and ii) the covariance matrix Σ, which is considered highly related to the noise level. Assume that the background noise is relatively stable, Σ_k can be measured based on the clustering result of previous transmission tasks.

Initializing P_T: As shown in Equation (2), P_T is related to two parameters: P_d and P_{trans}. Specifically, P_d is initialized based on Equation (1), where $T_d = \frac{2M_R}{3N_T \cdot B_{ave}}$. Here, B_{ave} can be estimated based on the available bitrate configuration of the backscatter system, and N_T can be estimated based on the total number of tags (denoted by N_{all}), and the data rate of each tag (denoted by P_{send}):

$$N = \sum_{n=1}^{N} \binom{N_{all}}{n} \cdot P_{send}^{n} \cdot (1 - P_{send})^{N_{all}-n}. \tag{9}$$

Initializing P_{trans}: P_{trans} is initialized as equal for all the cluster pairs.

Model update. P_d and P_{trans} are also updated at runtime. It sounds simple: we just count the average dwell time of the signal and the number of transitions between different clusters periodically, based on the previously identified samples. However, an underlying problem is: errors in the previous assignments, even rarely occurs, may lead to inaccurate estimation results of P_d and P_{trans}. Fortunately, we find that most of the errors can be detected and corrected, which will be explained in Section 4.4.

4.4 Error Handling

Error detection. Hubble detects errors leveraging the reliability of each assignment i (i.e., $P(i, S_k)$). Specifically, we find that most incorrectly assigned samples are located in the overlapping area. This leads to a relatively low P_{IQ}, and therefore a low $P(i, S_k)$. To verify this assumption, we consider the case in Figure 4(c) again. Figure 13(a) shows the distribution of $P(i, S_k)$ for correct and incorrect assignments. As shown, $P(i, S_k)$ is actually a indicator of error.

Can we just identify an assignment with low $P(i, S_k)$ as an incorrect assignment[1]? The answer is no because a correct assignment may also have low $P(i, S_k)$. Fortunately, we find that compared with the incorrect assignments, correct assignments typically lead to short *recovery time* (i.e., the number of successive assignments that with low $P(i, S_k)$). Figure 13(b) shows the distribution of recovery time for correct and incorrect assignments. We can see that more than 90% of the correct assignments lead to a recovery time less than 2 samples. For the false assignments, about 92% of them lead to a recovery time that fall between 2 and 4 samples.

[1]we use a threshold, which is empirically set at $P(i, k) = 0.7$, to find the low reliable assignments

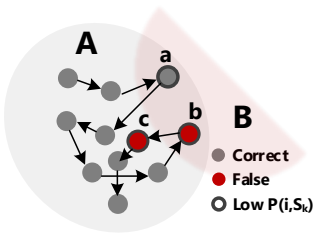

Figure 14: Different recovery time for correct and incorrect assignments.

We use an example shown in Figure 14 to explain the reason behind. Assume sample a is a correctly assigned sample (belongs to Cluster A) which exhibits low $P(a, A)$ since it is located in the overlapping area. As discussed in Section 4.2.1, signal samples that in the overlapping area usually exhibit low continuity. Therefore, it will probably leave the overlapping area soon, which leads to a high $P(\bullet)$ for the next assignment. For an incorrectly assigned sample b (which is assigned to Cluster B but indeed belongs to A), however, the assignment of the next sample c (which may also belong to A) will exhibit either a low P_{IQ} (if it is incorrectly assigned to B) or a low P_T (if it is correctly assigned A), both of which inevitably lead to a low $P(\bullet)$.

Based on the above discussion, we can identify an assignment sequence, which exhibits successive low $P(i, S_k)$, as a suspicious segment, and correct the errors using the method introduced as below.

Error correction. For error correction, we directly merge the samples in a suspicious segment to the cluster that the predecessor is assigned to (as shown in Figure 15(a)). In some rare cases, the suspicious segment accidently covers a bit boundary (as shown in Figures 15(b) and (c)). Therefore, the predecessor and the successor of the suspicious segment belong to different clusters (denoted as Clusters C_{pre} and C_{suc} respectively). In this case, we just merge the first $\frac{1}{2}$ samples in the suspicious segment to C_{pre} and the rest to C_{suc}. This may lead to some errors, but an incorrect merging in a cross-boundary case only leads to slight displacement of the bit boundary, which has little impact on the parameter estimation and decoding process.

There is still a problem: correct assignments can also occasionally lead to a long recovery time and therefore be incorrectly identified as a suspicious segment, as shown in Figure 13(b). This is caused by either the interference (which leads to successive low P_{loc}) or a too long dwell time of the signal (which leads to successive low PoF). Fortunately, as shown in Figures 15(d) and (e), the correction method will not lead to error for the normal case, and leads to only slight error in the cross-boundary case.

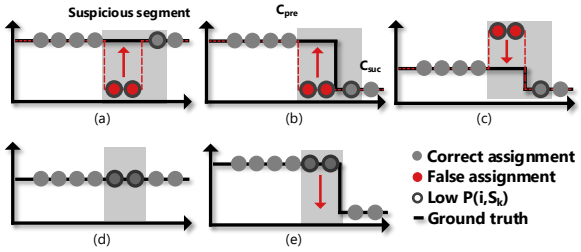

Figure 15: Error correction method.

5 DISCUSSION

Reducing the computation overhead. To reduce the computation overhead, instead of directly assigning each sample using Hubble, we may first divide the samples into grids[2] (which is much less than the samples) and cluster those grids using DBSCAN [3] (in the coarse clustering phase). Therefore, the computation overhead of this process is only $O(k_g \cdot logk_g)$, where k_g is the number of grids. Then Hubble separates the superclusters using the method introduced in Sections 4.2 and 4.3. The computation overhead of this process is $O(G + M \cdot (K + w))$, where M is the number of the samples in the superclusters.

Enabling partial decoding. In the cases where multiple clusters almost completely coincide with each other (which is caused by a particularly low signal SNR, like the case shown in Figure 2(b)), Hubble is unable to correctly find all the cluster centers. In such a scenario, Hubble tries to extract the strong signals (if exist) from the collision, and consider the weak signals as noise, achieving partial decoding. Take the collision signal depicted in Figure 2(b) as an example. The signal from Tag 3 is too weak, leading to overlapping between cluster HHL and HHH, and LLL and LLH. In this case, if Hubble is unable to separate the overlapped signal, it will ignore the signal from Tag 3 by merging Cluster HLL and HLH, and LHL and LHH, leaving only four clusters, which represent the combined states (i.e, LL, LH, HL, and HH) of tags 1 and 2. This ensures the collision signal of these two tags is correctly decoded.

After the center identification process, Hubble first check whether the identified number of centers is a power of 2. If not, Hubble will merge the cluster pairs one by one in the descending order of the distance of each cluster pair, until the number of clusters reaches the power of 2.

6 EVALUATION

6.1 Experiment Setting

We implement Hubble on the USRP N210. The state transition sequence identified by Hubble is further decoded by

[2]The grids used in this phase is different from that used in the center identification component.

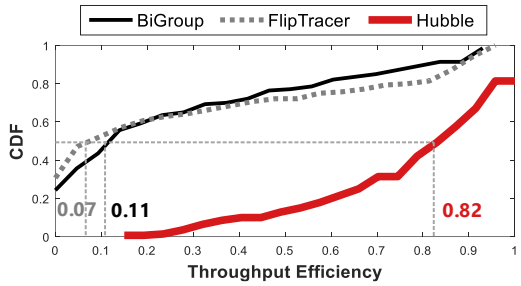

Figure 16: Overall performance.

FlipTracer in our implementation. The SDR reader is connect with two UBX RF daughterboards. The sampling rate of the reader is set at 20MHz.

The implementation of Hubble on WISP requires a slight modification to the EPCglobal C1G2 protocol. Specifically, we just remove the elements for Slot Aloha operation and thus the tags can respond concurrently. The default bitrate and packet length are set at 100 Kbps and 200 bits, respectively.

In the experiment, we compare the performance of Hubble with two schemes, both of which cluster the received signal using density-based clustering algorithm.

- **FlipTracer.** FlipTracer [8] assumes that signal's transitions among different combined states follows different probabilities. It therefore identifies the combined state of the cluster by counting the numbers of transitions among clusters.
- **BiGroup.** BiGroup [11] assumes signal transitions of each tag arrive with predictable interval. It extracts the sequence of each tag's signal transitions in the time domain, and connects the signal clusters in the IQ domain for parallel decoding.

Methodology: Our goal is to evaluate the performance of Hubble for supporting different number of concurrent tags, under different SNRs. The SNR is adjusted by changing the distance between the tag and the reader. Specifically, in the experiment, we change the number of concurrent tags from 2 to 6. For the case where n tags ($2 \leq n \leq 6$) transmit concurrently, we conduct n experiments. In the i-th ($1 \leq i \leq n$) experiment, we fix the locations of $n - i$ tags within a 0.5ft transmission range (the corresponding average SNR is 20dB), and move the other i tags (termed as *weak tags*) further and further away from the reader, until reaching the boundary for signal detection. For each location, the tags transmit packet concurrently for 100 rounds, and the reader separates and decodes the collided signal using different approaches.

6.2 Overall performance

First we discuss the aggregated results of the experiments described above. Figure 16 shows the CDF of the the *throughput efficiency* (defined as the ratio of throughput to aggregated PHY bit-rate) of Hubble, FlipTracer, and BiGroup. As shown in the figure, since FlipTracer and BiGroup work only when the signal strength is high, the median throughput efficiency of these two approaches are only 0.07 and 0.11, respectively. BiGroup performs slightly better than FlipTracer due to its ability to partially decode the received signal. This makes BiGroup more robust to low SNR scenarios, compared with FlipTracer. Hubble has a median throughput efficiency of 0.82, outperforming FlipTracer and BiGroup by 11.7× and 7.4×, respectively.

More detailed experiment results are given in Figure 17, which shows the throughput efficiency of Hubble, FlipTracer and BiGroup v.s. the SNR of the weak tags, under different numbers of concurrent tags and different proportions of weak tags. We can observe that all the above three factors seriously affect the performance of FlipTracer and BiGroup. In comparison, Hubble is not so sensitive to those factors. In the following sections, we access the performance of the three schemes under different influencing factors.

6.3 Impact of the signal SNR

As shown in Figure 17, the throughput efficiency of both FlipTracer and BiGroup decreases rapidly with SNR, especially when the SNR falls below 8dB. Such a trend becomes more significant when the number of tags increases from 2 to 6. In comparison, Hubble is much more robust to the low SNR. The throughput efficiency of Hubble keeps higher than FlipTracer and BiGroup no matter how many tags transmit concurrently. For example, when two tags transmit concurrently, Hubble achieves a 85% throughput efficiency even when the SNR of both the two tags is only about 4dB, while that for FlipTracer and BiGroup is only 0.01 and 0.03, respectively. Even when 6 tags transmit concurrently and some of the tags exhibit low SNR (where many signal clusters are completely overlapped), Hubble can still decode 15% packets.

A interesting observation is that for all the three approaches, the throughput does not strictly decrease with SNR. Sometimes the throughput slightly increases when the SNRs of the weak tags are particularly low. The reason is that in these scenarios, signals from the weak tags are too weak and can be considered as noises. This helps the receiver to extract the strong signals more accurately, enabling partial decoding.

6.4 Impact of the number of concurrent tags

Figure 18 shows the averaged throughput efficiency under different numbers of tags. As shown in the figure, the throughput of FlipTracer and BiGroup decreases significantly with the number of concurrent tags. The impact of the tag number is more apparent than that of the SNR because the

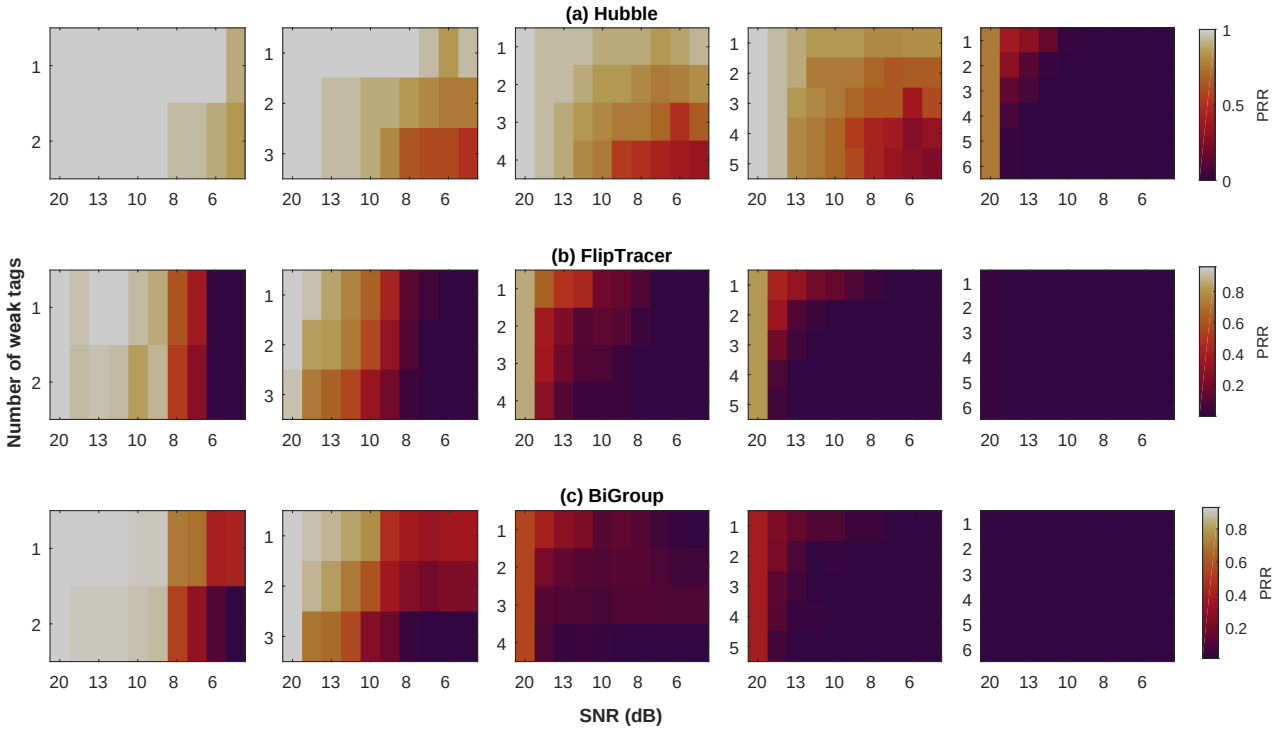

Figure 17: Performance of different approaches under different SNR and different numbers of colliding tags.

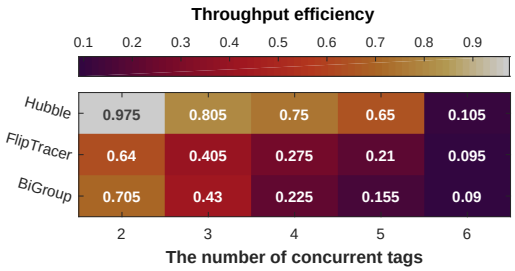

Figure 18: Averaged throughput efficiency under different numbers of tags.

exponentially increased number of clusters (from 2 to 64) steeply increases the probability of cluster overlapping. Compared to FlipTracer and BiGroup, the performance of Hubble is not so sensitive to the increase in the number of tags. Specifically, consider the scenario where 6 tags transmit concurrently (as shown in Figure 17), the throughput of FlipTracer and BiGroup reach to 0 even when the SNRs of all the tags are particularly high. In comparison, Hubble still achieves a 75% maximum throughput efficiency in this case. Another observation from Figure 17 is that the performance of Hubble becomes more sensitive to the SNR when there are more tags transmit concurrently.

Here we have to emphasize that the decoding capacity of Hubble is sufficient for most applications. Consider a scenario where 500 tags[3] co-exist in the communication range of the reader, which follow the EPC C1G2 standard. The on-air transmission time of a packet is short, usually a few milliseconds. Assume that the on-air time of one packet is 2ms (e.g., when the packet length is 200 bit and the bitrate is 100Kbps) and the tags transmit packets at a rate of 1 packet per second, the probability that more than 6 tags transmit packet concurrently is $1 - \sum_{k=0}^{6} C_{500}^{k} \cdot (\frac{1}{200})^{k} \cdot (1 - \frac{1}{200})^{500-k} = 0.0328$.

6.5 Impact of the proportion of weak tags

The number of weak tags is also an important influencing factor. Consider the scenario where 5 tags transmit concurrently, and there is only one weak tag, Hubble ensures more than 80% throughput efficiency even when the SNR of the weak tag is 6.6dB. When all the five tags are moved far away from the reader, the throughput efficiency steeply decreases to only 15%. Another important observation is, compared with FlipTracer, performance of BiGroup is more sensitive to the number of weak tags. The reason is that BiGroup's ability to partially decode the collided signal is weaken when more weak tags coexist.

[3]Since the communication range of the reader is only a few meters in practice [10], 500 tags indicates an extremely dense deployment.

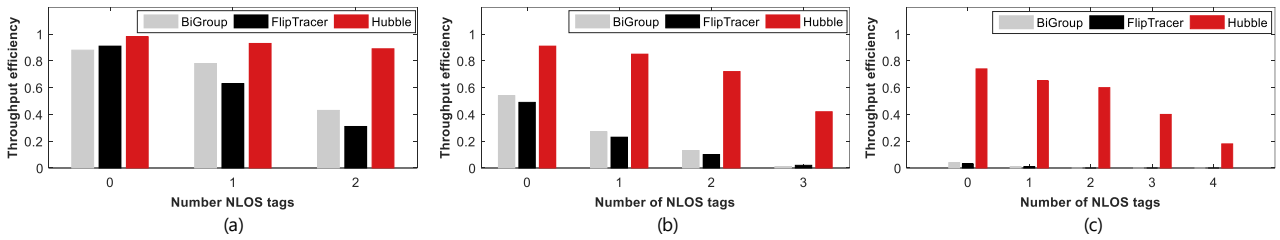

Figure 19: Performance for NLOS transmission: (a) two tags; (b) three tags; (c) four tags

6.6 Transmission across obstacle

In this section, we evaluate the performance of Hubble in a more challenging condition - the NLOS (non-line-of-sight) scenario - to further investigate the reliability of Hubble. In the experiment, a human stands 1ft away from the reader. We have two test positions: a LOS (line-of-sight) position and a NLOS position, both are located 2ft away from the reader. We change the number of concurrent tags from 2 to 4[4]. For the case where n tags transmit concurrently, we conduct $n+1$ experiments. Specifically, in the i-th experiment ($0 \leq i \leq n$), i tags are located at the NLOS position and the other $n-i$ tags are located at the LOS position. The tag located at the NLOS position is termed as a *NLOS* tag.

The results are shown in Figures 19. As shown, the presence of an obstacle decreases the throughput of all the three approaches. The influence is more obvious when there are more tags transmit concurrently. For Hubble, when there are only two tags transmit concurrently, the performance degradation caused by the presence of the obstacle is only 5% $\sim$ 9%. When 5 tags transmit concurrently, the performance degradation becomes much more obvious, reaching to 75% when all the tags are located at the NLOS location.

FlipTracer and BiGroup are much more vulnerable to the NLOS scenario, compared with Hubble. As shown in Figure 19(a), even when there are only two concurrent tags, the presence of an obstacle can lead to 51% and 65% performance degradation for BiGroup and FlipTracer, respectively. That is to say, the existing parallel decoding approaches do not work in highly cluttered environments such as warehouses, stores, and factories.

6.7 Impacts of practical factors

We now evaluate the impact of other practical factors.

Bitrate. One of the underlying assumption of Hubble is that the signal will dwell on a certain state before transit to another state. However, when tags transmit with a particularly high bitrate, signal transitions are stacked back to back, leading to short dwell time at each signal state, which may affect the performance of Hubble. So we conduct experiments

[4]This is because FlipTracer and BiGroup cannot support more than 4 tags in practical scenarios

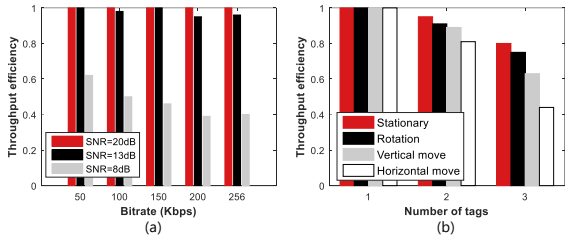

Figure 20: Impacts of practical factors: (a) bitrate; (b) dynamic environment.

to test the performance of Hubble under different bitrates. In the experiment, we use three tags. We vary the bitrate of the tags (from 100Kbps to 256Kbps) and the distance between the tag and the reader (which makes the SNR decreases from 20dB to 10dB). Note that a 256Kbps bitrate is the maximum bitrate that the WISP platforms can support. The result is shown in Figure 20(a). We can see that, compared with SNR, bitrate is not a key influencing factor of Hubble. Hubble can still achieves a decent performance at 256Kbps.

Dynamic environment. We further investigate the impact of dynamic working conditions. Three cases are compared: i) the location of the tags are fixed, but orientations of the tags keep changing; ii) the tags are moved towards and away from the reader (horizontal movement); and iii) the tags are moved in parallel to the reader (vertical movement). The experimental result is shown in Figure 20(b).

In case i, the performance of Hubble is very close to the stationary case, indicating that the impact of orientation changing is negligible. In comparison, the impact of the actual moving is more promise, especially in the case of horizontal moving, due to the changes in the SNR. This impact becomes more obvious when more tags transmit concurrently.

6.8 Real-world experiment

To better demonstrate the efficacy and efficiency of Hubble, we conduct real-world experiments to evaluate its performance in the wild. In the experiments, 8 tags are located within the communication range of the reader but at different distances to the reader. They transmit packets periodically, and the collision occurs randomly due to the different transmission schedules of the tags. Specifically, the packet

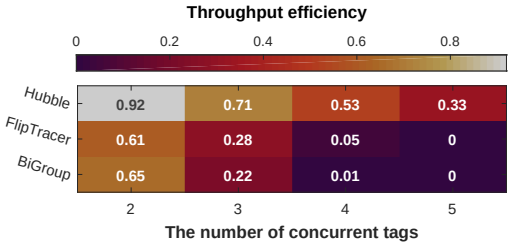

Figure 21: Performance in the wild.

interval is set at 20 ∼ 30 ms in this experiment to make the collision occur randomly, different tags transmit packets with different packet intervals.

Figure 21 shows the throughout efficiency of different approaches under different degrees of parallelism. We can observe slight performance degradation compared with that achieved in the controlled experiment. The degradation is caused by *partial collision* between two (or more) packets. Specifically, since the tags transmit packets blindly, the response offset among tags are totally random and uncontrolled. When the overlapped segment between two colliding packets is too short, the collected samples of the collided segment is insufficient for successful parallel decoding. Fortunately, this problem will not occur in the EPC protocol where the reader's QUERY synchronizes the tags' responses. Another important observation is that the in-the-wild performance of Hubble is still much better than FlipTracer and BiGroup, especially when the degree of parallelism increases.

6.9 Trace-driven Simulation

We perform the trace-driven simulations to investigate the performance gain of Hubble with a larger number of tags. In the simulation, N tags are randomly deployed in a $2R \times 2R$ area, where the reader is located at the center of the area. The reader queries the tags and the tags respond to the reader following the EPC standard. The number of slots (denoted as N_s) is adjusted by a parameter Q (i.e., $N_s = 2^Q$). In an arbitrary collision slot, the reader decodes the received signal using Hubble, FlipTracer and BiGroup.

Figure 22 shows the performance gain of Hubble compared with FlipTracer and BiGroup, under different N and R. A lighter color means a higher performance gain. Here, R is normalized as the factor of the signal detection range. Q is set at $\lfloor log_2^N \rfloor$, which makes the number of slots approximately equal to the number of tags.

The result tells that i) the performance gain increases significantly with R; ii) the number of tags has no obvious influence on the performance gain; iii) the maximum gains that Hubble achieves, compared with FlipTracer and BiGroup, is 3.9× and 3.7×, respectively.

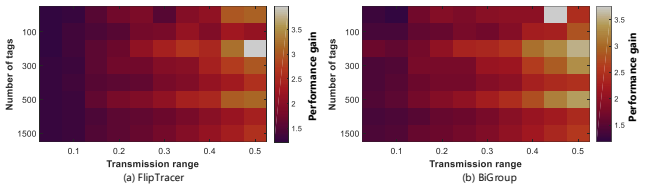

Figure 22: Performance gain when $N_s = N$.

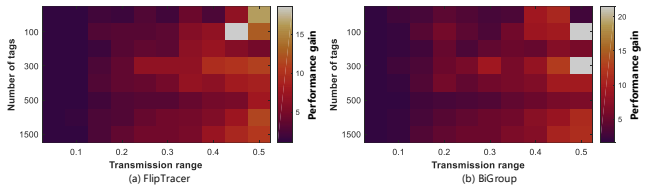

Figure 23: Performance gain when $N_s = N/3$.

We further evaluate the performance gain of Hubble under higher degree of parallelism, by decreasing Q to $\lfloor log_2^{\frac{N}{3}} \rfloor$ (which means smaller number of slots). The results are shown in Figure 23. By comparing Figures 22 and 23, we find that the performance gain is much higher than that achieved when $Q = \lfloor log_2^N \rfloor$. Specifically, the maximum gains compared with FlipTracer and BiGroup, increases to 19× and 21×, respectively.

7 CONCLUSIONS

This paper presents Hubble, a signal processing approach to reliably trace signal state transitions, based on the bursty Gaussian behavior of the signal. With Hubble, the potential of parallel backscatter [8] is realised, achieving a 11.7× improvement in the median throughput.

ACKNOWLEDGMENTS

We are grateful to the professional comments and kind help from the anonymous reviewers and shepherd. The first author of this paper Meng Jin was a PhD student of Northwest University of China when she visited Tsinghua University to carry out research on this work. She is now a PostDoc researcher in the School of Software and BNRist, Tsinghua University.

This work was supported by National Key R&D Program of China No.2017YFB1003000 and 2013BAK01B02, National Natural Science Foundation of China (61772306, 61772422, 61672428, 61572402), State Grid of China Research Fund, The Science and Technology Innovation Team Supported Project of Shaanxi Province 2018TD-O26.

REFERENCES

[1] Christoph Angerer, Robert Langwieser, and Markus Rupp. 2010. RFID Reader Receivers for Physical Layer Collision Recovery. *IEEE Transactions on Communications* 58, 12 (2010), 3526–3537.

[2] Aggelos Bletsas, John Kimionis, Antonis G. Dimitriou, and George N. Karystinos. 2012. Single-Antenna Coherent Detection of Collided FM0 RFID Signals. *IEEE Transactions on Communications* 60, 3 (2012), 756–766.

[3] Martin Ester, Hans-Peter Kriegel, Joerg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of IEEE KDD '96*.

[4] Wei Gong, Haoxiang Liu, Kebin Liu, Qiang Ma, and Yunhao Liu. 2016. Exploiting Channel Diversity for Rate Adaptation in Backscatter Communication Networks. In *Proc. of IEEE INFOCOM '16*.

[5] Pan Hu, Pengyu Zhang, and Deepak Ganesan. 2014. Leveraging Interleaved Signal Edges for Concurrent Backscatter. In *Proc. of ACM HotWireless '14*.

[6] Pan Hu, Pengyu Zhang, and Deepak Ganesan. 2015. Laissez-Faire: Fully Asymmetric Backscatter Communication. In *Proc. of ACM SIGCOMM '15*.

[7] EPCglobal Inc. 2008. EPCglobal Class 1 Generation 2 V. 1.2.0. http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2.

[8] Meng Jin, Yuan He, Xin Meng, Yilun Zheng, Dingyi Fang, and Xiaojiang Chen. 2017. FlipTracer: Practical Parallel Decoding for Backscatter Communication. In *Proc. of ACM MobiCom '17*.

[9] RS Khasgiwale, RU Adyanthaya, and DW Engels. 2009. Extracting Information from Tag Collisions. In *Proc. of RFID '09*.

[10] Yunfei Ma, Nicholas Selby, and Fadel Adib. 2017. Drone Relays for Battery-Free Networks. In *Proc. of ACM SigComm '17*.

[11] Jiajue Ou, Mo Li, and Yuanqing Zheng. 2015. Come and Be Served: Parallel Decoding for COTS RFID Tags. In *Proc. of ACM MobiCom '15*.

[12] Longfei Shangguan, Zhenjiang Li, Zheng Yang, Mo Li, and Yunhao Liu. 2013. OTrack: Order Tracking for Luggage in Mobile RFID Systems. In *Proc. of IEEE INFOCOM '13*.

[13] Longfei Shangguan, Zheng Yang, Alex. Liu, Zimu Zhou, and Yunhao Liu. 2015. Relative Localization of RFID Tags using Spatial-temporal Phase Profiling. In *Proc. of USENIX NSDI '15*.

[14] Dawei Shen, Grace Woo, David P. Reed, Andrew B. Lippman, and Junyu Wang. 2009. Efficient and Reliable Low-Power Backscatter Networks. In *Proc. of RFID '09*.

[15] Jue Wang, Haitham Hassanieh, Dina Katabi, and Piotr Indyk. 2012. Efficient and Reliable Low-Power Backscatter Networks. In *Proc. of ACM SIGCOMM '14*.

[16] Jue Wang and Dina Katabi. 2013. Dude, Where's My Card? RFID Positioning That Works with Multipath and Non-Line of Sight. In *Proc. of ACM SIGCOMM '13*.

[17] Teng Wei and Xinyu Zhang. 2016. Gyro in the Air: Tracking 3D Orientation of Batteryless Internet-of-Things. In *Proc. of ACM MobiCom '16*.

[18] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A Fine-Grained Indoor Location System. In *Proc. of USENIX NSDI '13*.

[19] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using Cots Devices. In *Proc. of ACM MobiCom '14*.

[20] Lei Yang, Qiongzheng Lin, Xiang-Yang Li, Tianci Liu, and Yunhao Liu. 2015. See Through Walls with COTS RFID System!. In *Proc. of ACM MobiCom '15*.

[21] Pengyu Zhang, Jeremy Gummeson, and Deepak Ganesan. 2012. BLINK: A High Throughput Link Layer for Backscatter Communication. In *Proc. of ACM MobiSys '12*.

[22] Yuanqing Zheng and Mo Li. 2013. ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems. In *Proc. of IEEE INFOCOM '13*.

[23] Yuanqing Zheng and Mo Li. 2014. Read Bulk Data from Computational RFIDs. In *Proc. of IEEE INFOCOM '14*.