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

Meng Jin, *Member, IEEE, ACM*, Yuan He, *Senior Member, IEEE, Member, ACM*, Xin Meng, Dingyi Fang, *Member, IEEE, ACM*, and Xiaojiang Chen, *Member, IEEE, ACM*

***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.**

***Index Terms*— Backscatter, wireless, parallel transmission.**

I. 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 [1]–[9]. As the IoT technology proliferates in areas like logistics, warehouses, manufacture, and retail, 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 [10]–[14].

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 [10], 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 [15]–[18] 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, the RSS at the reader is relatively weak, which means 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 and more tags join the parallel backscatter, the chance of superclustering also dramatically increases, which further decreases the decoding rate (the percentage of packets successfully decoded).

In order to the address the above problem, we in this paper propose Hubble, a reliable signal processing approach to

Manuscript received April 22, 2019; revised July 17, 2020; accepted September 2, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Jamieson. Date of publication October 21, 2020; date of current version February 17, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1003000; in part by the National Natural Science Foundation of China under Grant 61772306, Grant 61772422, Grant 61672428, and Grant 61902213; in part by the Research and Development Project of Key Core Technology and Generic Technology in Shanxi Province under Grant 2020XXX007; and in part by the Science and Technology Innovation Team Supported Project of Shaanxi Province under Grant 2018TD-O26. *(Corresponding author: Yuan He.)*

Meng Jin and Yuan He are with the School of Software, Tsinghua University, Beijing 100084, China, and also with BNRist, Tsinghua University, Beijing 100084, China (e-mail: he@greenorbs.com).

Xin Meng is with the School of Information Science and Technology, Northwest University, Xi'an 710069, China.

Dingyi Fang is with the Shaanxi International Joint Research Center for the Battery-free Internet of Things, Northwest University, Xi'an 710069, China, and also with the Internet of Things Research Center, Northwestern University, Evanston, IL 60208 USA.

Xiaojiang Chen is with the Shaanxi International Joint Research Center for the Battery-free Internet of Things, Northwest University, Xi'an 710069, China, and also with the Jingdong Wisdom Cloud Joint Research Center for AI and IoT, Northwest University, Xi'an 710069, China.

Digital Object Identifier 10.1109/TNET.2020.3027735

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 (e.g. the reader), 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 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 $11.7\times$ higher than the state-of-the-art approachs.

The rest of this paper is organized as follows. Section II discusses the related work. Section III presents the background knowledge and the motivation of this work. In Section IV we elaborate on the design of Hubble. After discussion on several important issues in Section V, we present the evaluation results in Sections VII. Section VIII concludes this work.

II. Related Work

The central task of parallel decoding for backscatter communication is to identify the state of every collided signal. Depending on how to distinguish different signal states, the existing works can be classified into three categories.

Decoding based on IQ domain information. Early works [19]–[23] 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 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 [15]–[17]. 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 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 the works in this category is that they don't rely on any prior knowledge of channel coefficients or linear dependency among collision states. The limitation is that they require relatively precise timing, i.e. low drifting rate (less than 200 ppm [17]) 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 [17], [18]. The intrinsic clock drifts cause degradation of decoding rate in practice.

Decoding based on the law of state transitions. A recent work FlipTracer [18] proposes to identify transitions of signal states 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 comparing the traced state transitions with 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 find that the key factor that limits the applicability of parallel decoding is the requirement on signal quality. That means the tags have to be located very close to the reader without obstacles. This is not the case in most IoT applications with backscatter devices. How to achieve reliable decoding of parallel backscatter in the wild? This is a problem with great significance in both research and application fields. Hubble is the first work to address this problem.

III. Background and Motivation

A. 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 $S = [s_1, s_2, \ldots, s_N]$, 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

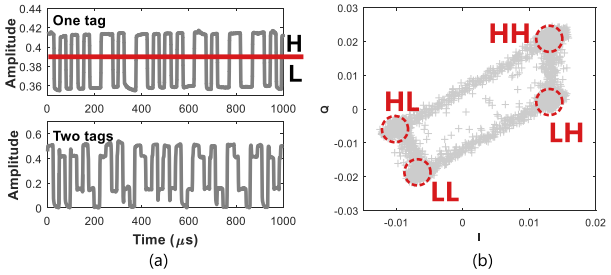


Fig. 1. Signals from the tags: (a) time domain signals from a single tag (top) and two concurrent tags (bottom). (b) IQ domain signals.

[15]–[18]. 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 [15]–[18]. Then, tracing signals' state transitions among the clusters tells the transmitted bits of each tag. The above is the basic mechanism of parallel decoding.

B. Challenges

A fundamental problem underlying the existing parallel decoding method is that signal clusters in the IQ domain are not always distinguishable from each other. In practice, the noise level in the environment determines the radius of a cluster, while the RSS of the signals determines inter-cluster distances. When N tags transmit concurrently, the signal RSS of each tag can affect the distances between 2^{N-1} pairs of clusters. For example, in Figure 1(b), distances between clusters LS and HS (here $S = H$ or L) are affected by the signal power of Tag 1. Distances between SL and SH are affected by the signal power of Tag 2. So in this case, high signal strength of both tags are required to make the four clusters distinguishable from each other.

We use $P(RSS_i)$ to denote the probability that the two states of a single tag i are distinguishable under the signal strength RSS_i. Clearly, a higher signal strength RSS_i leads to a higher $P(RSS_i)$. To decode the collided signal, the existing parallel decoding methods require that *each two clusters* are distinguishable. In a N-tag-collision case, for each tag i, its signal strength RSS_i determines the distance between $2^(N-1)$ pairs of clusters. So, the probability that each two clusters are distinguishable is expressed as:

$$P_{suc} = \prod_{i=1}^{N} P(RSS_i)^{2^{N-1}}. \quad (1)$$

Equation (1) indicates that both the decrease in RSS and the increase in tag number lead to steep decrease in decoding rate. To verify this point, we in Figure 2 shows the collided signal under different tag numbers and RSS. Figure 2(a) shows the case where three tags are located only 1 ft away from the reader. We can observe 8 clearly separated clusters in this case. When we move one of the tags to 3 ft away from the reader, the decrease in the tag's RSS leads to cluster overlapping, forming *superclusters*, as shown in Figure 2(b). The receiver even cannot figure out how many signal states there are, not to

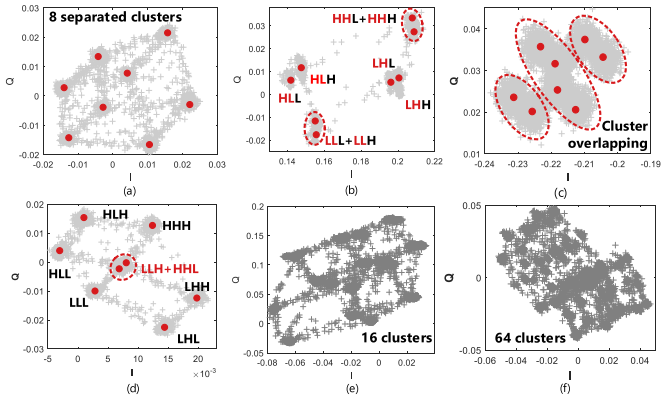


Fig. 2. Examples of the collided signals: (a) three strong tags; (b) coexistence of strong and weak tags. (c) three weak tags; (d) coincidental overlapping; (e) four strong tags; (f) six strong tags.

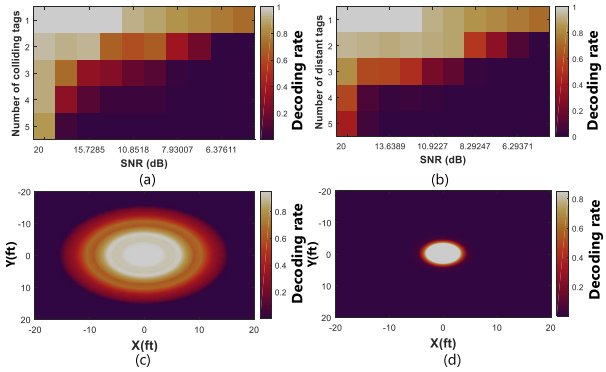


Fig. 3. Performance of existing methods: (a)-(b) Performance of FlipTracer and BiGroup under different SNRs; (c)-(d) Distribution of FlipTracer's decoding rate when 2 and 5 tags transmit in parallel.

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 extracted from the IQ domain.

It is worth noticing that low RSS is 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 signal 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 4 and 6 tags collide, 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 [18] and BiGroup [15] 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. Figures 3(a) and (b) respectively show the decoding rates (the percentage of packets successfully decoded) of the two approaches. 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. The figures tell that the theoretical results claimed by the existing approaches

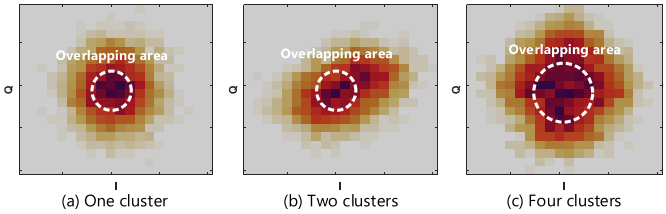


Fig. 4. Density distribution of collided signal.

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.

In order to provide a conceptual understanding of how the superclustering problem affects the coverage of the existing parallel decoding method, we in Figures 3(c) and (d) combine the experimental result in Figures 3(c) and (d) and the signal attenuation model to calculate the distribution of FlipTracer's decoding rate when 2 and 5 tags transmit in parallel. Specifically, Figures 3(c) and (d) plots the result 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.

Clearly, 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 [18]) may help to resolve the superclustering problem. As a counter-example, 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 the central cluster areas, because the overlapped area gathers signals from multiple clusters. So, only utilizing the IQ domain information is no longer effective in identifying different clusters.

C. 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. So, 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).

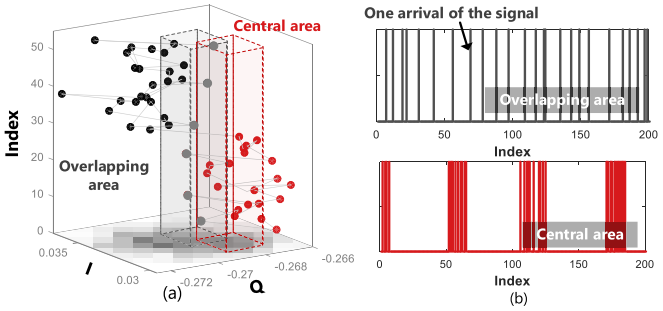


Fig. 5. (a) The bursty Gaussian behavior of signal; (b) The arrival of signals that confined in overlapping area and central area.

We have two important findings from the above 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 traceable 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.

IV. Hubble Design

Hubble is designed to extract the underlying state sequence from the seriously jumbled collision signal based on the bursty Gaussian behavior of the signal. Figure 6 summarizes the sketch of Hubble. Specifically, Hubble first identifies the locations of the cluster centers 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. With the identified state transition sequence, the collided signal can be decoded using existing parallel decoding methods [15], [18].

In the following of this section, we first propose the interstellar travelling model which captures signal's bursty Gaussian behavior, followed by the design details of Hubble.

A. 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. Since the reader can oversample the received signal, signal will dwell on a

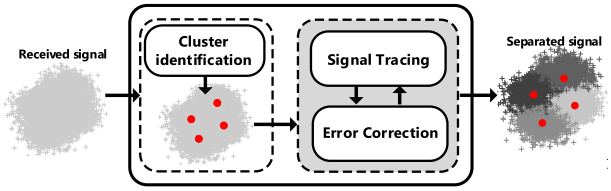


Fig. 6. Workflow of Hubble.

state before transiting to the other state. Specifically, 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 $\frac{M_R}{B_T}$ and $\frac{M_R}{2 \cdot B_T}$ samples' length, when transmitting bit 0 and bit 1, respectively. So, when one tag is transmitting, signal's dwell time is averaged at $T_d = \frac{3M_R}{2 \cdot B_T}$.

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 $S_1, \ldots, S_K$. Since the flips of different tags usually interleave each other, signal's average dwell time can be expressed as:

$$T_d = \frac{2M_R}{3\sum_{i=1}^{N_T} B_i} \tag{2}$$

We model the signal's dwell time as following the exponential distribution. Specifically, the exponential distribution describes the time interval between events which occur randomly at a stable average rate. There are two main properties of the collided signal that make the dwell time follows the exponential distribution:

1) The average dwell time is constant. As shown in Eq (2), the average dwell time of a signal is solely determined by the number of tags (N_T) and the bitrate of each tag (B_i). Although different tags have different clock frequencies and drift times, the average bitrate of a tag B_i (although deviates from the claimed value) is constant in a collision case. That is to say, the average dwell time is stable.
2) The dwell time of the collided signal is highly random and unpredictable. First, due to the difference in tags' power charging rate, different tags have different response delays. This leads to different and random initial offsets of the tags' signals. Second, due to the high drifting rate of the tags' clock, the bit durations of the tags neither identical among tags nor stable over time. The randomness in response delays and bit durations of the tags makes the dwell time of the collided signal highly random and unpredictable.

Based on the above discussion, the probability that a signal will dwell on a state for t_d samples can be expressed as:

$$P_d(t_d) = \frac{1}{T_d} \cdot e^{-\frac{1}{T_d} \cdot t_d} \tag{3}$$

We further conduct a set of experiments to verify our assumption on the exponential distribution. In the experiment, we let 4∼6 WISP 4.0 tags transmit concurrently. Bitrate of the tags changes from 50 Kbps to 250 Kbps. Figures 7 (a)-(c) show the observed distribution of t_d and the fitted exponential distribution under different bitrates with different numbers of tags. We find that the dwell time of the signal properly fits

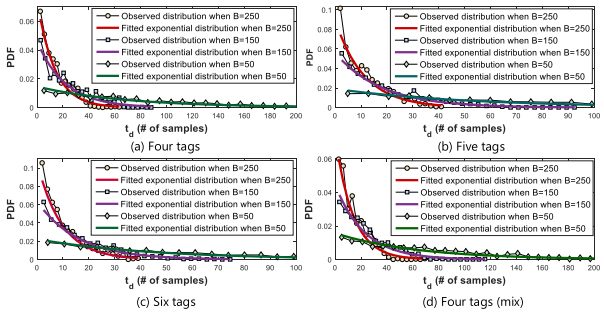


Fig. 7. Distribution of signal's dwell time.

the exponential distribution. We further conduct experiments to observe the distribution of t_d when tags with different model transmit concurrently. Figure 7 (d) shows the result when there are two WISP 4.0 tags and two WISP 5.0 tags. Clearly, the distribution of t_d will not be affected by a tag's model.

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:

$$P_T(i, S_k|t_d) = \begin{cases} P_d(t_d), & s_{i-1} = S_k \\ (1 - P_d(t_d))P_{trans}(S_q, S_k), & s_{i-1} = S_q \end{cases} \tag{4}$$

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

$$P_{IQ}((I, Q)|S_k) = \pi(I, Q, I_k, Q_k, \mathbf{\Sigma}) \tag{5}$$

where $\pi(\bullet)$ is the probability density of the Gaussian distribution. $\mathbf{\Sigma}$ 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):

$$P_{BG}((I, Q)|i) = \sum_{k=1}^{K} P_{IQ}((I, Q)|S_k) \cdot P_T(k|i, t_d) \tag{6}$$

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

B. Extraction of Combined States

The key in state extraction is to find the signal clusters, which represent different signal states. We divide this process into two phases. In the first step, Hubble coarsely clusters the samples using the density-based clustering method [24]. Very likely, some overlapped clusters may be grouped into a supercluster in this step. 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 step, exploring how to reliably identify cluster centers in a supercluster.

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 III-C tell that although these two areas exhibit similar IQ domain representation (i.e., density), their representation on time domain exhibit different levels of burstiness/randomness. As shown in Equation (6), 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 (which is captured by P_{IQ}).

On one hand, for a location (I, Q) that 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., $s_i = S_k$). 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. Equation (4) tells that a signal's state exhibits strong burstiness in time domain if (I, Q) is close to a cluster center.

On the other hand, if (I, Q) is on an overlapping area, $P_{IQ}((I, Q)|S_k)$ exhibits no significant skewness for different S_k. In this case, $P_{BG}((I, Q)|i)$ will not change with the state of sample i (i.e., $P_T(i, k)$). That is to say, signal samples will arrive at the overlapping area with a constant average rate. In addition, due to the Gaussian noise, the signal's arrival at the overlapping area occurs randomly and independently. So, we can model the signal's arrival in the overlapping area as a Poisson process. The interval between successive arrivals follows the exponential distribution.

We use the supercluster shown in Figure 4(c) as an example to verify the above assumption. Specifically, we extract the signal samples from a central area and an overlapped area (as shown in Figure 8(a)). 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 $\mathbf{Int_o}$.

We can see that, although the average values of $\mathbf{Int_c}$ and $\mathbf{Int_o}$ are similar (indicating their similar densities), their distributions are quite different. The distribution of $\mathbf{Int_o}$ properly fits the exponential distribution, as shown in Figure 8(c). 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.

2) Center Identification: 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 $\frac{W}{2}$. Here, W can be adapted according to the noise level. For each location g, we denote the occurrence intervals between the samples in the window by $\mathbf{Int_g}$. We calculate two metrics of each location g: i) the number of samples in g, denoted by $D(g)$; and ii) the 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.

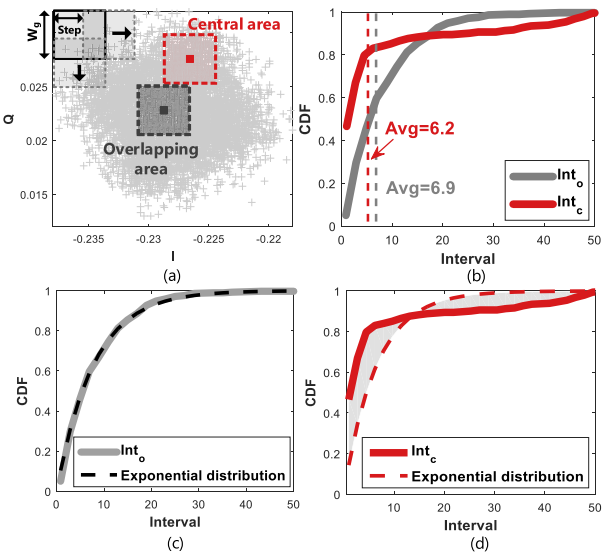


Fig. 8. Temporal distribution of the signal: (a) collided signal in IQ domain; (b) comparing the distribution of $\mathbf{Int_c}$ and $\mathbf{Int_o}$; (c) distribution of $\mathbf{Int_o}$; (d) distribution of $\mathbf{Int_c}$.

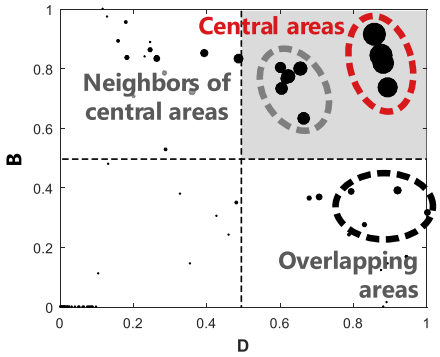


Fig. 9. Center identification based on $D(g)$ and $B(g)$.

Figure 9 shows the normalized density and burstiness of each grid g in Figure 8. We term this graph as a *determine graph*. We can see that only the central areas exhibit both high density and high burstiness. Then we propose a metric to quantify how likely a grid g is a central area:

$$P_c(g) = \beta \cdot B(g) \cdot D(g). \tag{7}$$

where β serves as a normalization constant.

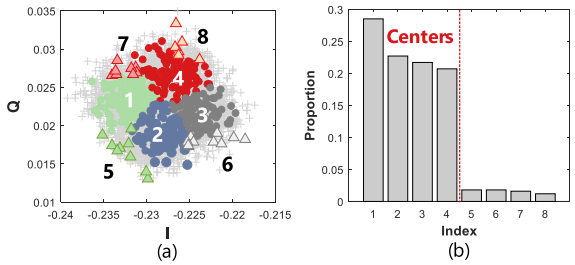


Fig. 10. Results of the trial process: (a) the result observed at the IQ domain; (b) proportion of the training samples that assigned to different candidates.

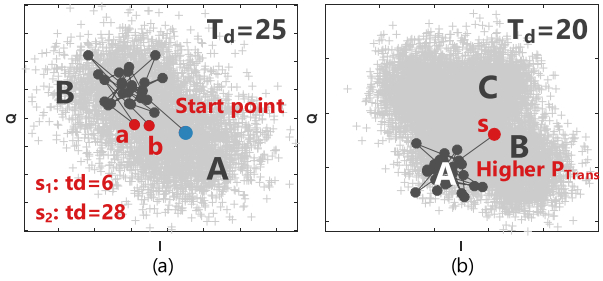


Fig. 11. Assigning a sample that located in the overlapped area: (a) two-cluster-overlapping case; (b) four-cluster-overlapped case.

A challenge we meet in identifying the centers based on P_c is that we do not know the number of clusters. A naive method 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 appropriate 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 *trial process*. Specifically, since central areas usually exhibit both high density and high burstiness, we consider the points that locate at the top right area of the decision graph (e.g., $D(g) > 0.5$ and $B(g) > 0.5$), 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 IV-C (where P_c serves as the weight of each candidate). The result can indeed help to recognize the cluster centers.

As an example, Figure 10 shows the trial 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. Here, the anomalously decrease in the 5-th rank helps us to identify the top 4 candidates as the cluster centers.

C. 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. 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 an example to illustrate our idea.

Figure 11(a) shows a example with two clusters A and B. In this case, it's hard to identify the states of the samples located in the overlapping area (e.g., a and b) based on their locations. By further considering the time domain information, we find that when sample a arrives, the signal has stayed at A for only 6 samples. While when b arrives, the signal as stayed for 28 samples. Since the average dwell time T_d is 25 samples in this case, we can infer that the signal is likely to flip when sample b arrives. So, sample a probably belongs to cluster A, while sample b probably belongs to cluster B.

In some cases, utilizing only the dwell time is not effective. As shown in Figure 11(b), 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 A based on the dwell time, but we cannot determine where the signal is transiting to. We solve this problem based on signal's different transition probability between clusters, as discussed in [18].

Based on the above discussion, a sample's state should be identified based on a joint consideration of the sample's location, signal's probable dwell time at a state, and the transition probabilities between states. We model this process as a Markov-based model. 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* and P_T, which captures the temporal burstiness of the signal acts as the *transition probability*. Then the likelihood for assigning sample i to state S_k is given by:

$$P(i, S_k) = P_w(S_k) \cdot P_{IQ}((I, Q)|S_k) \cdot P_T(i, k|t_d), \quad (8)$$

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

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.

$$\arg\max_{\mathbf{C}} \sum_{i=1}^{N} P(i, k). \quad (9)$$

This problem can be solved 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:

P_{IQ}: Equation (5) tells that P_{IQ} is related to two parameters: i) the location of cluster k; and ii) the covariance matrix $\mathbf{\Sigma}$. Assume that the background noise is relatively stable, Σ_k can be measured based on the clustering result of previous transmission tasks.

P_T: As shown in Equation (4), P_T is related to two parameters: P_d and P_{trans}. Specifically, P_d is initialized based on Equation (3), 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 number of transmission slots in one frame (denoted by K).

$$N = \sum_{R=1}^{N_{all}} R \binom{N_{all}}{R} \left(\frac{1}{K}\right)^R \left(1 - \frac{1}{K}\right)^{N_{all}-R} \quad (10)$$

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

Model update. P_d and P_{trans} are updated periodically at run-time. In each period, we just count the dwell time of the

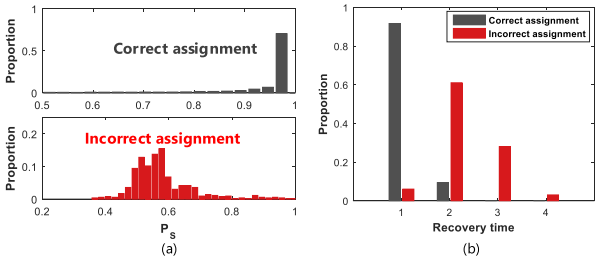


Fig. 12. Distribution of (a) $P(i,k)$ and (b) recovery time.

signal and the number of transitions between different clusters, based on the samples that are already identified. However, errors in the assignments, even rarely occurs, may lead to inaccurate estimation of P_d and P_{trans}. So, we propose a method to correct those errors, as explained in Section IV-D.

D. 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 12(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.

However, we cannot just identify an assignment with low $P(i,S_k)$ as an incorrect assignment[1] 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 12(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. Based on the above discussion, we can identify an assignment sequence, which exhibits successive low $P(i,S_k)$ as a suspicious segment.

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 13(a)). In some rare cases, the suspicious segment accidently covers a bit boundary (as shown in Figures 13(b) and (c)). 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}. Note that an incorrect merging in a cross-boundary case only leads to slight displacement of the bit boundary, which does not affect the decoding process.

V. Practical Issues

This section discusses some practical problems.

Partial collision problem. Partial collision problem refers to the case where the responses from two colliding tags suffer serious packet-level offset, as shown in Figure 14(a). This

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

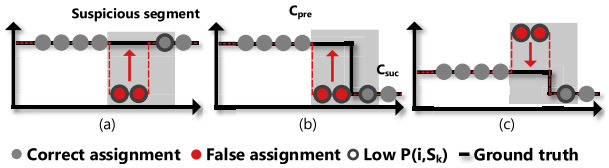


Fig. 13. Error correction method.

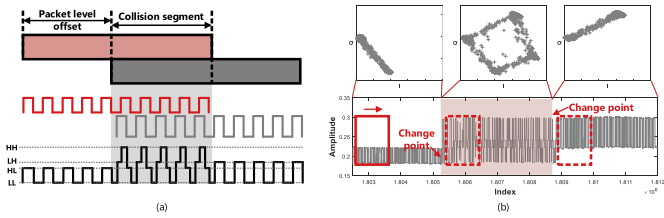


Fig. 14. Partial collision: (a) a two-tag example; (b) received signal.

makes the collided signal unevenly distributed on the IQ domain, which may lead to signal clustering errors. In the previous sections of this paper, we have assumed that the transmission of the colliding tags are synchronized by the reader (packets transmitted from different tags do not seriously offset with each other). However, if we relax the restriction, allowing tags to transmit totally asynchronously, the partial collision problem will occur.

To solve this problem, we use a moving window to detect the change point at which the number of colliding tags changes, as shown in Figure 14(b). Specifically, in each window, we count the number of clusters in the IQ domain. Once the cluster number changes, which indicates that the number of tags changes, a change point is detected. Then we segment the signal based on the change points and decode the signal in each segment individually. The number of colliding tags in each segment is unchanged.

Enabling partial decoding. Hubble cannot separate two clusters which are completely overlapped. This happens when a tag suffers particularly low signal SNR. In this case, Hubble tries to extract only the signals of the strong tags (if exist), and consider the weak signals as noise. Take the case in Figure 2(b) as an example, the signal from Tag 3 is too weak, making the clusters HHL and HHH, and LLL and LLH completely overlap with each other. In this case, Hubble 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 of Tags 1 and 2.

To decide which clusters should be merged, we exploit the bottom-up hierarchical clustering algorithm. Specifically, we measure the inter-dissimilarity between each pair of clusters (the distance between cluster centers), and merge the cluster pairs one by one in the descending order of the inter-dissimilarity. After each merging, we measure the self-dissimilarity of the combined clusters (the distance between the cluster centers of the two merged clusters). We get the optimal clustering result when both high inter-dissimilarity and low self-dissimilarity are achieved.

As an example, Figure 15 shows the hierarchical clustering process for the IQ signal in Figure 2(b). Figure 15(a) shows

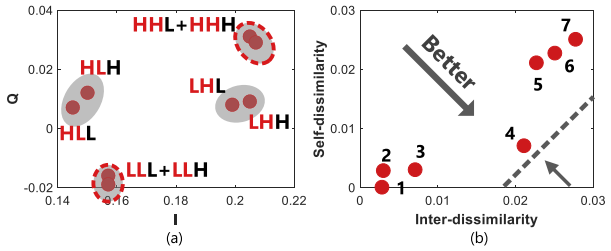


Fig. 15. Partial decoding: (a) IQ domain location; (b) cluster merging based on inter-dissimilarity and self-dissimilarity.

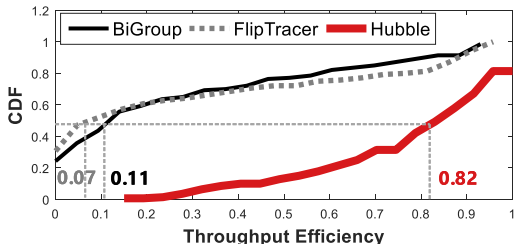


Fig. 16. Overall performance.

the locations of the cluster centers. Figure 15(b) shows the minimum value of inter-dissimilarity and the maximum value of self-dissimilarity that measured after each merging. The marked numerals means the number of clusters after each merging. Then we can get the optimal clustering result by finding the point that located at the lower-right corner.

VI. Discussion

Signal SNR v.s. Bitrate. The performance of a parallel decoding method is affected by two factors: channel quality (SNR) and the sampling rate of the reader, which respectively determine the signal resolution in IQ and time domains. Consider that a reader's sampling rate is two orders of magnitude higher than tags' bitrate, channel quality is the bottle-neck factor that restricts the decoding performance. Hubble is designed to break through this limitation. It achieves this by exploiting the bursty Gaussian behavior of the collided signal, which is brought by the asymmetry between the tags' bitrate and the reader's sampling rate. In another word, Hubble provides an opportunity to trade tags' bitrate for high parallel decoding reliability under low SNR. This motivates future research on optimizing the parallel decoding method, e.g., bitrate adjustment according to the channel quality.

Computation overhead. To reduce the computation overhead, instead of directly processing each sample using Hubble, we first divide the samples into grids (which is much less than the samples) and cluster those grid using DBSCAN. Therefore, the computation overhead for this process is only $O(k_g \cdot log k_g)$, where k_g is the number of the grids. Then Hubble separates the combined clusters using the method introduced in Sections IV-B and IV-C. The computation overhead of this process is $O(G + M \cdot (K + w))$, where M is the number of the samples that in the supercluster.

Generalization of Hubble. Hubble can be generalized to various platforms using OOK modulation. However, it has different performance when implemented on different platforms, depending on the MAC layer protocol the platform uses.

If we use the COTS tags, we need to strictly follow the standard EPC protocol where collisions are avoided by RN16 packets. Therefore, the performance gain of parallel decoding will be limited. Even so, achieving a 5∼6 decoding capacity can still significantly increase the channel utilization. The analysis result in [21] shows that a 5-tag capacity can increase the channel utilization from 37% to 88%.

If we remove the elements for channel contention in the EPC protocol, the tags can directly transmit their EPC ID. When collision occurs, the reader decodes the collided EPC ID using Hubble. In this case, the maximum performance gain of Hubble, compared with the EPC protocol, achieves 10.7×, as shown by our evaluation results in Section VII-H.

Impact of channel variation. Dynamic in both the working environment and the tags lead to channel variation, which changes the locations of the clusters on the IQ domain. Hubble is robust to such dynamics because the transmission time of a typical RFID packet is as short as a few milliseconds, which is within channel coherence time. This is a reasonable assumption verified in many existing works [15], [18].

VII. Evaluation

A. Experiment Setting

We implement Hubble on the USRP N210 and programmable tags (WISP). The SDR reader is connect with two UBX RF daughterboards. The sampling rate of the reader is set at 20MHz. The default bitrate of the tags is set at 100Kbps and the default packet length is set at 200 bits.

Our current implementation inherits some limitations of the WISP. Specifically, its coverage is limited by WISPs' communication range, which is typically within 2 m. This limitation is not inherent to our design. Hubble is generalizable to any backscatter platforms. For example, it can work with COTS tags whose communication range is 10m, or emerging RFID systems which enables long-range communication [7].

In the experiment, we compare the performance of Hubble with two state-of-the-art methods, **FlipTracer** [18] and **BiGroup** [15], which perform parallel decoding based on time and IQ domain information, respectively. Both of the two methods assumes separated clusters on IQ domain.

Methodology: We evaluate the performance of Hubble under different numbers of tags and different SNRs. The SNR is adjusted by changing the tag-reader distance. In the experiment, we change the number of tags from 2 to 6. In the case with n tags, 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, and move the other i tags (termed as *weak tags*) further away from the reader, until reaching the boundary for signal detection. In each location, the tags transmit 100 packets concurrently, and the reader decodes the collided signal using different approaches.

B. Overall Performance

Figure 16 shows the CDF of the *throughput efficiency* (defined as the ratio of throughput to aggregated PHY bitrate)

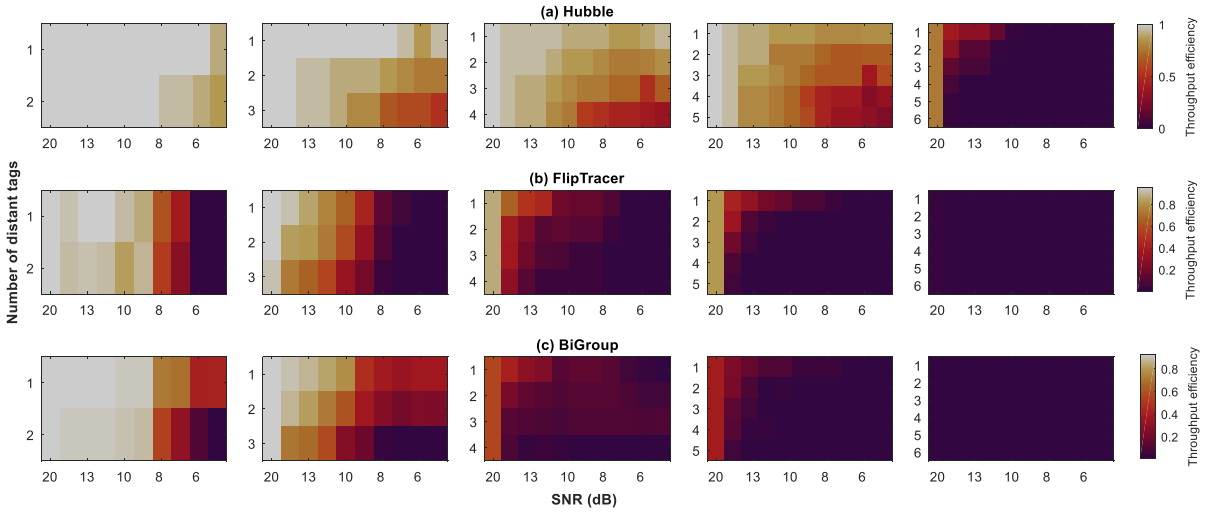


Fig. 17. Performance of different approaches under different SNR and different numbers of colliding tags.

of Hubble, FlipTracer, and BiGroup. The figure tells that the median throughput efficiency of FlipTracer and BiGroup are only 0.07 and 0.11, respectively. BiGroup performs slightly better than FlipTracer in low-SNR scenarios due to its ability to partially decode collisions. Hubble has a median throughput efficiency of 0.82, outperforming FlipTracer and BiGroup by $11.7\times$ and $7.4\times$, respectively.

More detailed results are given in Figure 17, which shows the throughput efficiency of the three schemes v.s. the SNR of the weak tags, under different numbers of concurrent tags and different proportions of weak tags. Clearly, all the three factors seriously affect the performance of FlipTracer and BiGroup. In comparison, Hubble is not so sensitive to those factors due to its ability to separate overlapped clusters.

C. Impact of the Signal SNR

As shown in Figure 17, the throughput efficiency of both FlipTracer and BiGroup decreases rapidly with SNR. This phenomenon becomes more apparent with more concurrent tags. In comparison, Hubble is much more robust to the low SNR. For example, when two tags transmit concurrently, Hubble achieves a 0.85 throughput efficiency even when the SNR of both the two tags is only about 4dB, while that of FlipTracer and BiGroup are 0.01 and 0.03, respectively. Even when 6 tags transmit concurrently and some of the tags exhibit low SNR, Hubble can still decode 15% packets.

D. Impact of the Number of Concurrent Tags

Figures 18(a) and 18(b) show the averaged throughput efficiency and bit error rate (BER) under different numbers of tags. As shown in the figure, with the increased number of concurrent tags, the BER of FlipTracer and BiGroup increase significantly due to the serious superclustering problem. The impact of the tag number is more apparent than that of the SNR because the number of clusters increases exponentially with the number of tags. Compared to FlipTracer and BiGroup, the performance of Hubble is not so sensitive to the increase

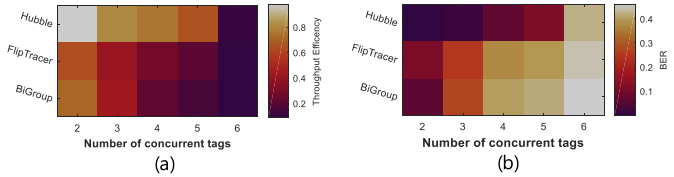


Fig. 18. Impact of tag number: (a) Throughput comparison; (b) BER comparison.

in the number of tags. Consider the case with 6 tags 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% throughput efficiency.

E. Transmission Across Obstacle

In this section, we evaluate the performance of Hubble in the NLOS (non-line-of-sight) scenario. The obstacle includes a $23 \times 16 \times 4$cm book, a glass panel, and a human. We have two test positions: the LOS position and the NLOS position, both are located 2ft away from the reader. For each kind of obstacle, we change the number of concurrent tags from 2 to 4. In the case with n tags, 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 tags are located at the LOS position. The result is shown in Figures 19.

As shown in the figure, the presence of an obstacle decreases the throughput of all the three approaches. For Hubble, when there are only two tags transmit concurrently, the performance degradation caused by the presence of the human is only $5\% \sim 9\%$. When 4 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. As shown in Figure 19, even when there are only two concurrent tags, the presence of an obstacle can lead to 51% and 65% performance degradation for BiGroup

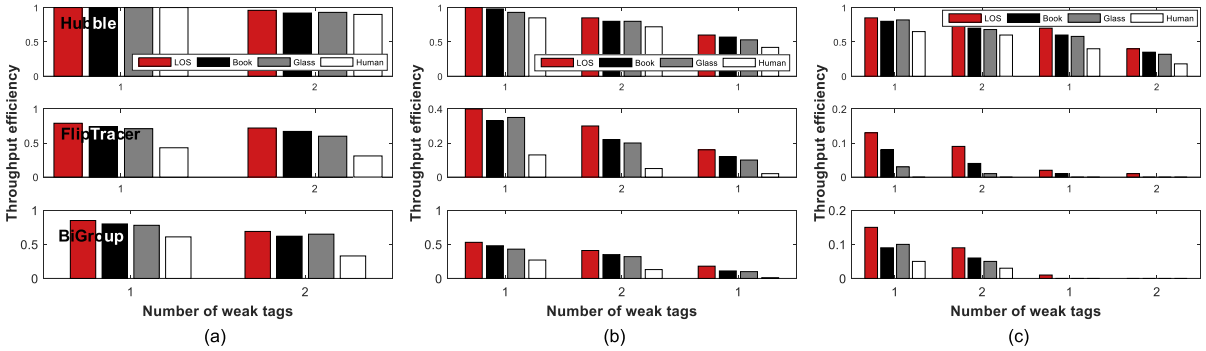


Fig. 19. Performance for NLOS transmission: (a) two tags; (b) three tags; (c) four tags.

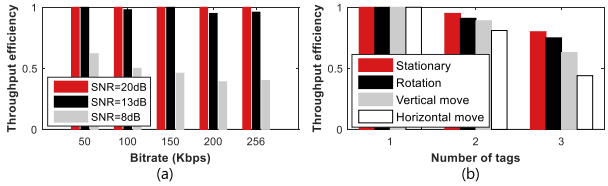


Fig. 20. Impacts of practical factors: (a) bitrate; (b) environment.

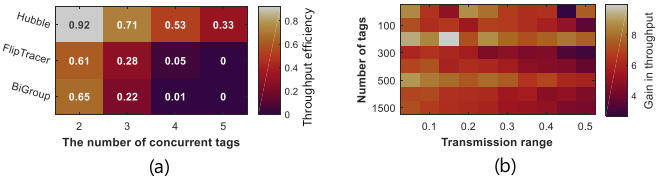


Fig. 21. Integrating Hubble with different MAC layer protocol: (a) random schedule; (b) EPC protocol.

and FlipTracer, respectively. So, the existing parallel decoding approaches do not work in highly cluttered environments.

F. Impacts of Practical Factors

Bitrate. Bitrate of the tags is also a important factor which affects the performance of Hubble. To evaluate the impact of bitrate, we conduct an experiment with three tags, and vary both the bitrate of the tags and the signal SNR. The result is shown in Figure 20(b). We can see that, Hubble can still achieve a decent performance at 256Kbps (the maximum bitrate that the WISP platforms can support).

Dynamic environment. We further investigate the impact of dynamic working conditions. Three cases are compared: i) the 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 result is shown in Figure 20(b).

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

G. In-the-Wild Experiment

To better demonstrate the efficacy of Hubble, we further evaluate its performance in the wild. In the experiments, 8 tags are located within the communication range (about 6 feet for the WISP tags) of the reader but at different distances to the reader. They transmit packets periodically with different interval ($20 \sim 30$ ms) and collisions occur randomly due to the different transmission schedules of the tags.

Figure 21(a) 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 packets. We can see that although slightly decreased, the in-the-wild performance of Hubble is still much better than FlipTracer and BiGroup.

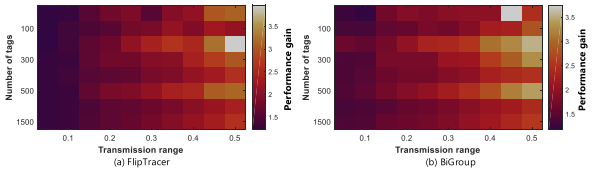


Fig. 22. Performance gain when $N_s = N$.

H. Trace-Driven Simulation

We perform trace-driven simulations to investigate the performance gain of Hubble when it is integrated with EPC protocol. 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 tags respond to the reader following the EPC standard. The number of slots N_s is adjusted by a parameter Q ($N_s = 2^Q$). In a collision slot, the reader decodes the received signal using Hubble, FlipTracer, and BiGroup, respectively.

Figure 21(b) compares the performance of Hubble with the traditional EPC protocol where collision slots will be discarded. For the traditional EPC protocol, we set $Q = \lfloor log_2^N \rfloor$. For Hubble, we set $Q = \lfloor log_2^{\frac{N}{3}} \rfloor$ to create more collisions. Figure 22 shows the performance gain of Hubble under different N and R. Here, R is normalized as the factor of tags' communication range. The result tells that i) the performance gain decreases with R. ii) the number of tags has no obvious influence on the performance gain;

Figure 22 shows the performance gain of Hubble compared with FlipTracer and BiGroup. In this experiment, Q is set

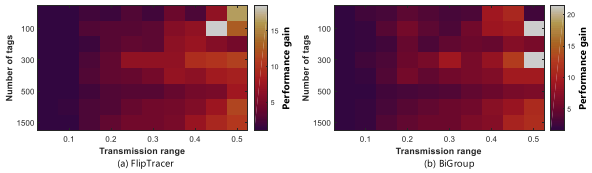


Fig. 23. Performance gain when $N_s = N/3$.

at $\lfloor log_2^N \rfloor$. The result tells that Hubble's performance gain compared with FlipTracer and BiGroup increases significantly with R. This is because that Hubble is more reliable than FlipTracer and BiGroup, especially in low SNR scenarios.

We further decrease Q to $\lfloor log_2^{N/3} \rfloor$ to evaluate the performance gain of Hubble with more collision slots. The result is shown in Figure 23. Clearly, the performance gain is much higher than that achieved when $Q = \lfloor log_2^N \rfloor$.

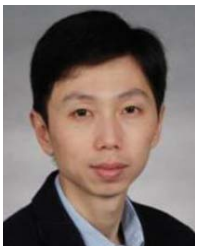
VIII. Conclusion

This paper presents Hubble, a signal processing approach to reliably trace signal state transitions, based on the bursty Gaussian behavior of the signal. Hubble achieves a 90% decoding rate when two tags transmit concurrently under a 5dB SNR, and a 70% decoding rate when five tags transmit under a 10dB SNR. With Hubble, the median throughput of parallel backscatter can be improved by 11.7×. However, due to the limited quality of the channel and the limited sampling rate of the reader, the decoding capacity of Hubble is limited to 6 tags. Potential solution may include integrating Hubble to multi-channel (multi-carriers) or multi-antenna scenarios, where the decoding capacity can be further multiplexed.

References

[1] J. Wang and D. Katabi, "Dude, where's my card?: RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM Conf.*, 2013, pp. 51–62.
[2] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. USENIX NSDI*, 2013, pp. 71–84.
[3] L. Shangguan, Z. Li, Z. Yang, M. Li, and Y. Liu, "OTrack: Order tracking for luggage in mobile RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 3066–3074.
[4] L. Yang, Q. Lin, X. Li, T. Liu, and Y. Liu, "See through walls with COTS RFID system!" in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 487–499.
[5] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 237–248.
[6] L. Shangguan, Z. Yang, A. Liu, Z. Zhou, and Y. Liu, "Relative localization of RFID tags using spatial-temporal phase profiling," in *Proc. USENIX NSDI*, 2015, pp. 251–263.
[7] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 335–343.
[8] T. Wei and X. Zhang, "Gyro in the air: Tracking 3D orientation of batteryless Internet-of-Things," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 55–68.
[9] X. Tong, F. Zhu, Y. Wan, X. Tian, and X. Wang, "Batch localization based on OFDMA backscatter," in *Proc. ACM Ubicomp*, 2019, pp. 1–25.
[10] EPCglobal Inc. (2008). *Epcglobal Class 1 Generation 2 v. 1.2.0*. [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2
[11] P. Zhang, J. Gummeson, and D. Ganesan, "BLINK: A high throughput link layer for backscatter communication," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 99–112.
[12] Y. Zheng and M. Li, "Read bulk data from computational RFIDs," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 1–5.
[13] W. Gong, H. Liu, K. Liu, Q. Ma, and Y. Liu, "Exploiting channel diversity for rate adaptation in backscatter communication networks," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.
[14] R. Zhao *et al.*, "OFDMA-enabled Wi-Fi backscatter," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2019, pp. 1–5.
[15] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for cots RFID tags," in *Proc. ACM MobiCom*, 2015, pp. 500–511.
[16] P. Hu, P. Zhang, and D. Ganesan, "Leveraging interleaved signal edges for concurrent backscatter," in *Proc. 1st ACM Workshop Hot Topics Wireless*, 2014, pp. 13–18.
[17] P. Hu, P. Zhang, and D. Ganesan, "Laissez-faire: Fully asymmetric backscatter communication," in *Proc. ACM Conf. Special Interest Group Data Commun.*, 2015, pp. 255–267.
[18] M. Jin, Y. He, X. Meng, Y. Zheng, D. Fang, and X. Chen, "FlipTracer: Practical parallel decoding for backscatter communication," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 1–5.
[19] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM Conf. Appl., Technol.*, 2012, pp. 1–6.
[20] D. Shen, G. Woo, D. P. Reed, A. B. Lippman, and J. Wang, "Efficient and reliable low-power backscatter networks," in *Proc. ACM RFID*, 2009, pp. 61–72.
[21] C. Angerer, R. Langwieser, and M. Rupp, "RFID reader receivers for physical layer collision recovery," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3526–3537, Dec. 2010.
[22] A. Bletsas, J. Kimionis, A. G. Dimitriou, and G. N. Karystinos, "Single-antenna coherent detection of collided FM0 RFID signals," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 756–766, Mar. 2012.
[23] R. S. Khasgiwale, R. U. Adyanthaya, and D. W. Engels, "Extracting information from tag collisions," in *Proc. IEEE Int. Conf.*, Apr. 2009, pp. 131–138.
[24] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. IEEE KDD*, 1996, pp. 226–231.

Meng Jin (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Northwest University, Xi'an, China, in 2012, 2015, and 2018, respectively. She is currently a Post-Doctoral Researcher with the School of Software, Tsinghua University. Her main research interests include backscatter communication, wireless network co-existence at 2.4 GHz, mobile sensing and clock synchronization. She is a member of the ACM.



Yuan He (Senior Member, IEEE) received the B.E. degree from the University of Science and Technology of China, the M.E. degree from the Institute of Software, Chinese Academy of Sciences, and the Ph.D. degree from the Hong Kong University of Science and Technology. He is currently an Associate Professor with the School of Software and TNLIST of Tsinghua University. His research interests include the Internet of Things, sensor networks, pervasive computing, and cloud computing. He is a member of the ACM.

Xin Meng received the B.S. degree in computer science from Northwest University, Xi'an, China, in 2012, where he is currently pursuing the master's degree. His main research interests include localization and wireless networks.

Dingyi Fang (Member, IEEE) received the Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2001. He is currently a Professor with Northwest University, Shaanxi International Joint Research Center for the Battery-free Internet of Things, Xi'an, China, and the Internet of Things Research Center, Northwestern University, Xi'an, China. His current research interests include mobile computing and distributed computing systems, network and information security, and wireless sensor networks. He is a member of the ACM.

Xiaojiang Chen (Member, IEEE) received the Ph.D. degree from Northwest University, Xi'an, China, in 2010. He is currently a Professor with Northwest University, Shaanxi International Joint Research Center for the Battery-free Internet of Things, Xi'an, China and Northwest University-Jingdong Wisdom Cloud Joint Research Center for AI & IoT, Xi'an, China. His current research interests include localization and performance issues in wireless ad hoc, mesh, and sensor networks and named data networks. He is a member of the ACM.