

Packet Separation in Phase Noise Impaired Random Access Channel

Pajovic, M.; Ozcan, G.; Koike-Akino, T.; Wang, P.; Orlik, P.V.

TR2018-167 December 12, 2018

Abstract

A growing number of applications may benefit from embedding receiver with the capability to separate collided packets directly on a physical layer. For example, in IoT's random access channel scenario, a number of IoT devices are placed in the same cell, occasionally transmit messages, and are assigned non-orthogonal channel resources, giving rise to packet collisions. While channel access techniques for sharing a common channel are usually employed in most multi-user communication systems, they cause channel underutilization and increased latency. This paper considers a scenario where multiple users asynchronously transmit packets over a shared channel and are not subject to a random access mechanism, or the mechanism itself fails to prevent packet collisions. Assuming that each packet consists of a common preamble and information bearing payload, and experiences random delay, frequency offset, phase noise variation and block flat fading channel, we propose a packet separation algorithm which recovers collided packets, estimates their parameters and detects corresponding payload symbols. The performance of the proposed method is validated using simulations and benchmarked against bounds.

IEEE Global Communications Conference (GLOBECOM)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Packet Separation in Phase Noise Impaired Random Access Channel

Milutin Pajovic¹, Gozde Ozcan², Toshiaki Koike-Akino¹, Pu Wang¹, Philip Orlik¹

¹Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139, USA

² Broadcom Limited, Irvine, CA, 92617, USA

Email: pajovic@merl.com

Abstract—A growing number of applications may benefit from embedding receiver with the capability to separate collided packets directly on a physical layer. For example, in IoT's random access channel scenario, a number of IoT devices are placed in the same cell, occasionally transmit messages, and are assigned non-orthogonal channel resources, giving rise to packet collisions. While channel access techniques for sharing a common channel are usually employed in most multi-user communication systems, they cause channel underutilization and increased latency. This paper considers a scenario where multiple users asynchronously transmit packets over a shared channel and are not subject to a random access mechanism, or the mechanism itself fails to prevent packet collisions. Assuming that each packet consists of a common preamble and information bearing payload, and experiences random delay, frequency offset, phase noise variation and block flat fading channel, we propose a packet separation algorithm which recovers collided packets, estimates their parameters and detects corresponding payload symbols. The performance of the proposed method is validated using simulations and benchmarked against bounds.

I. INTRODUCTION

A communication system involving multiple users which transmit signals over non-orthogonal resources (space, frequency, time, and/or code) conventionally has a builtin channel access mechanism whose goal is to reduce likelihood of collisions between packets sent by different users, e.g., [1]. While channel access techniques reduce the number of packet collisions, they lead to channel underutilization and increased latency. Consequently, a variety of applications may benefit from embedding the receiver with the capability to separate collided packets directly on a physical layer.

One such application that has recently received a considerable attention pertains to the Internet of Things (IoT) realm, where a large number of IoT devices are placed within the same cell and occasionally transmit short messages to the common base station [2], [3]. Employing conventional communication strategies in the design of such a system turns out to be highly challenging. Namely, due to a large number of IoT devices and short messages they transmit, allocating orthogonal resources to them is nearly impossible. Also, since the IoT devices only occasionally transmit messages, using network resource management and control may lead to an excessive communication overhead. Hence, resolving packet collisions directly on a physical layer seems to be the most promising strategy.

In some other applications, the capability of separating collided packets directly on a physical layer may be a viable feature in addition to a conventional channel access mechanism. In particular, the channel access mechanism may grant transmission to a latency-critical packet which may collide with a packet already on the air, thus prompting the need to separate the colliding packets in the receiver. For example, different entities on a factory floor wirelessly exchange command and control messages. Reliable delivery of those messages with low latency is the main requirement to be met to facilitate efficiency in manufacturing processes [4]. In yet another application, the V2X paradigm is expected to enhance safety on the roads and, consequently, a timely delivery of safety related messages is critical.

Motivated by the above applications, we consider a random channel access problem where multiple users asynchronously transmit packets in the same frequency band. Due to Doppler shift and imperfections in the transmitters' oscillators, caused for example by employing inexpensive oscillators in the IoT devices, each transmitted packet exhibits frequency offset and phase noise impairment. To aid packet separation, each packet consists of a common preamble, known in advance, while the information bits are contained in the packet payload. We propose a packet separation algorithm which estimates collided packets' delays, frequency offsets, corresponding channel coefficients and phase noise variations, as well as detects payload bits.

II. RELATION TO PRIOR WORK

A growing research body of packet separation for random access channel assumes that only a small portion out of a large and known number of users simultaneously transmit symbols and exploits sparse recovery framework to detect active users and their transmitted information [5]. As such, assuming perfect time synchronization, [6], [7] employ element-wise sparsity, [8] performs block sparsity aided separation, [9] probabilistically treats the separation problem by imposing an arrival model on users, while [10], [11] propose reduced-dimension multi-user detection (MUD) processors exploiting analog sparsity. Sparse recovery approaches are also employed in smart-meter applications [12], [13].

Few more recent works are relevant to current paper. As such, [14] considers time and frequency asynchronous transmissions of known and user-specific packets that go

through unknown multipath channels and separates them on the receiver side for the purpose of link acquisitions of active users. The separation algorithm from [15] considers packets encoded with spreading codes, asynchronous in time, synchronous in frequency, and transmitted through known, single path channels. Finally, [16], [17] propose separation algorithms when transmitted packets are encoded with spreading codes, transmitted through unknown multipath channels, asynchronous in time with relative delay of up to one signaling interval, and synchronous in time. In comparison, each packet in our setup has the same preamble and unknown data content, not subject to a spreading/precoding, and channels from active users are unknown and estimated on the receiver side. In addition, the transmissions in our setup are asynchronous in time and frequency, and each packet undergoes a random and independent phase noise variation, estimated and corrected on the receiver side.

Since the transmitted packets undergo random frequency offsets, the considered scenario resembles a sinusoid separation problem setup. A vast literature has been devoted to a sinusoid separation problem, with prominent results being the MUSIC algorithm [18], recent off-the-grid optimization-based approaches exploiting frequency domain sparsity [19], [20] and a practical OMP-like recovery with frequency estimate refinement step [21]. Although the proposed algorithm separates colliding packets in the frequency offset and delay domain, we emphasize that the considered problem is more challenging than sinusoid separation problem in a sense that each atom in a given frequency offset bin carries unknown payload symbols and experiences unknown delay and phase noise variation.

The closest work to this paper is [22]. In comparison, we reduce the computational complexity of the algorithm in [22] by shrinking the search space for a packet. More importantly, we address the issue of phase noise impairments, arising due to using inexpensive oscillators in the future IoT devices and sensors. Specifically, phase noise variation from each active user is estimated on the receiver side and corrected to yield accurate payload symbol detection.

III. ARRIVAL MODEL

In the considered scenario, K different users transmit signals to a receiver employing a single antenna. The symbols of the k -th user are formatted into packets $\mathbf{s}_k \in \mathbb{C}^{N \times 1}$, where each packet is concatenation of preamble $\mathbf{p} \in \mathbb{C}^{M_1 \times 1}$ and payload $\mathbf{p}_k \in \mathbb{C}^{M_2 \times 1}$, i.e., $\mathbf{s}_k = [\mathbf{p}^T \quad \mathbf{p}_k^T]^T$ and $M_1 + M_2 = N$. The preamble is known in advance and same for all packets. The payloads carry information and are detected on the receiver side.

Each user, k , modulates its packet \mathbf{s}_k and transmits a waveform $x_k(t)$ obtained as

$$x_k(t) = \mathcal{M}(\mathbf{s}_k), 0 \leq t \leq T, \quad (1)$$

where \mathcal{M} denotes a modulation operator. The duration of the transmitted waveform is $T = NT_s$, where T_s is the symbol duration. Without loss in generality, we assume the transmitted

power of user k is contained in the corresponding channel coefficient.

The users send signals asynchronously so that the transmitted waveform, $x_k(t)$, experiences delay τ_k . This delay also incorporates the propagation delay between user k and the receiver. In addition, each waveform is impaired by frequency offset f_k and phase noise $\theta_k(t)$. The signal at the receiver is a superposition of the received signals originating from all active users. The overall received signal is pre-processed, downconverted, and its complex baseband representation is given by

$$r(t) = \sum_{k=1}^K h_k x_k(t - \tau_k) e^{j2\pi f_k(t - \tau_k)} e^{j\theta_k(t)} + v(t). \quad (2)$$

Above, $v(t)$ is a circularly symmetric additive white Gaussian noise (AWGN), i.e., $v(t) \sim \mathcal{CN}(0, \sigma^2)$, while $h_k \in \mathbb{C}$ is a coefficient (gain) of the channel between user k and the receiver. This model corresponds to a block flat fading channel and handles narrowband systems with relatively short packets (shorter than the channel coherence time). Additionally, the model can also handle channels with a single dominant path.

We note that the frequency offsets f_k and phase noise realizations $\theta_k(t)$ are independent across different users because they are caused by different oscillators. Each $\theta_k(t)$ is modeled as a Wiener process, meaning that the phase noise increments during two non-overlapping time intervals are independent and Gaussian distributed, i.e.,

$$\theta_k(t + \Delta t) - \theta_k(t) \sim \mathcal{N}(0, \sigma_p^2(\Delta t)), \quad (3)$$

where the variance σ_p^2 depends on the time difference Δt .

The complex received signal $r(t)$ over some observation time window of duration T_{obs} is sampled with oversampling factor N_o , yielding $N_o T_{\text{obs}}/T_s$ samples. The vector of the generated samples, \mathbf{r} , is the input to the proposed packet separation algorithm. The proposed method estimates the packet delays τ_k , frequency offsets f_k , phase noise realizations $\theta_k(t)$ and channel gains h_k , and detects the transmitted payload symbols \mathbf{p}_k , $k = 1, \dots, K$.

The described packet separation method is, in general, obsolete to modulation format in (1). However, we focus on the Gaussian Minimum Shift Keying (GMSK) modulation [23], as it has some desirable features well suited for potential applications of our interest. Namely, the GMSK waveform has constant modulus so that relatively cheap radio frequency (RF) components can be used, which is especially desirable for IoT applications. In addition, the GMSK has relatively good spectral efficiency and reduced sidelobe levels. While the use of the GMSK modulation is reflected in the way we perform joint carrier phase estimation (CPE) and payload detection in Section IV-C, we stress out that our packet separation method can handle any other modulation format by employing an appropriate joint CPE and symbol detection algorithm.

IV. PROPOSED PACKET SEPARATION ALGORITHM

This section presents the proposed packet separation algorithm. We start with a discussion on sparse recovery formu-

lation of the problem and then proceed with a more detailed elaboration of the proposed algorithm.

A. Sparse Recovery Formulation

Packets transmitted by different users likely differ in delay, or frequency offset, or both. Therefore, we specify a delay-frequency offset domain as

$$\mathcal{S} = \mathcal{T} \times \mathcal{F} = \{(\tau, f) : \tau \in [0, T_{\max}], f \in [f_{\min}, f_{\max}]\}, \quad (4)$$

where $T_{\max}, f_{\min}, f_{\max}$ are, respectively, maximum delay, minimum and maximum frequency offset, all dependent upon a specific application. The range of delays \mathcal{T} and frequency offsets \mathcal{F} are discretized to yield, respectively, \mathcal{T}_d and \mathcal{F}_d . The bin size chosen in each discretization depends on application, resolution properties of the employed preamble, and available computational resources. Overall, the discretized search space for the packet separation method is $\mathcal{S}_d = \mathcal{T}_d \times \mathcal{F}_d$.

Packet collisions occur as a result of transmissions from only a small number of users in the IoT random access channel setting, and are thus described within a sparse recovery framework. Neglecting phase noise impairments in our problem setup for a moment, the received signal can be represented as a sparse and linear combination of atoms, where an atom is a waveform characterized with particular delay, frequency offset and payload symbols, while the coefficients in the linear combination are channel gains corresponding to active user. More formally,

$$\mathbf{r} = \mathbf{A}\mathbf{h} + \mathbf{v}, \quad (5)$$

where columns (i.e., atoms) in \mathbf{A} are obtained by modulating each possible combination of payload bits, appended to the fixed and known preamble, and applying each possible delay and frequency offset $(\tau, f) \in \mathcal{S}_d$. Sparse vector \mathbf{h} contains non-zero channel coefficients in the entries corresponding to active atoms, while \mathbf{v} is the discretized noise vector.

The sparse recovery formulation has been utilized in other works on random access channel [6]–[9], [12], [13]. However, non-zero entries in an unknown sparse vector in those works are transmitted symbols of active users, while the columns of the sensing matrix depend on users' spreading codes and full knowledge of channels of all users available at the receiver. In addition, in case of asynchronous transmissions [24], atoms also encode possible delays. In comparison to those works, we consider a more realistic scenario where users' channels are not available at the receiver. This, in fact, gives rise to quite a different model (5) from the one considered in the referenced papers.

In principle, one may apply a sparse recovery algorithm to (5) to detect active atoms in \mathbf{A} and, in turn, estimate delays, frequency offset and payloads, along with the corresponding channel coefficients. However, a closer look reveals that such an approach would suffer from a prohibitive computational complexity because the number of possible payload combinations for each (τ, f) pair in \mathcal{S}_d is exponentially large. For example, a simple BPSK modulation yields 2^{M_2} payload combinations for each (τ, f) . Accounting for phase noise further exacerbates this issue.

To overcome the computational issue, we build upon [22] and extend it to handle transmissions with phase noise impairments. The crux of our algorithm lies in approximating model (5) with

$$\mathbf{r} \approx \mathbf{A}'\mathbf{h}' + \mathbf{v}, \quad (6)$$

where a column in \mathbf{A}' , corresponding to a particular (τ, f) , represents the most likely waveform that would be received if indeed there were a transmission with delay τ and frequency offset f . That is, instead of considering all possible payload combinations for a given $(\tau, f) \in \mathcal{S}_d$, and placing the corresponding waveforms into separate columns of \mathbf{A} in (5), we estimate a single most likely waveform for each (τ, f) . We note that in this formulation, \mathbf{h}' is a shrunk version of \mathbf{h} , and contains the same non-zero entries as \mathbf{h} .

The proposed algorithm iteratively recovers collided packets using (6), where the approximated sensing matrix \mathbf{A}' is learned in each iteration. The details are elaborated in the following part.

B. Packet Separation Algorithm

To simplify the exposition, we describe the algorithm in the continuous time domain. However, in practice, upon the discretization of the received signal $r(t)$ into \mathbf{r} , all processing is done in the discrete time domain. The pseudo-code of the algorithm is given in Algorithm 1.

1) *Reduced-Complexity Learning of Sensing Matrix:* To avoid determining the most likely waveform for each $(\tau, f) \in \mathcal{S}_d$, a matched filter bank (MFB) is implemented to indicate relevant (τ, f) bins for a packet search. That is, for each (τ, f) bin from the set of bins containing the location of the MFB peak $(\hat{\tau}_0, \hat{f}_0)$, and a certain number of surrounding bins, we detect what the transmitted symbols would be if a packet were indeed located at that (τ, f) bin, and estimate the corresponding waveform. This essentially results in estimates of the relevant atoms in \mathbf{A}' .

Specifically, the received signal $r(t)$ is processed through the MFB, where the impulse response of each filter is the preamble waveform, tuned to a specific frequency offset from \mathcal{F}_d . Denoting with $p(t)$ the modulated waveform of the preamble, the template of a filter in the MFB, turned to a particular frequency $f \in \mathcal{F}_d$, is given by

$$p_f(t) = p(t)e^{j2\pi ft}, 0 \leq t \leq T_1. \quad (7)$$

The location of the MFB output magnitude is given by

$$(\hat{\tau}_0, \hat{f}_0) = \arg \max_{(t,f) \in \mathcal{S}_d} |r(t) \star p_f(-t)|, \quad (8)$$

where \star denotes the convolution. The set of (τ, f) bins around $(\hat{\tau}_0, \hat{f}_0)$, relevant for a packet search is identified as

$$\mathcal{N}(\hat{\tau}_0, \hat{f}_0) = \{(\tau, f) : |\tau - \hat{\tau}_0| \leq \bar{T}, |f - \hat{f}_0| \leq \bar{F}\}, \quad (9)$$

where \bar{T} and \bar{F} specify the size of the search area. These values are chosen based on application, desired accuracy, employed preamble, and expected arrival scenario.

In the following, we determine most likely payload symbols for each $(\tau, f) \in \mathcal{N}(\hat{\tau}_0, \hat{f}_0)$. Specifically, if a packet has delay

τ , the portion of the received signal $r(t)$ containing that packet is

$$r_\tau(t) = r(t + \tau)\mathbb{1}_{[0, T]}(t), 0 \leq t \leq T, \quad (10)$$

where $\mathbb{1}_{[t_1, t_2]}(t)$ is 1, if $t_1 \leq t \leq t_2$, and zero otherwise. Similarly, if this packet has frequency offset f , $r_\tau(t)$ is frequency compensated to yield

$$\tilde{r}_{\tau, f}(t) = r_\tau(t)e^{-j2\pi ft}, 0 \leq t \leq T. \quad (11)$$

Assuming the power of the interfering packets is relatively small, the compensated signal $\tilde{r}_{\tau, f}(t)$ effectively contains the considered (hypothesized) packet, impaired by carrier phase noise, block flat fading channel and AWGN noise. This signal is input to a block which jointly performs CPE, estimation of channel coefficient and GMSK demodulation to yield, respectively, the estimates of carrier phase noise $\hat{\theta}_{\tau, f}(t)$, channel coefficient $\hat{h}_{\tau, f}$, and payload symbols $\hat{\mathbf{p}}_{\tau, f}$. This joint estimation is outlined in IV-C.

Once the most likely payload symbols corresponding to a hypothesized transmission at (τ, f) are determined, we estimate the corresponding waveform. Specifically, assuming payload symbols $\hat{\mathbf{p}}_{\tau, f}$ were transmitted with delay τ , frequency offset f and phase noise $\hat{\theta}_{\tau, f}(t)$, the received waveform would be, up to a channel coefficient, given by

$$z_{\tau, f}(t) = \mathcal{M}([\mathbf{p}^T \hat{\mathbf{p}}_{\tau, f}^T]^T) e^{j2\pi ft} e^{j\hat{\theta}_{\tau, f}(t)}, 0 \leq t \leq T. \quad (12)$$

The discretized version of this waveform is a column in \mathbf{A}' , corresponding to the considered $(\tau, f) \in \mathcal{N}(\hat{\tau}_0, \hat{f}_0)$.

2) *Packet Recovery*: Upon learning the relevant atoms in \mathbf{A}' , standard steps from greedy sparse recovery algorithms, such as matching pursuit (MP) or orthogonal MP (OMP), are applied for packet recovery and residual signal update. Specifically, a packet is recovered by maximizing an objective function given as a cross-correlation between the received segment $r_\tau(t)$ and $z_{\tau, f}(t)$, i.e.,

$$J(\tau, f) = \int_0^T r_\tau(t) z_{\tau, f}^*(t) dt, \quad (13)$$

where the integral above is effectively an inner product between the corresponding vectors in the discrete time domain. The delay and frequency offset of the recovered packet are obtained as

$$(\hat{\tau}, \hat{f}) = \arg \max_{(\tau, f)} J(\tau, f), \quad (14)$$

where the search is done over the set $\mathcal{N}(\hat{\tau}_0, \hat{f}_0)$. The payload symbols, phase noise and channel coefficient corresponding to the estimated pair $(\hat{\tau}, \hat{f})$ are already determined in the joint CPE and GMSK demodulation step. They are indexed with $(\hat{\tau}, \hat{f})$ and are, respectively, $\hat{\mathbf{p}}_{\hat{\tau}, \hat{f}}$, $\hat{\theta}_{\hat{\tau}, \hat{f}}(t)$ and $\hat{h}_{\hat{\tau}, \hat{f}}$.

3) *Residual Signal Update*: The procedure described in the previous parts recovers a single packet. The contribution of that packet to the received signal $r(t)$ is given by

$$r_{\hat{\tau}, \hat{f}}(t) = \hat{h}_{\hat{\tau}, \hat{f}} \mathcal{M}([\mathbf{p}^T \hat{\mathbf{p}}_{\hat{\tau}, \hat{f}}^T]^T) e^{j2\pi \hat{f}t} e^{j\hat{\theta}_{\hat{\tau}, \hat{f}}(t)}, 0 \leq t \leq T. \quad (15)$$

The contribution of the recovered packet to the received signal is removed from the received signal to obtain a residual signal

$$\tilde{r}(t) = r(t) - r_{\hat{\tau}, \hat{f}}(t - \hat{\tau})\mathbb{1}_{[\hat{\tau}, T + \hat{\tau}]}(t), 0 \leq t \leq T. \quad (16)$$

The residual signal is then processed using the steps described in Sections IV-B1 and IV-B2, where it is treated as a received signal. Notably, the residual signal update from (16) resembles that of the MP approach. Alternatively, the OMP-like residual signal update can be used and is expected to perform better at the expense of increased computational complexity.

The packet separation method continues as an iterative process, where each iteration recovers one packet. The iterations continue until some stopping criterion is satisfied, for example, when the power of the residual signal falls below some threshold dependent on the AWGN noise variance. Alternatively, if Cyclic Redundancy Check (CRC) is part of the transmitted packet, it is used to check if the detected symbols are correct. In that case, the recovery proceeds to next iteration as long as the most recently detected packet passes the CRC.

C. Joint Carrier Phase Estimation and Payload Detection

As already pointed out, the estimation of the carrier phase noise and channel coefficient, as well as payload symbol detection are all performed jointly. The input signal to this block is $\tilde{r}_{\tau, f}(t)$, evaluated in (11), and we assume it contains a single, frequency compensated packet. The joint CPE and payload detection is performed in three processing steps, outlined in this part.

First, known preamble waveform $p(t), 0 \leq t \leq T_1$, is utilized to estimate the phase noise corresponding to the preamble segment of the received signal. Without delving into the details, the phase information is contained in [25]

$$u(t) = [\tilde{r}_{\tau, f}(t)\mathbb{1}_{[0, T_1]}(t)] p^*(t). \quad (17)$$

The signal $u(t)$ is processed through a moving average (MA) filter of certain length. The phase estimate $\hat{\theta}_{\tau, f}(t), 0 \leq t \leq T_1$, is obtained after taking the angle argument of the MA filter output and performing the angle unwrapping operation.

Second, the estimated phase noise corresponding to the preamble part is used to compensate $\tilde{r}_{\tau, f}(t)$ to yield

$$\tilde{r}'_{\tau, f}(t) = [\tilde{r}_{\tau, f}(t)\mathbb{1}_{[0, T_1]}(t)] e^{-j\hat{\theta}_{\tau, f}(t)}, 0 \leq t \leq T_1. \quad (18)$$

The channel coefficient is then estimated from $\tilde{r}'_{\tau, f}(t)$ and preamble waveform $p(t)$ using the least squares (LS) fit.

Finally, the phase noise of the payload part is estimated blindly using the approach proposed in [25]. Essentially, the method applies a phase noise estimate corresponding to a certain symbol period to compensate received signal at the following symbol period, detects the symbol upon phase noise compensation, and re-estimates the corresponding phase noise. In addition, phase noise estimates of a certain number of consecutive symbols are filtered using moving average (MA) filter in order to reduce phase estimation error. In comparison to the original CPE method from [25], which is designed for memory-less modulation formats, we extend it to

Algorithm 1: Packet Separation Algorithm

Require: Received signal $r(t)$
Initialize search space \mathcal{S} as in (4)
Initialize MFB as in (7)
Initialize residual signal $\tilde{r}(t) = r(t)$
for $i = 1, 2, \dots, K$ **do**
 Process $\tilde{r}(t)$ through the MFB.
 Find coarse estimates $\hat{\tau}_0$ and \hat{f}_0 as in (8)
 Define search area $\mathcal{N}(\hat{\tau}_0, \hat{f}_0)$ as in (9)
 for $(\tau, f) \in \mathcal{N}(\hat{\tau}_0, \hat{f}_0)$ **do**
 Extract received portion $r_\tau(t)$ as in (10)
 Compensate freq. offset to obtain $\tilde{r}_{\tau,f}(t)$ as in (11)
 Estimate $\hat{\theta}_{\tau,f}(t)$ and $\hat{h}_{\tau,f}$, and detect $\hat{\mathbf{p}}_{\tau,f}$ from $\tilde{r}_{\tau,f}(t)$ as outlined in Section IV-C
 Evaluate $z_{\tau,f}(t)$ as in (12)
 Evaluate objective function $J(\tau, f)$ as in (13)
 end for
 Find $\hat{\tau}^{(i)}, \hat{f}^{(i)} = \arg \max_{\tau, f} J(\tau, f)$
 Detected payload symbols $\hat{\mathbf{p}}^{(i)} = \hat{\mathbf{p}}_{\hat{\tau}^{(i)}, \hat{f}^{(i)}}$
 Find residual $\tilde{r}(t)$ as in (16)
end for
return Parameters and symbols of the collided packets

handle formats with memory, such as the GMSK modulation. Towards that end, a symbol-by-symbol GMSK detection is required and we employ the optimum detection described in [23]. However, any other sub-optimum and computationally less demanding GMSK detection can also be used.

V. SIMULATION RESULTS

The proposed algorithm is validated with Monte-Carlo simulations where we consider an arrival scenario consisting of two packets. Since separating two packets that have fairly different delays and frequency offsets is relatively easy, to validate the algorithm, the two packets in our scenario completely overlap in time so that their delays are $\tau_1 = \tau_2 = 7$. The normalized frequency offset of one packet, referred to as Packet 1, is $f'_1 = f_1 T_s = 6.5 \times 10^{-4}$, while two different cases for the normalized frequency offset of the other packet, referred to as Packet 2, are considered, $f'_2 = 0.05$ and $f'_2 = 0.025$. The received power of Packet 1 is 0 dB and the AWGN variance is -15 dB. The power of Packet 2 is subject to a sweep from 0 dB down to -10 dB, so that the power ratio of the two packets, P_1/P_2 , changes from 0 dB to 10 dB.

Each packet in our study consists of the same and known 32-bit preamble and an unknown 192-bit payload. The payload bits for each packet are generated uniformly at random, appended to the preamble and modulated using the IoT-friendly GMSK modulation with memory, with bandwidth–symbol duration product $BT_s = 0.5$ and oversampling rate of 4. The phase noise is independent across packets and along each packet follows a Wiener process with $\sigma_p = 1^\circ$ per symbol. The channel coefficients corresponding to each packet

have independent, uniformly distributed phase, while their magnitude depends on the desired received packet power.

The received signal is the input to the described packet separation algorithm. The algorithm is run two iterations so that it recovers two packets and outputs detected payload bits and estimated parameters (delays, frequency offsets, channel coefficients and phase noise waveforms). The normalized frequency offset domain contains 251 bins spanning from -0.1 to 0.1 , while the resolution of the delay domain is equal to the sampling period ($1/4$ of the symbol duration). The grid search area in (9) is specified with $\bar{T} = 1$ and $\bar{F} = 15$.

As a performance metric, we measure bit and packet error rate (BER and PER) of each packet. As benchmarks, we simulate the performance of the successive interference cancellation (SIC) and maximum likelihood (ML) joint detection (i.e., Viterbi algorithm) with perfect knowledge of the packets' delays, frequency offsets, channel coefficients and phase noise realizations, respectively referred to as the SIC and ML bound. The averaging in all three schemes is done over 1,000 Monte-Carlo runs (so that the minimum measurable BER and PER are, respectively, 5.2×10^{-6} and 10^{-3}). Each simulation run has different realization of the AWGN, phase noise in each packet, and phases of channel coefficients.

The BER's and PER's of the two packets are shown in Figures 1, 2, 3 and 4, for $f'_2 = 0.05$, and in Figures 5, 6, 7 and 8, for $f'_2 = 0.025$. As can be seen, the BER and PER performance of Packet 1 detection exhibit insignificant degradation compared to the SIC bound in both considered cases, and no bit error has been observed for P_1/P_2 above 5dB in all cases. Also, the BER and PER corresponding to Packet 2 are fairly close to the SIC bound for $f'_2 = 0.05$, indicating that the proposed algorithm fairly accurately estimates packets' parameters. As the two packets get closer in the frequency offset domain when $f'_2 = 0.025$, the BER and PER performance of Packet 2 detection deteriorate, however is still relatively close to the SIC bound over a number of P_1/P_2 points. The largest deterioration happens at P_1/P_2 of around 5 – 6dB. A closer examination reveals that in this regime, the errors of phase noise estimation of Packet 1, even though not large to impact payload detection of Packet 1, occasionally cause inaccuracies in the residual signal (that should contain only Packet 2) that, in turn, yields completely wrong localization of Packet 2 in the delay-frequency offset domain. Approaches to overcome this issue will be sought for in our future research.

VI. CONCLUSION

An IoT scenario where multiple users asynchronously transmit packets over a shared channel causing packet collisions on the receiver side is considered in this paper. Each packet consists of known and common preamble, and information bearing payload. The signal from each user is impaired with frequency offset and carrier phase noise, and is transmitted through a block flat fading channel. The described packet separation algorithm estimates the parameters of the colliding packets (i.e., relative delays, frequency offsets, phase noise realizations and channel coefficients) and detects their payload

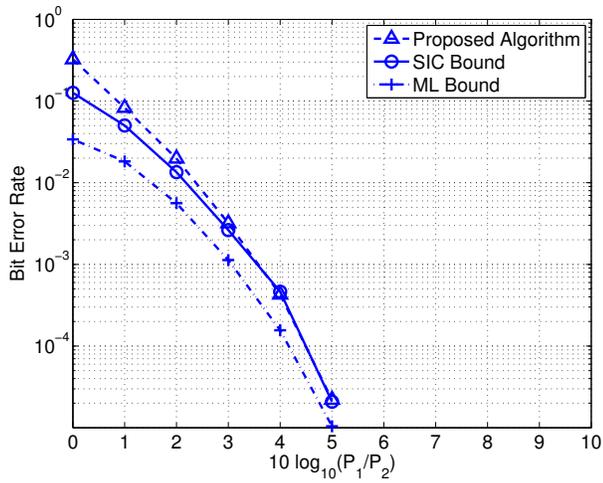


Fig. 1: BER of Packet 1 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.05$.

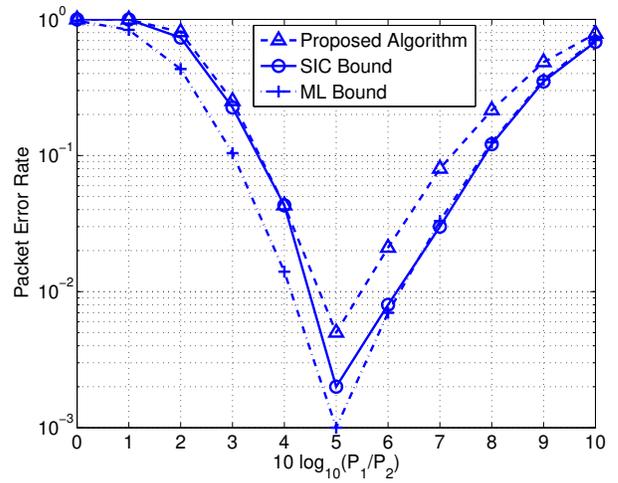


Fig. 4: PER of Packet 2 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.05$.

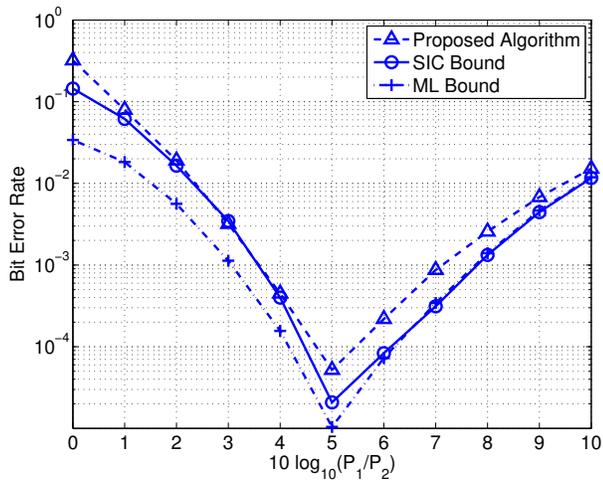


Fig. 2: BER of Packet 2 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.05$.

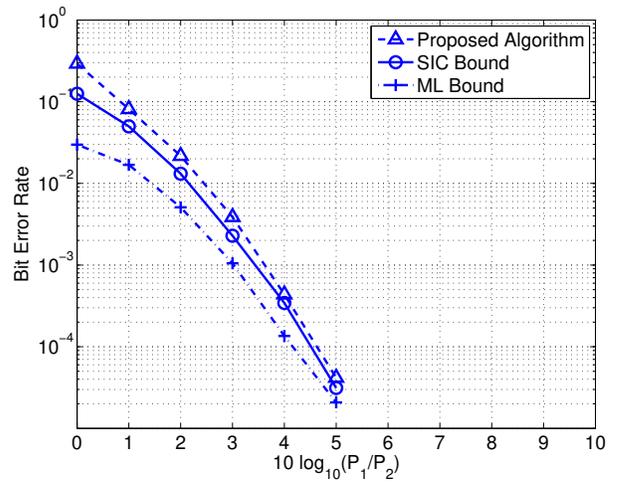


Fig. 5: BER of Packet 1 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.025$.

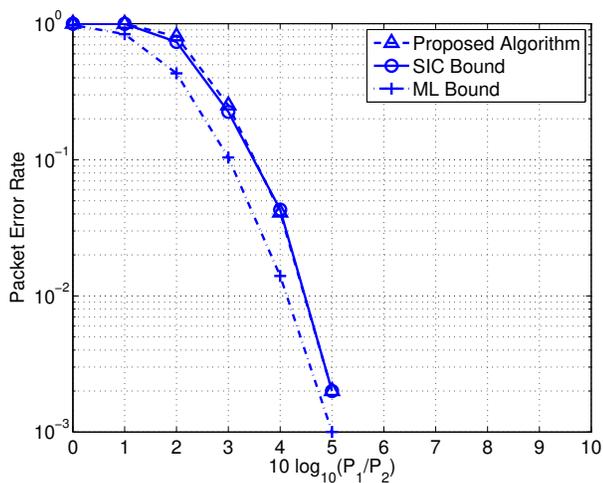


Fig. 3: PER of Packet 1 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.05$.

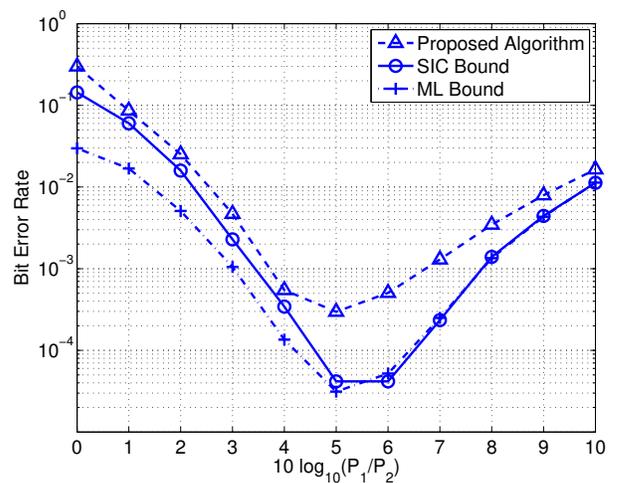


Fig. 6: BER of Packet 2 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.025$.

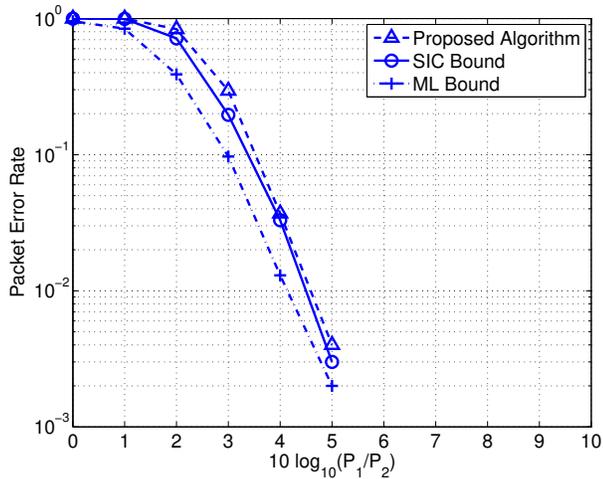


Fig. 7: PER of Packet 1 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.025$.

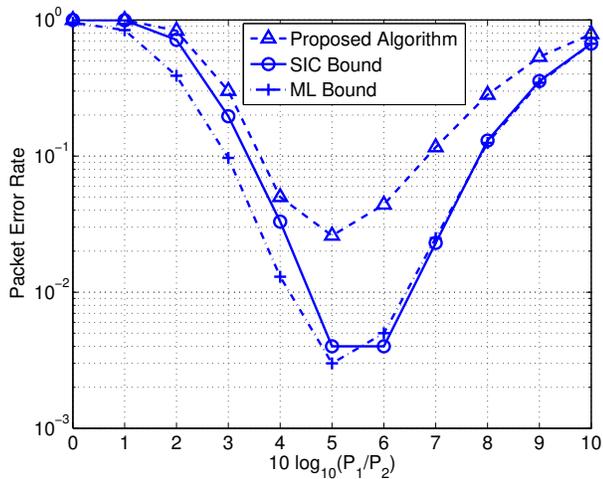


Fig. 8: PER of Packet 2 for $f'_1 = 6.5 \times 10^{-4}$ and $f'_2 = 0.025$.

symbols. The described algorithm is validated using simulations of a two-packet arrival scenario where the two packets completely overlap in time and are close in the frequency offset domain.

REFERENCES

- [1] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 2, pp. 544–556, 2012.
- [2] L. Dai, B. Wang, Y. Yuan, S. Han, C. I. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, Sept 2015.
- [3] G. Wunder and et. al., "5GNOW: Non-orthogonal, asynchronous waveforms for future mobile applications," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 97 – 105, Feb 2014.
- [4] A. Frotzsch, U. Wetzker, M. Bauer, M. Rentschler, M. Beyer, S. Elspass, and H. Klessig, "Requirements and current solutions of wireless communication in industrial automation," *2014 IEEE International Conference on Communications Workshops (ICC)*, pp. 67–72, June 2014.
- [5] A. K. Fletcher, S. Rangan, and V. K. Goyal, "On-off random access channels: A compressed sensing framework," *arXiv preprint arXiv:0903.1022*, 2009.
- [6] H. F. Schepker and A. Dekorsy, "Sparse multi-user detection for CDMA transmission using greedy algorithms," in *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*. IEEE, 2011, pp. 291–295.
- [7] C. Bockelmann, H. F. Schepker, and A. Dekorsy, "Compressive sensing based multi-user detection for machine-to-machine communication," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 4, pp. 389–400, 2013.
- [8] R. Xie, H. Yin, X. Chen, and Z. Wang, "Many access for small packets based on precoding and sparsity-aware recovery," *IEEE Transactions on Communications*, vol. 64, no. 11, pp. 4680 – 4694, Nov 2016.
- [9] H. Zhu and G. B. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 454–456, Feb 2011.
- [10] Y. Xie, Y. C. Eldar, and A. Goldsmith, "Reduced-dimension multiuser detection," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3858–3874, 2013.
- [11] M. Pajovic and P. Orlik, "Reduced-dimension symbol detection in random access channel," *IEEE International Conference on Communications (ICC)*, May 2018.
- [12] R. H. Y. Louie, W. Hardjawana, Y. Li, and B. Vucetic, "Distributed multiple-access for smart grid home area networks: Compressed sensing with multiple antennas," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2938–2946, Nov 2014.
- [13] H. Li, R. Mao, L. Lai, and R. C. Qiu, "Compressed meter reading for delay-sensitive and secure load report in smart grid," *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct 2010.
- [14] X. Li, A. Rueetschi, A. Scaglione, and Y. C. Eldar, "Compressive link acquisition in multiuser communications," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, June 2013.
- [15] Y. Chi, Y. Xie, and R. Calderbank, "Compressive demodulation of mutually interfering signals," *IEEE Statistical Signal Processing Workshop*, 2012.
- [16] S. Wang, J. An, Y. Ren, T. Wang, and X. Bu, "Compressed receiver for multipath dsss signals," *IEEE Communication Letters*, vol. 18, no. 8, Aug 2014.
- [17] S. Wang, S. AN, X. Miao, Y. Ma, and S. Luo, "Compressed sensing assisted joint channel estimation and detection for ds-cdma uplink," *IEEE Communication Letters*, vol. 19, no. 10, Oct 2015.
- [18] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar 1986.
- [19] E. J. Candès and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," *Communications on Pure and Applied Mathematics*, vol. 67, no. 6, pp. 906–956, 2014.
- [20] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5987–5999, Dec 2013.
- [21] B. Mamandipoor, D. Ramasamy, and U. Madhow, "Newtonized orthogonal matching pursuit: Frequency estimation over the continuum," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5066–5081, Oct 2016.
- [22] G. Shulkind, M. Pajovic, and P. Orlik, "Packet separation in random access channels via approximate sparse approximation," *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov 2017.
- [23] J. G. Proakis, *Digital Communicatians*, 4th, Ed. McGraw-Hill, 2000.
- [24] H. F. Schepker, C. Bockelmann, and A. Dekorsy, "Coping with CDMA asynchronicity in compressive sensing multi-user detection," in *IEEE 77th VTC Spring*. IEEE, 2013, pp. 1–5.
- [25] E. Ip and J. M. Kahn, "Feedforward carrier recovery for coherent optical communications," *Journal of Lightwave Technology*, vol. 25, no. 9, pp. 2675–2692, Sept 2007.