Unsourced Random Access With Threshold-Based Feedback

Murwan Bashir, Member, IEEE, Ehsan Nassaji, Graduate Student Member, IEEE, Dmitri Truhachev, Member, IEEE, Alireza Bayesteh, Member, IEEE, and Monirosharieh Vameghestahbanati, Member, IEEE

Abstract—In this paper we study feedback mechanisms for unsourced random access (URA) communications. We propose an algorithm to construct feedback packets broadcasted to the users by the base station (BS) and a feedback packet format that allows the users to estimate their channels and infer positive or negative feedback based on the presented thresholding algorithms. We show that the proposed feedback technique leads to a substantial reduction in the packet error rates and signal-to-noise ratios (SNRs) required to support various numbers of active users. We also demonstrate that the proposed feedback imposes a much smaller complexity burden on the users compared to the feedback that acknowledges only successful or only all undecoded users. Finally, we present a theoretical analysis framework that closely matches the experimental results.

Index Terms—Unsourced random access (URA), feedback, multiple user detection, compressed sensing.

I. INTRODUCTION

UNSOURCED random access (URA) is a grant-free multi-user communication setup where a multitude of active users transmit their messages to a sink point in a sporadic manner, without a mechanism to identify the transmitting users [1]. The time-varying set of active users is a subset of a much larger pool of potential users that may eventually communicate. In URA applications, the unique identifiers (IDs) of the active users are unimportant and the receiver is only interested in the message content itself. Such setting presents itself naturally in various sensor network applications, where the knowledge of the mapping between the messages and user IDs does not add extra value. Absence of user IDs is especially important for short packet transmission, since inclusion of a user ID into the message will significantly reduce the data portion of the packet. Moreover, the complexity of a multiple user detection (MUD) receiver that pre-allocates resources based on pre-defined IDs of all possible users can be extremely high for a large pool of potential users.

The ID-free access strategy also reflects itself in the measurement of the URA performance. While the traditional multi-access communications targets high reliability across all users, the goal of the URA is to ensure that the majority of the transmitted users’ messages are recovered. Hence, the per user probability of error (PUPE) has been proposed in [1] to measure URA performance reflecting the focus on the number of successfully delivered messages, rather than on delivering all messages error-free.

Multiple URA systems proposed in the literature attempt to approach the achievability benchmark derived in [1], in terms of the minimum SNR required to support a given number of active users at a specific level of PUPE. Because of the sparse nature of the user activity in URA, where only a small number of users are active at any time slot, design of access methods that enable compressed sensing (CS) detection was among the first attempts of URA implementations. For example, in [2], [3], and [4] the transmitted packet is partitioned into sub-blocks that are encoded by a code constructing a binary tree. The receiver first utilizes CS detection algorithms to recover the sets of active user message partitions for each sub-block. Following that, the tree decoders reconstruct the complete user messages from these sets.

Another large class of URA systems, capable to support high numbers of active users, is based on the preamble-payload format [5], [6], [7]. In the preamble-payload format the packet is sub-divided into a small preamble and a larger payload. The preamble carries a part of the message data and, at the same time, enables channel access by acting as a temporary user ID. CS detection algorithms are typically used to recover the preamble part. The payload is usually constructed via preamble-dependent encoding and modulation through permutation and scrambling. At the receiver, the payloads are retrieved using MUD algorithms with some form of inter-interference cancellation and error-correction decoding.

In general, a URA system allows for packet transmission without enforcing reliability as an aspect of the system design. However, informing active users about the status of their received messages is beneficial in terms of stopping frequent re-transmissions. Moreover, implementation of feedback to enhance reliability enables the design of re-transmission protocols that further reduce the required system SNR compared to the feedback-free URA as we demonstrate in this paper.

0090-6778 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Prince Edward Island. Downloaded on June 19, 2023 at 10:02:53 UTC from IEEE Xplore. Restrictions apply.
We also show that the feedback can help saving power at the active user nodes which are often low-power battery-operated devices.

The classic grant-free communications follows two main approaches to ensure low error probability of the packet reception. The first approach targets reduction of the packet collision probability by simple packet repetition [8] or linear network coding [9]. However, in this approach, the receiver is only benefiting from the repetition diversity or the imposed error-correction, and any knowledge of the packet delivery success is not communicated back to the active users. The second approach introduces hybrid automatic repeat request (HARQ) feedback protocol, where the receiver broadcasts the decoding status to the transmitters. It has been shown that the latter outperforms the repetition-like approaches in terms of reliability [10] at the cost of some additional signaling. For example, in a technique proposed in [11] and [12] the BS sends a negative acknowledgment (NACK) to colliding users in an attempt to stop simultaneous re-transmissions. Similarly, the status of packet decoding can be fed back to assist in scheduling and collision mitigation as shown in [13] and [14]. In [15], for the case of low number of active users, positive feedback is broadcasted to the successful active users. The users employ a message-passing algorithm to detect their signatures in the feedback message and decide on the type of the received feedback (ACK or NACK). If active users can obtain channel state information, power-based threshold access can be used to enhance packet reliability at the receiver [16].

In asynchronous grant-free transmission settings, feedback is utilized to inform candidate active users waiting for re-transmission about vacant channel slots [17]. Another utilization of feedback, focused on resource allocation, is presented in [18], where the feedback is used in a grant-based setting to avoid collisions and allow for user scheduling. In [18], the authors advocate that their proposed alternative system can be used to perform the same function as the URA system presented in [1]. In [12], a grant-free access system with feedback is proposed to resolve collisions using orthogonal pilots during re-transmission. A similar feedback strategy is employed in [19] to prevent user collisions in case active users have unequal data packet sizes.

In [20], an upper bound on the number of feedback bits required to ensure collision-free transmission is provided, where the feedback is solving, optimally, a collision-free scheduling problem in a massive random access. In [21], bounds on the minimum number of feedback bits for a positive-only feedback for a grant-free access in a packet-erasure channel with false positives is derived. The bounds, and their practical realisations show a significant reduction in terms of feedback length compared to the naive feedback solution which combines ID-based feedback messages into one broadcasted feedback packet. The authors suggest that the analysis may be extended to URA systems on wireless channels. In [22], users utilize secret keys and feedback to generate authentication codes for message verification at the receiver.

In order to support the reliability requirements of the (massive machine-type communication) mMTC using (massive multiple access) MMA, [23] presents a repetition-based spatial coupling technique introduced at the medium access control (MAC) layer. The physical layer processing imperfections deteriorate the performance of the decoder working on the spatially coupled repeated packets. Therefore, (pilot-based) feedback is used to support the reliability of the decoder, limit the number of transmissions (repetition rate), and lower the energy consumption. The feedback is assumed to be perfect.

Finally, in URA setting, the feedback has been used very recently to enhance system reliability. In [24], the Base Station utilizes the correctly decoded messages to create a hash, and with use of beamforming, the acknowledgments are broadcasted with directionality to target the intended users. This acts as a positive feedback acknowledgement and it is shown in [24] that the length of the feedback messages needs to scale linearly with the number of the targeted users.

In this paper, we study a generic problem of setting up feedback in a URA system and identify the advantages the feedback can bring to URA. We start with two baseline feedback mechanisms, the positive-only feedback in which the BS sends a positive acknowledgement to all successful users, and the negative-only feedback where the BS sends a negative acknowledgement to all detected but undecoded users. While URA systems do not utilize fixed user IDs, focusing on preamble-payload URA allows us to work with feedback messages formed as a superposition of a subset of all active user preambles broadcasted back to the user pool. We then propose a feedback format in which the users can estimate their channels and use thresholding to infer a success or failure of their packet on the feed-forward link or receive a personalized feedback message. We further develop this idea to construct a system where most of the users can save power by processing only a fraction of the feedback message. In order to describe the system mathematically and quantify the gains of the proposed feedback mechanism numerically, we utilize the preamble-payload format presented in [6] and [7]. We note, however, that the proposed techniques are applicable to any preamble-payload URA format.

Our contributions can be summarized as follows:

- We propose a general threshold-based feedback framework for URA systems. We demonstrate that the proposed approach improves reliability while conserving the overall system resources compared to the positive-only or negative-only baseline feedback techniques.
- We demonstrate that the proposed feedback results in a significant reduction in the SNRs required to support high numbers of active users in block Rayleigh fading channel. We show that the system with feedback and a single re-transmission can provide a significant advantage over a feed-forward only URA.
- We present a double-threshold feedback system to enable further complexity reduction on the active user side by allowing partial processing of the feedback message.
- We present an analytic framework to quantify the PUPE, throughput, and SNR of a URA system with feedback.

The paper is organized as follows. Section II presents the system model and the structure of the feed-forward link.
at the transmitter and receiver. In Section III we present two baseline feedback systems of POF and NOF before presenting the proposed single and double threshold-based feedback approaches. We then propose the utilization of partial signatures and a feedback detection mechanism at the active users side. Section IV describes the performance analysis. Section V presents the numerical results, and Section VI concludes the paper.

II. SYSTEM MODEL

A. Feed-Forward Link

In the preamble—payload URA approach [5], [6], [7], a packet of an active user consists of the preamble and the payload. The preamble encodes a fraction of user’s data bits, and identifies the unique features utilized in payload encoding such as scrambling and permutation sequences. These features facilitate the separation of users’ payloads at the receiver side via MUD. In the next sections we detail the operation of each system unit of the feed-forward link.

Consider a pool of $K_{\text{tot}}$ users. Only a much smaller subset that consists of $K_a \ll K_{\text{tot}}$ is active at any given time slot. The $k^{th}$ active user encodes its $n$-bit data message using a forward error-correction (FEC) encoder and obtains the codeword $v^k = [v^k_1, v^k_2]_0$ of length $B$, $k = 1, 2, \cdots, K_a$. The vector $v^k$ is further encoded to form the preamble, while the vector $v^k_2$ is modulated to form the payload. The vector $v^k$ that contains $B_3$ bits is mapped into the preamble sequence $x^k_p \in \mathbb{C}^{N_p \times 1}$ using a CS encoder. The sequence $x^k_p$ corresponds to a column in the sensing matrix $A = [a_1, a_2, \cdots, a_N]$, where the column index $\nu(k) = (v^k_1)_0$ and $(\cdot)_0$ indicates binary to decimal conversion, $x^k_p = a_{\nu(k)}$. The $N_p \times N$ matrix $A$ where $N = 2^{B_p}$, composed of i.i.d. complex Gaussian entries, is chosen a priori and contains the pool of unitary preambles. These preamble sequences are selected by the active users according to the above-mentioned mapping. We will denote the set of active user indices by $K_a = \{\nu(k)\}$, $k = 1, 2, \cdots, K_a$. In case several users select the same preamble sequence a collision occurs. Hence, the set $K_a$ may contain duplicate indices.

The payload is formed as follows. The $B_3$-bit sequence $v^k_d$ (where $B_3 + B_p = B$) is encoded using repetition, permutation, and scrambling. Based on the index $\nu(k)$ the $k^{th}$ user selects a pair of payload scrambling and permutation sequences $(\pi_{\nu(k)}, s_{\nu(k)})$ from a pool of $2^{B_p}$ scrambling and permutation sequence pairs and uses it for payload encoding. Each bit of the data vector $v^k_d$ is repeated $M$ times and permuted using $\pi_{\nu(k)}$. The resulting permuted bit replicas are used to form quadrature phase shift keying (QPSK) symbols which are then scrambled (multiplied symbol by symbol) with $s_{\nu(k)}$, which is a sequence with random unitary complex numbers to obtain the payload $x^k_p$ that consists of $N_d$ symbols, where $N_d = N_3 + N_p$.

B. Preamble Decoder

The receiver operates in two stages. First the CS decoder decodes the preambles and estimates, simultaneously, the set of the active user indices $K_a$ and channel coefficients $h_k$, $k \in K_a$ for each user. Note that once the activity detection (AD) estimates the preambles and resolves $v^k_p$, it also identifies the scrambling—permutation pairs $(\pi_{\nu(k)}, s_{\nu(k)})$ used for payload encoding. Following that, the results of the preamble decoding are passed to the MUD that decodes the payloads of the packets via an iterative parallel interference cancellation and data decoding algorithm. The composite received signal for the preambles of the users’ packets is given by

$$y_p = \sum_{k=1}^{K_a} h_k x^k_p + z_p = \mathbf{A} \mathbf{h} + z_p,$$

where $\mathbf{h}$ is a sparse $(2^{B_p} \times 1)$ vector of the user activity and channel coefficients that has non-zero entries at $k \in K_a$. The received signal $y_p$ contains preamble sequences $a_k$ of the active users, weighted by the channel coefficients $h_k$. We consider user transmission over independent block Rayleigh channels with coefficients $h_k$. The additive white Gaussian noise (AWGN) at the receiver is denoted by $z_p \sim C(0, N_0 I_{N_p})$. The preamble CS decoder utilizes an iterative algorithm proposed in [6] that is based on the approximate message-passing (AMP) [25]. Iteration $l$, $l = 0, 1, 2, \cdots$, starts with calculation of the residual noise

$$z^l_p = y_p - \widetilde{\mathbf{h}}^{l-1} + O(z^l_p),$$

where $\mathbf{h}$ is the vector of all channel coefficient estimates at iteration $l$, $\mathbf{h}_l = \mathbf{0}$. The residual noise and interference vector $z^l_p$ is matched-filtered with the sensing matrix $A$ and accumulated with the previous estimate, to find the initial estimate $r^l$ of $\mathbf{h}$, that is

$$r^l = \mathbf{h}_l + A^* z^l_p.$$

The estimate $r^l$ is further improved to produce

$$\hat{\mathbf{h}}_l = \eta(r^l, \tau^2_l),$$

using a soft denoiser $\eta(\cdot)$ (see definition in [4] for the real case and also [6] for the complex case), where $\tau^2_l$ is the variance of the components of the vector $z^l_p$ computed via $\|z^l_p\|^2 / N_p$. Finally, thresholding with $c\tau_l$ is applied

$$\hat{h}_{j+1} = \begin{cases} \hat{h}_j, & |\hat{h}_j| > c\tau_l, \\ 0, & \text{otherwise}, \end{cases} j = 1, \cdots, 2^{B_p}$$

to reduce the number of false alarms and assist in convergence. The threshold multiplier $c$ is usually selected in the range $2 \leq c \leq 4$. This step is similar to the thresholding techniques that appear in many versions of the AMP [25], [26]. After the final iteration, the receiver obtains the estimated indices of the active user preambles $\nu(k) \in \hat{K}_a$ and their corresponding channel estimates $\hat{h}_k = \hat{h}_{\nu(k)}$, $k = 1, 2, \cdots, \hat{K}_a$, where $\hat{K}_a$ is the number of the detected active users. The user activity detection errors consist of miss-detected users (where $\kappa \in K_a$ but $\kappa \notin \hat{K}_a$) and false alarms ($\kappa \notin K_a$ but $\kappa \in \hat{K}_a$).
C. Payload MUD

The MUD structure which includes both the interference cancellation and the error-correction decoding, shown in Fig. 1 (see [6]), outputs a list \( L \) of the decoded codewords \( \hat{b}^k \) for all detected users’ signals. The composite received signal for the payloads is given by

\[
y_d = \sum_{k=1}^{K_a} h_k x_d^k + w_d, \tag{6}
\]

where \( w_d \) is i.i.d. Gaussian measurement noise. At the beginning of the iterative MUD process, the channel estimate \( h_k \) of each detected user \( k \), obtained by the AD algorithm, is utilized to produce the matched filtered received signal

\[
r_k = h_k^* y_d = |h_k|^2 x_d^k + \sum_{k' \neq k} h_k^* h_k' x_d^{k'} + h_k^* w_d + e_k^* h_k x_d^k, \tag{7}
\]

where \( e_k = h_k - h_k \). The MUD algorithm aims to iteratively reduce the inter-payload interference, and ameliorate user’s payload data bits (see Fig. 1)

\[
\vartheta_{k,j}^t = \sum_{m=1}^{M} \tilde{\lambda}_{k,j,m}^t, \tag{9}
\]

where \( \tilde{\lambda}_{k,j,m}^t \) is the LLR for the \( m \)-th replica of the coded bit \( \nu_{k,j,m} \), \( j \in \{1, \ldots, B_d\} \) of user \( k \). Based on the LLRs we obtain soft bit estimates

\[
\tilde{v}_{k,j,m} = \tanh \left( \sum_{m' = 1}^{M} \tilde{\lambda}_{k,j,m'}^t \right). \tag{10}
\]

The bit estimates in (10) are used to re-modulate the user’s signal with application of permutation, signature and a channel estimate to produce

\[
y_{d}^{k,t} = h_k^* x_d^{k,t}, \tag{11}
\]

which are utilized in the interference cancellation.

When the signal-to-noise and interference ratio (SINR) at iteration \( t \) is above a certain pre-set threshold \( \alpha \) the error-correction decoder is activated at the receiver instead of the repetition-based soft-bit estimation (10). The LLRs \( \vartheta_{k,j}^t \) are passed to the input of the error-correction decoder together with the detected preamble data bits \( \hat{v}_{p}^k \). We consider polar codes with cyclic redundancy check (CRC) bits for data encoding, while successive cancellation list decoding is used at the receiver. If the decoded user’s data codeword checks the CRC = 0 for two consecutive iterations, it is assumed that the user’s data is decoded successfully. The resulting re-modulated payload will further participate in hard interference cancellation and its channel estimate update. The threshold \( \alpha \) utilized here ranges in \((-20, -11)\) dB depending on the number of active users \( K_a \).

The bits output by the error-correction decoder (or soft bits (10)) are utilized to form hard QPSK symbols, \( q_{k,j}^t \) that are
repeated $M$ times, permuted, and scrambled, to re-modulate the data of the respective packet prior to the interference cancellation. In order to aid the remaining users, the channels of the successfully decoded users are re-estimated using the decoded data as pilots. The part of the received signal that corresponds to the users, successfully detected and decoded at iteration $t$ (set $S_n[t]$), is given by

$$y_c = \sum_{k \in S_n[t]} h_k x^k + w_t = X_I h_t + w_t,$$

(12)

where $X_I$ is the matrix of the stacked successfully decoded data vectors acting as a pilot signal, and $w_t$ is the signal of noise and interference from other channels missed users (for which the interference cancellation is applied), with variance $\sigma^2_{\text{NIP}}(t)$ (the noise and interference power (NIP)). The linear minimum mean square estimator (LMMSE) of the respective channel coefficients $h_t$ vector is given by

$$\hat{h}_t = X^*_I \left( X_I X^*_I + \sigma^2_{\text{NIP}}(t) \right)^{-1} y_d$$

$$= \frac{X_I}{\sigma^2_{\text{NIP}}(t)} \left( I - X^*_I \left( I + \frac{X^*_I X_I}{\sigma^2_{\text{NIP}}(t)} \right)^{-1} \right) y_c.$$

The usefulness of the new channel estimates $\hat{h}_t$ is evaluated by the reduction of NIP before and after the estimation. The NIP after the channel estimation is given by

$$\sigma^2_{\text{NIP}}(t^+) = \| y_c - X_I \hat{h}_t \|^2,$$

(13)

where $t^+$ we denote an intermediate channel estimation step within the $t$-th iteration. If $\sigma^2_{\text{NIP}}(t^+) < \sigma^2_{\text{NIP}}(t)$ then the new channel estimates reduce the overall NIP and are considered beneficial for the convergence of the MUD. The new channel estimates can be combined with the updated ones, but, for simplicity, we replace some channel estimates provided by the AD algorithm with the new updated ones. Finally, the interference cancellation is performed following

$$\tilde{r}_k^t = \hat{h}_{t,k} y_d - \sum_{k' \neq k} \hat{h}_{t,k'} \tilde{x}_{d,k'},$$

$$\sigma^2_{k,t} = \text{var} \left( \hat{h}_{t,k} y_d - \sum_{k'=1}^{K_a} \hat{h}_{t,k,k'} \tilde{x}_{d,k'} \right)$$

in preparation for the next, $(t+1)$th iteration.

\subsection{D. Performance Metric}

By $\Pr(E_k)$ we denote the probability of error in receiving user $k$’s message, where the error event $E_k = \{ v^k \notin L \} \cup \{ u^k = v^k, k \neq i \}$ and $L$ is the list of the decoded messages. The performance indicator PUPE is given by the average number of the incorrectly received messages

$$P_e = \frac{1}{K_a} \sum_{k=1}^{K_a} \Pr(E_k).$$

(14)

This includes the error due to the missed users. The CRC, which is the part of the payload error correction, is implicitly used to detect false alarms.

\section{III. Proposed Feedback Mechanism}

\subsection{A. Baseline Feedback Systems}

Fundamentally, there are two basic approaches to perform the feedback in classical multi-user systems. In both approaches, the receiver explicitly informs a group of users about their detection/decoding status. In the positive-only feedback approach this group comprises of successful users (correctly detected and decoded users), while in the negative-only feedback this group consists of detected but incorrectly decoded users. In this section, we review these two basic feedback systems in the context of URA and use them as a baseline to compare with our proposed feedback mechanism.

1) Positive-Only Feedback (POF) Approach: In this approach, once the AD and MUD algorithms output the decoded users’ data, the BS receiver constructs a feedback signal to acknowledge all successful users of their decoding status. The decoding success of a user on the feed-forward link is declared after checking the cyclic redundancy check (CRC) bits which are a part of the error-correction encoding. For the purpose of forming the feedback signal, the receiver utilizes the preamble sequences $a_k$ selected by the users on the feed-forward link

$$x = \sum_{k \in \mathcal{S}} a_k,$$

(15)

the set of the successfully decoded active users is denoted by $\mathcal{S}$. We also define the set of the users that are detected but failed in decoding and denote it by $\mathcal{F}$. The set of users that are missed (not detected) by the activity detection algorithm is denoted by $\mathcal{M}$.

Each active user processes the feedback signal $x$ broadcasted by the BS on the feedback link. If it detects its signature $a_k$ in $x$, then a successful decoding acknowledgement is delivered to the $k$-th user. Otherwise, the active user declares failure of detection/decoding and re-transmits its packet. Note that in the POF approach the re-transmitting users are both the missed and the incorrectly decoded users, i.e., the set $\mathcal{M} \cup \mathcal{F}$. Hence, all failed users have a chance to be recovered. Since most of the users are successful, high number of non-orthogonal preamble sequences is expected in (15). That can cause high sequence interference and complicate the feedback detection process at the active users’ side.

2) Negative-Only Feedback (NOF) Approach: In this approach, the feedback message contains only the preamble sequences of the failed users. The feedback signal in this case is given by

$$\varphi = \sum_{k \in \mathcal{F}} a_k.$$

(16)

User $k$ processes the feedback message $\varphi$ by looking for the presence of its preamble sequence $a_k$ in the feedback message to decide whether it is intended by the BS for re-transmission. Unlike the POF approach, the NOF signal contains a smaller number of preamble sequences $a_k$ but the complexity can still be significant in case of a large user pool. Note that the missed users (those in the set $\mathcal{M}$) will not re-transmit as they infer positive feedback acknowledgment. The latter causes higher feedback error compared to the POF approach. In the
next section we propose a novel threshold-based feedback mechanism that simultaneously delivers positive and negative feedback and, at the same time, addresses the majority of the successful users.

B. Single-Threshold Feedback (STF)

Fig. 2 (a) shows the three classes of the active user messages after the AD and MUD processing on the feed-forward link. Some users are missed by the AD (set $M$), some of the detected users are not decoded successfully by the MUD (set $F$), and, finally, some users are both successfully detected and decoded (set $S$). We recall that on the feed-forward link the AMP channel estimation algorithm uses the received signal and the preamble sequence dictionary matrix $A$ to iteratively produce the estimates of the user channel gains $\hat{h}_k$ (corresponding to the preamble sequences $a_k$, $k \in \{1, 2, \ldots, 2^{B_n}\}$, where $2^{B_n}$ is the number of possible preambles). At each iteration, the AMP algorithm applies component-wise thresholding with $cT$ (as shown in Section II-B (5)) to its estimated channels vector $\hat{h}$ in order to reject small entries which are perceived to be the noise rather than active users’ channels. This reduces the presence of false alarms which can otherwise cause AMP to diverge. The threshold is applied actively to suppress noise spikes assuming that all detected entries with magnitude below $cT$ are caused by the noise. The majority of AD algorithms follow the same thresholding strategy, and, therefore, our feedback strategy is applicable to a general thresholding AD algorithm.

We assume that the channel between a user and the BS satisfies the reciprocity condition, i.e., the channel gains on the feed-forward and feedback links are the same for a single feed-forward slot followed by a feedback slot. In fact the proposed technique can work with a milder assumption that just the absolute value of the channel gain is constant (approximately). Then, an active user $k$ is able to learn whether it has been missed by the AD algorithm on the feed-forward link if it knows the threshold $cT$ of the AD algorithm and an estimate of its channel $\hat{h}_k$ on the feedback link. That is, if $|\hat{h}_k| < cT$ then the $k$-th user was most likely missed by the AD algorithm on the feed-forward link, and is required to retransmit. While users below the AD threshold $cT$ are definitely missed by the AD, some users above the threshold are also missed due to noise and estimation errors. Negative feedback needs to be provided to these users as well. Note that instead of using the threshold $cT$ of the AD algorithm we can use a generalized threshold $\tilde{c}T$, where $c < \tilde{c}$, and $\tilde{c}$ can be chosen such that all users above $\tilde{c}$ are detected correctly by the AMP and all users below it are missed. Fig. 2 (b) depicts active user categorization with respect to the threshold $\tilde{c}T$.

The sets $\hat{M}$ and $\tilde{M}$ correspond to the missed users with channels above ($|\hat{h}_k| \geq \tilde{c}T$) and below ($|\hat{h}_k| < \tilde{c}T$) the threshold respectively. The set $\hat{M}$ denotes the set of the failed users above the threshold (detected by the AD but incorrectly decoded), while $\tilde{M}$ denotes the set of the failed users below the threshold. Finally, the set of successfully detected and decoded users above the threshold is denoted by $\tilde{S}$, while the set of successful users below the threshold is denoted by $\hat{S}$.

In the next sections we first show how the feedback is designed to instruct each user group with the necessary action to re-transmit or not to re-transmit. We then compute the corresponding cost functions and expected reduction in PUPE. Additionally, we design a categorization of the users around two thresholds instead of one and show that it can further reduce computational demands on the active user resources.

Once the active users acquire the knowledge of the threshold $cT$ and channels $h_k$, the user groups below the threshold, i.e., $\hat{F}$ and $\tilde{M}$, can re-transmit without reception of negative feedback messages addressed specifically to them. We call this an implicit negative feedback. The problem with such approach, however, is that the users in the set $\tilde{S}$ will re-transmit as well, since they also infer negative feedback. Therefore, direct positive feedback messages are provided to the users in $\tilde{S}$ so that they don’t re-transmit. Once the users above the threshold learn their channels and the threshold itself, they immediately infer positive feedback. The failed users in the set $\hat{F}$ will receive an explicit negative feedback commanding them to re-transmit.

The PUPE of the URA system without the feedback equals

$$P_e = \frac{|\hat{M}|}{K_a} = \frac{|\hat{F}| + |\hat{M}|}{K_a}.$$  \hspace{1cm} (17)

In the case of optimal threshold selection and perfect user’s channel knowledge, the users that belong to the groups $\hat{F}$ and $\tilde{M}$ will be recovered with the proposed STF approach. The error after feedback processing assuming a successful re-transmission is then reduced to

$$P_e = \frac{|\tilde{M}|}{K_a},$$  \hspace{1cm} (18)

since the users in $\tilde{M}$ infer positive feedback and do not re-transmit despite the fact that they fail on the feed-forward.
link. Note that this is highly unlikely for a user to fall into this group, as most of the miss-detection events in the AD algorithm occur as a result of low-power users treated as noise. In order to allow active users to acquire both channel and threshold knowledge to process the feedback as explained above, we construct the feedback packet \( x_f \), broadcasted by the BS on the feedback slot, according to

\[
x_f = [p \ c \tau \ p \ a]^{T}.
\]  

The first part of the feedback packet is a short pilot signal \( p \) utilized by the active users to obtain estimates \( \hat{h}_k \) of their channels. The second part is the broadcast pilot signal \( c \tau \) scaled by the threshold value \( c \tau \). This allows the users to estimate the threshold \( c \tau \). Alternatively, a quantized and encoded numerical value of \( c \tau \) can be included into the feedback packet. The third part, i.e., \( a \), is a superposition of the preamble sequences \( a_k \) of the users in \( \mathcal{F} \) and \( \mathcal{S} \). Formally we construct,\(^1\)

\[
a = \sum_{k \in (\mathcal{F} \cup \mathcal{S})} (\pm) a_k.
\]  

The \( k \)-th active user receives the feedback signal

\[
y_k = h_k x_f + z_k,
\]  

where \( z_k \sim \mathcal{C}(0, \sigma^2) \) is the AWGN vector, \( z_k = [z_k^{(1)}, z_k^{(2)}, z_k^{(3)}] \), and the SNR on the feedback link is \( \frac{\|a\|^2}{\sigma^2} \). The user then starts processing the first part (see (19)), i.e., the received pilot signal

\[
y_k^{(1)} = h_k p + z_k^{(1)},
\]  

(22) to estimate its channel. Algorithm 1 describes the decision process at the user end. The linear minimum mean square error (LMMSE) of the user’s channel is given by

\[
\hat{h}_k = p^* (pp^* + \sigma^2 I)^{-1} y_k^{(1)} = p^* \left( I - \frac{pp^*}{\sigma^2 + |p|^2} \right) \frac{y_k^{(1)}}{\sigma^2},
\]  

(23)

where the matrix inversion lemma is utilized to simplify the calculations and avoid a matrix inversion. The noise power \( \sigma^2 \) can be estimated using silence periods. The \( k \)-th user’s channel estimate obtained on the feedback link is denoted by \( \hat{h}_k \), while the estimate of the same channel (due to reciprocity assumption) performed by the BS on the feed-forward link is denoted by \( \tilde{h}_k \). Note that the pilot-dependent part of the feedback signal (such as \( pp^* \) and \( |p|^2 \)) can be pre-calculated and saved at the user devices to further reduce the complexity. Following the channel estimation, the users recover the threshold from the received signal

\[
y_k^{(2)} = h_k c \tau p + z_k^{(2)} = h_k c \tau p + z_k^{(2)}
\]  

(24) using the LMMSE

\[
\tilde{\tau} = \hat{h}_k p^* \left( I - \frac{\hat{h}_k p^*}{\sigma^2 + |\hat{h}_k|^2 |p|^2} \right) \frac{y_k^{(2)}}{\sigma^2}.
\]  

(25)

\(^1\)The signs \(+/-\) are optional. Even without a sign, a user with the threshold and channel knowledge, can infer/decide about re-transmission.

### Algorithm 1 User’s Decision Procedure For STF

**Input:** \( \hat{h}_k, c \tau, \gamma_k, \tilde{\gamma} \).

**Output:** re-transmit or not to re-transmit decision

if \( \text{Re}(\gamma_k) > \tilde{\gamma} \) then

if \( |\hat{h}_k| < c \tau \) then

1. do not re-transmit

else

1. re-transmit

end else

if \( |\hat{h}_k| \geq c \tau \) then

1. do not re-transmit

else

1. re-transmit

end

end

In order to finally decide on re-transmission, each user processes the targeted acknowledgement part \( a \) of the packet, i.e., the third part of the received signal

\[
y_k^{(3)} = h_k a + z_k^{(3)} = h_k a_k + h_k \sum_{k \neq k} a_k + z_k^{(3)}.
\]  

(26)

The \( k \)-th user correlates \( y_k^{(3)} \) with its signature \( a_k \) to produce the statistics

\[
\gamma_k = \frac{a_k^* y_k^{(3)}}{h_k}.
\]  

(27)

If \( \text{Re}(\gamma_k) > \tilde{\gamma}_k \), where \( \tilde{\gamma}_k \) is a certain preset threshold, the user decides that it detects a presence of the positive or negative targeted feedback. The number of signatures in \( a \) equals \( |\mathcal{S}| + |\mathcal{F}| \) and is threshold-dependant. Clearly, the lower is the signature density, the more accurate is the feedback detection. The complete detection algorithm is described by Algorithm 1. The processing cost at the active user side is defined as the number of users that have to process the entire feedback packet \( x_f \). For the case of single-threshold feedback approach it is given by \( C_{UE} = K_a \), and is threshold-independent, since all active users have to process the entire signal \( x_f \).

### C. Single-Threshold Negative Feedback (STNF)

In the STF approach we aim to inform successful users below the threshold about the success of their packets. The STNF approach presents an important modification of the STF where the targeted signal \( a \) contains only the signatures of the failed users above the threshold, i.e.,

\[
a = \sum_{k \in \mathcal{F}} a_k.
\]

All users with channels above the threshold follow the same strategy as of those in the STF and re-transmit if their signature is detected in \( a \). However, the users below the threshold re-transmit without further processing. Low-power users that are often missed of fail in the decoding on the feed-forward link may also suffer from erroneous
feedback detection. Hence, in a practical scenario where a large number of such users is present in the system, it is more beneficial if the users below the threshold always re-transmit. In addition, in case of STF the targeted feedback signal \( \hat{x} \) includes less messages and the detection of the feedback is simplified. In the next section, we propose a double-threshold feedback approach that also reduces the users’ processing cost.

### D. Double-Threshold Feedback (DTF)

In case of STF all active users process the entire received feedback signal \( [y_k^{(1)}, y_k^{(2)}, y_k^{(3)}] \) to decide whether to re-transmit or not. Since most of the users are detected and decoded correctly on the feed-forward link (\( \in \hat{S} \)), it would be more energy-efficient if they could acquire their status without processing the entire feedback signal. This reduces the cost of the feedback processing significantly at a small additional signalling overhead. To save the power at the users’ side, we propose a DTF mechanism that takes into account the fraction of users with high detection/decoding success.

Fig. 3 shows user categorization imposed by two thresholds \( \tilde{c}_1 \tau \) and \( \tilde{c}_2 \tau \), where \( \tilde{c}_2 > \tilde{c}_1 \). Unlike the STF case, all users with \( |h_k| < \tilde{c}_1 \tau \), i.e., in groups \( \hat{S}, \hat{F}, \) and \( \hat{M} \), always re-transmit without checking on targeted feedback messages in \( \hat{x} \). Similarly, all users with \( |h_k| > \tilde{c}_2 \tau \), i.e., in groups \( \hat{M}, \hat{F}, \) and \( \hat{S} \), refrain from re-transmission since they infer positive feedback. Only users between the two thresholds (i.e. with \( \tilde{c}_1 \tau < |h_k| < \tilde{c}_2 \tau \)) will process the entire feedback signal including the targeted feedback part \( \hat{x} \). The feedback signal \( x_1 \) for the double-threshold approach has the format

\[
x_1 = [p \quad \tilde{c}_2 \tau p \quad \tilde{c}_1 \tau p \quad \hat{x}]^T,
\]

where signal \( \hat{x} \) is given by

\[
\hat{x} = \sum_{k \in \hat{S}} a_k,
\]

and \( \hat{S} \) denotes the set of successful users between the two thresholds. The signal \( \hat{x} \) instructs the users in \( \hat{S} \) not to re-transmit. The sets of missed \( \hat{M} \) and failed users \( \hat{F} \) between the two thresholds infer implicit negative feedback and re-transmit. As in the case of the STF, all users utilize the first part of the received feedback signal (pilot) to estimate their channels. All users also process the second part of the feedback signal to acquire \( \tilde{c}_2 \tau \). The thresholds are selected in such a way that most users infer positive feedback and stop after the 2-nd stage. Only users that fall below the second threshold start processing the third block. After acquiring \( \tilde{c}_1 \tau \), even fewer users (in sets \( \hat{S}, \hat{F}, \) and \( \hat{M} \)) engage in processing of the final part \( \hat{x} \). The user’s cost, therefore, drops to

\[
C_{UE}^d = |\hat{S}| + |\hat{F}| + |\hat{M}|,
\]

which is threshold dependent and is significantly smaller than the user’s cost for the STF method which is the entire \( K_a \).

In the STF, the feedback errors (in case of ideal feedback reception) are caused by the set of missed users above the threshold, i.e., by the set \( \hat{F} \) only. In the DTF the detected/decoded users above \( \tilde{c}_2 \tau \) add to the feedback error which, assuming successful re-transmission, becomes

\[
P_e^d = \frac{|\hat{M}| + |\hat{F}|}{K_a} = P_e + \frac{|\hat{F}|}{K_a}.
\]

We note that the overall PUPE of the system with DTF is slightly higher than that of the STF for the same parameter settings. Table I summarizes and compares the costs the feedback approaches.

The feedback approaches discussed above necessitate utilization of the entire preamble signature. In case of short signatures, required to support large numbers of active users, the feedback packet consumes a significant portion of channel resources (channel uses), and reduces the system throughput. In order to conserve the available channel resources, we propose consideration of systems where only a portion of each user’s preamble sequence is used to construct the feedback signal \( \hat{x} \) instead of the full preamble sequence. Moreover, the shortened feedback further reduces the energy spent at the user’s side to process the feedback.

### E. Feedback Cost

Transmission of the feedback packet introduces extra latency due to the extra time needed before the new group
of users can start transmitting in the next feed-forward slot. The latency increase factor equals
\[ \ell = \frac{N_t + N_i}{N_t} - 1 \] (31)
where \( N_t \) is the length of the feedback packet in terms of channel uses. This length equals
\[ N_t = \alpha N_p + l_p, \]
where \( \alpha \) is the preamble signature utilization factor, in case only a part of the signature is used for the feedback. The segments of the feedback signal dedicated to enabling channel and threshold estimation are of length \( l_p \) (where \( l_p \approx 100 \) was used in experiments in Section V, and can, potentially, be lowered).

Given the numerical results presented in Section V, for the two systems detailed in Table III, the extra latency amounts to \( \ell = 23 - 28\% \) for the case of full signature utilization \( \alpha = 1 \) and \( \ell = 4\% \) for partial signature utilization \( \alpha = 0.1 \). The extra latency is equivalent to a decrease of the system’s throughput (by the respective percentage) or an increase in the required bandwidth. The transmitters also incur some energy costs related to the feedback reception and the respective signal processing.

Even though the feedback results in the above-mentioned costs, the overall equivalent SNR is significantly lower than the SNR required for the feed-forward only system (see Section V). The transmitters, on the other hand, will benefit from a deterministic feedback acknowledgement that may stop frequent repeated re-transmissions.

### IV. Performance Analysis

In this section, we present a framework for performance analysis of a URA system which includes both the feed-forward and the feedback links. Based on the feed-forward link parameters \( K_a, P_e \) and \( \frac{E_b}{N_0} \), the proposed analysis predicts the PUPE for the system that includes feedback, the equivalent \( \frac{E_b}{N_0} \), and the average number of the new users \( K_a \) entering the system at every time slot. The analysis utilizes a state machine representation of the packet states and state transitions due to feedback and re-transmission. By deriving the transition probabilities, a system designer can predict the overall system performance based on the feed-forward link information, and adjust the system parameters to attain the required performance level. In this analysis setup each user is allowed to re-transmit its packet only one time, based on the feedback received after the initial transmission attempt. In general, we could allow any specific number of re-transmissions.

Fig. 4 depicts a state machine where \( S, F, \) and \( M \) denote the successful, failed decoding, and missed (miss-detected)

**Table I**

| Feedback approach | Number of Signatures | \( C_{UP} \) |
|-------------------|----------------------|--------------|
| STF               | \(|S| + |F|\)          | \( K_a \)    |
| DTF               | \(|S| + |F|\)          | \(|S| + |F| + |M|\) |
| STNF              | \(|F|\)               | \( K_a - |S| - |M| - |F|\) |

states of a packet after the receiver processing of the feed-forward link. The arrows show possible packet transitions to other states after the feedback and re-transmission (in case the re-transmission occurs). For example, a missed packet which is at state \( M \) at time \( i \) may transition to a successful state \( S \) at time \( i + 1 \) with probability \( P_{M \rightarrow S}[i+1] \). Since our analysis computes steady state quantities, we will use the compact notation \( P_{M \rightarrow S} \). The transition probabilities depend on the type of feedback as well as AD and MUD algorithms used in the feed-forward link. We consider the four types of feedback discussed above. For all types of feedback, a successful user (in group \( S \)) is expected not to re-transmit. However, because of the presence of other users’ signatures in the feedback message, the feedback may be received incorrectly and a successful user may erroneously re-transmit with a non-zero probability \( P_{fa} \). Moreover, for all types of feedback, a failed user (in group \( F \)) or missed user (in group \( M \)) is expected to re-transmit. In the event of erroneous feedback processing the no re-transmission event occurs with probability \( P_{md} \).

#### A. Analysis Framework

In this section, we explain the steps we make to derive an analytical expression of the overall system PUPE \( P_{e}^{c} \) accounting for feedback and re-transmission, the corresponding number of new users \( K_a \) and the equivalent \( E_b/N_0 \).

1) **PUPE After Feedback:** The average number of users in error in a feed-forward link is given by \( K_e = K_a P_e \). The set of erroneous users is composed of both failed and missed users, i.e., \( F \cup M \).

In the ideal case of perfect feedback reception and error-free re-transmission, the transition probabilities \( P_{M \rightarrow S} = P_{F \rightarrow S} = 1 \). Moreover, in such ideal case no successful user transitions to a failed or missed state: \( P_{S \rightarrow M} = P_{S \rightarrow F} = 0 \). However, in practice, because of errors in processing of the feedback and re-transmission, there will be a user subset of \( M \cup F \) that will either miss their feedback with probability \( P_{md} \) or not succeed in their re-transmission. Similarly, some users in the set \( S \) miss-interpreter the feedback (false alarm on the feedback with probability \( P_{fa} \)), are unsuccessful in re-transmission, and change the state from \( S \) to \( F \) or \( M \). These error events of miss-interpretation of the feedback combined with re-transmission errors impact the ultimate number of users in error given by

\[
K_e = K_e - E\{F[i] \cap S[i+1]\} - E\{M[i] \cap S[i+1]\} + E\{S[i] \cap F[i+1]\} + E\{S[i] \cup M[i+1]\} \\
= K_a P_e - K_a \left( P_{F \rightarrow S} + P_{M \rightarrow S} \right) + K_a \left( P_{S \rightarrow F} + P_{S \rightarrow M} \right).
\] (32)
The overall system's PUPE after the feedback and possible re-transmission is given by

\[ P_e^f = \frac{K_a}{K_a} = P_e - P_{F\rightarrow S} - P_{M\rightarrow F} + P_{S\rightarrow F} + P_{S\rightarrow M}. \]  

(33)

2) Number of New Users: Since the number of users at every feed-forward slot is constant and equals \( K_a \), we assume that the number of new users entering the system is reduced to allow for the re-transmitting users. As we explained above, the set of re-transmitting users includes both the intended users that acquired correct feedback signals commanding them to re-transmit, and the unintended users that falsely re-transmit. Hence, the average number of new users equals \( K_a = K_n - K_r \), where \( K_r \) is the average number of re-transmitting users. For any type of feedback, the number of new users is given by

\[ K_r = K_a P_{fa} - K_a (1 - P_{md}) \]

\[ + K_a \left( P_{S\rightarrow F} + P_{F\rightarrow F} + P_{F\rightarrow M} + P_{M\rightarrow F} + P_{M\rightarrow M} \right) \]  

(34)

where \( K_a \) denotes the number of successful users per slot, and \( K_a + K_c = K_a \). A subset of successful users will re-transmit with probability \( P_{fa} \) and only a fraction of \( (1 - P_{md}) \) of the number of erroneous users \( K_a \) successfully infers the feedback and re-transmits. The last term in (34) adds back the number of users that have already re-transmitted from the previous time slot but either failed or were missed again in the current time slot. These users will not be allowed to re-transmit again (since they reached the maximum number of their re-transmission attempts).

3) Equivalent \( E_b/N_o \): Because the re-transmission process resembles a repetition code, the average \( E_b \) is higher than the \( E_b \) allocated for the single feed-forward slot, that is

\[ E_b = \frac{K_a \bar{E}_b + K_c \bar{E}_b}{K_a} = \left( 1 + \frac{K_c}{K_a} \right) \bar{E}_b. \]  

(35)

4) Analysis Steps: In the following sections we derive the analysis details for the four types of feedback protocols presented in this paper. To do so, we first find the transition probabilities, in terms of the feedback probabilities \( (P_{fa}, P_{md}) \). We assume that the SNR \( \bar{E}_b/N_o \) of the feed-forward link, and the corresponding \( P_e \) are given. These parameters are used to find the overall PUPE \( P_e^f \), the equivalent \( E_b/N_o \) and the number of new users \( K_n \) that enter the system at every time slot. The predicted quantities are confirmed to be very close to the experimental ones, for both cases of signature utilization, full and partial, as we will see in Section V.

B. Analysis of Positive-Only Feedback

We start with the successful users group. On the feed-forward link, a user success occurs with probability \( P_s = 1 - P_e \). The user remains successful with probability

\[ P_{S\rightarrow S} = \Pr(S[i+1] \cap S[i]). \]

Note that the PUPE \( P_e \) results from two mutually exclusive events: failed decoding (with probability \( P_f \)) and miss-detection of the user (with probability \( P_M \)). Hence, \( P_e = P_F + P_M \). Since the operations of the MUD at times \( i + 1 \) and \( i \) are independent of each other, and the re-transmission occurs in the event of feedback false alarm we have

\[ P_{S\rightarrow S} = (1 - P_e) \times P_{fa} (1 - P_e) = P_{fa} (1 - P_e)^2. \]  

(36)

The probability of the feedback false alarm, where a successful user re-transmits equals

\[ P_{fa}^+ = \Pr(K[i] \cap S[i]), \]  

(37)

where \( K[i] \) is the set of re-transmitting users. Similarly,

\[ P_{S\rightarrow F} = \Pr(S[i] \cap F[i+1]) = (1 - P_e) P_{fa}^+ P_F, \]  

(38)

\[ P_{S\rightarrow M} = \Pr(S[i] \cap M[i+1]) = (1 - P_e) P_{fa}^+ P_M. \]  

(39)

The same approach is applied to compute the rest of the state transition probabilities. The results are shown in Table II. The feedback signal received by a successful user \( k \) on the feedback link is given by

\[ y_k = h_k a_k + a_n, \]

where \( n \sim \mathcal{C}(0, \sigma_n^2) \) and the SNR of the feedback link equals \( \frac{||\gamma||^2}{\sigma_n^2} \). The output of the match filtering with signature \( a_k \) computed by the \( k \)-th user’s receiver on the feedback link is given by

\[ r_k = a_k^* y_k h_k^{-1} = |a||^2 + \sum_{k \neq k \in S} a_k^* a_k + \frac{n}{h_k} \approx 1 + \sum_{k \neq k \in S} a_k^* a_k = 1 + \zeta, \]  

(40)

where \( \zeta = \zeta + j \zeta \) and \( \zeta \sim \mathcal{C}(0, \sigma_z^2) \) is the interference caused by the signatures of the other users. The ratio \( \frac{n}{h_k} \) is omitted since the interference is typically dominant, i.e., \( \zeta \gg \frac{n}{h_k} \). For the case when the \( k \)-th user signature is not present in the feedback, the matched filter output equals

\[ r_k = \sum_{k \neq k \in S} a_k^* a_k + \frac{n}{h_k} \approx \zeta. \]  

(41)

Hence, the detection rule is given by

\[ \gamma_k = \text{Re}(r_k) \]

(42)

decision to re-transmit = \{ true, \( \gamma_k \geq \lambda \) \} \}

otherwise, \( \{ false \} \}

(43)

where \( \lambda \) is the decision threshold. The probability that a successful user (i.e., for \( \gamma_k \sim \mathcal{N}(1, \frac{\zeta}{\sigma_z^2}) \)) falsely detects the feedback and re-transmits (false alarm) equals

\[ P_{fa} = \Pr(\gamma_k < \lambda) = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{\lambda - 1}{\sigma_z} \right) \right]. \]

The signature interference power \( \sigma_z^2 \) is given by

\[ \sigma_z^2 = \frac{K_a (1 - P_e) - 1}{N_p}. \]  

(44)
Similarly, the probability of miss-detection of the feedback is given by

\[ P_{md} = 1 - \Pr \{ K[i] \cap \mathcal{F}[i] \} = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{\lambda}{\sigma_p} \right) \right], \]  

(45)

where \( \text{erf}(\cdot) \) is the error function. The probability of the event of correct feedback detection (\( \gamma_k \sim \mathcal{N}(0, \frac{\sigma_S^2}{2}) \)) equals \( \Pr \{ K[i] \cap \mathcal{F}[i] \} = \Pr (\gamma_k < \lambda) \), where missed and failed users re-transmit – otherwise they don’t and count towards miss-detection. Using (33), and after simplification, the PUPE after the feedback equals

\[ P_\alpha = P_e - P_e(1 - P_e)(1 - P_{md}) + P_e(1 - P_e)P_{fa} + (1 - P_e)P_{fa}P_{M}, \]  

(46)

where \( P_\alpha \leq P_e \). Using similar arguments, the number of re-transmitting users in (34) simplifies to

\[ K_r = K_a(1 - P_e)^2P_{fa} + K_a(1 - P_{md})(P_e - P^2_M - P^2_e) \]  

(47)

that can be used to calculate the number of unique users in the system, as well as the equivalent \( E_b/N_0 \). Finally we note, that the analysis for the NOF follows the same lines. The transition probabilities are given in Table II.

We note that for a fixed SNR \( E_b/N_0 \) the transition probabilities (see Table II) as well as (46) are linear combinations of the miss \( P_{md} \) and false alarm probabilities \( P_{fa} \). These are, in turn, functions of the feedback reception threshold \( \lambda \) and the feedback threshold(s) \( \bar{\tau} \), i.e., \( P_{md}(\lambda, \bar{\tau}) \) and \( P_{fa}(\lambda, \bar{\tau}) \). This allows for optimization of the PUPE via selection of \( \lambda \) and \( \bar{\tau} \).

### C. Analysis of Single-Threshold Feedback

We assume that each user estimates both the channel \( h_k \) and the threshold \( \bar{c}\tau \) accurately such that \( h_k = \tilde{h}_k \) and \( \bar{\tau} = \bar{\tau} = \bar{c}\tau \). The received feedback signal of the \( k \)-th user, is given by

\[ y_f = \tilde{h}_k \sum_{k\in S[i]\cup \mathcal{F}[i]} a_k + n, \]  

(48)

where \( n \) is the noise at the feedback channel. If \( k \in S[i] \), i.e., it is successfully decoded and has a channel \( |h_k| > \bar{\tau} \), the user is instructed not to re-transmit. If \( k \in \mathcal{F}[i] \), with channel such that \( |h_k| > \bar{\tau} \) the user is instructed to re-transmit. Each user then performs (as in formulas (42) above)

\[ \gamma_k = \text{Re}(r_k), \quad \text{where } \gamma_k^0 \sim \mathcal{N}(0, \frac{\sigma_S^2}{2}), \gamma_k^1 \sim \mathcal{N}(1, \frac{\sigma_S^2}{2}). \]  

(49)

The power of the other users signatures correlated with the signature of interest is given by

\[ \sigma_S^2 \approx \frac{K_aP_c\tilde{P}_h + K_a(1 - P_c)\tilde{P}_h}{N_p}, \]  

(50)

since the channel at time \( i+1 \) is independent of \( i \) and Rayleigh distributed, i.e.,

\[ \tilde{P}_h = \Pr(|h_k| > \bar{\tau}) = e^{-\bar{\tau}^2}, \]  

\[ \tilde{P}_h = \Pr(|h_k| < \bar{\tau}) = 1 - e^{-\bar{\tau}^2}. \]  

(51)

Assuming the AD algorithm on the feed-forward link fully converged, the residual noise of the preamble part satisfies \( \tau^2 \approx \sigma_{n_0}^2 \) and, therefore, \( \bar{\tau} = \bar{c}\sigma_{n_0} \). The false alarm of the feedback (successful users’ re-transmission probability) is then given by

\[ P_{fa} = \Pr(\gamma_k^0 > \lambda) = 1 - \frac{1}{2} \left( 1 + \text{erf} \left( \frac{\lambda}{\sigma_S} \right) \right). \]  

(52)

The probability of feedback false alarm by

\[ P_{md} = \Pr(\gamma_k^1 < \lambda) = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{\lambda - 1}{\sigma_S} \right) \right). \]  

(53)

and both of these probabilities are used to derive the transition probabilities in Table II, the overall PUPE \( P_\alpha \) (46), and the corresponding number of re-transmitting users \( K_r \) (47).

The STNDF approach is analyzed via the same methodology as the STF. The users with channels below the threshold re-transmit with probability \( P_{fa} \). The users above the threshold experience the same rate of false alarms and missed detection as in the NOF. The transition probabilities are given in Table II.

### Table II: Transition Probabilities of the Feedback Protocols

| Protocol | Feedback | STF | STNDF |
| --- | --- | --- | --- |
| P\text{S-S} | \( P_a(1 - P_e)^2 \) | \( P_a(1 - P_e)^2 \) | \( (1 - P_e)^2(\tilde{P}_h, \tilde{P}_h) \) |
| P\text{S-F} | \( P_a(1 - P_e)P_f \) | \( P_f(1 - P_e)P_{fa} \) | \( P_f(1 - P_e)[\tilde{P}_h, \tilde{P}_h(1 - P_{md})] \) |
| P\text{S-M} | \( (1 - P_f)P_{fa}P_M \) | \( (1 - P_f)P_{fa}P_M \) | \( (1 - P_f)(\tilde{P}_h, \tilde{P}_h)P_M \) |
| P\text{S-S} | \( P_f(1 - P_e)(1 - P_{md}) \) | \( P_f(1 - P_e) \times (1 - P_{md}) \) | \( (1 - P_f)\tilde{P}_h(1 - P_{md}) \) |
| P\text{S-F} | \( (1 - P_{md})P_f^2 \) | \( P_f(1 - P_{md})P_f^2 \) | \( P_f(1 - P_{md}) + \tilde{P}_h(1 - P_{fa}) \) |
| P\text{S-M} | \( P_f(1 - P_{md})P_{fa} \) | \( P_fP_{fa} \times (1 - P_{fa}) \) | \( (1 - P_f)\tilde{P}_h(1 - P_{fa}) \) |
| P\text{M-S} | \( P_M(1 - P_e)(1 - P_{md}) \) | \( P_M(1 - P_e) \times (1 - P_{md}) \) | \( P_M(1 - P_e)(\tilde{P}_h, \tilde{P}_h)P_M \) |
| P\text{M-F} | \( P_M\tilde{P}_h(1 - P_{md}) \) | \( P_M\tilde{P}_h \times (1 - P_{md}) \) | \( P_M\tilde{P}_hP_M + \tilde{P}_hP_M \) |
| P\text{M-M} | \( P_M^2(1 - P_{md}) \) | \( P_M^2 \) | \( P_M^2\tilde{P}_h \times (1 - P_{md}) \) |
In this section we study the effects of the presented feedback algorithms on the system performance. We show that the threshold-based feedback can significantly enhance the reliability (reducing PUPE), without providing an explicit feedback to all URA users. We demonstrate that the STF approach has the complexity of the NOF but with the gain of the POF. More importantly, the proposed feedback conserves the spectral efficiency of the system and can efficiently operate with partial signatures. Additionally, the proposed feedback mechanism allows a system designer to reduce the overall $E_b/N_0$ required to attain various levels of the overall system reliability (e. g. PUPE = 0.05). The number of iterations required for the AD and the MUD algorithm are set to insure full convergence of these algorithms.

We consider a system where a user who receives (or infers) a negative feedback is allowed to re-transmit. The user messages are encoded with a Polar code and CRC. At slot $i$, $i = 1, 2, \ldots$ a set of $K_a$ active users is allowed to transmit. The set is comprised of re-transmitting users from the previous slot $i-1$, and the new users. The signal $y[i]$ received on the feed-forward link is then split into $y_u[i]$ which undergoes AD and $y_d[i]$ for MUD processing. The feed-forward link processing starts with the AD that produces the set of preamble signatures, permutations, and scrambling sequences for the detected users. These are then utilized by the MUD algorithm to generate the lists of successfully decoded users $S$ and failed users $F$. These lists are created with the aid of the CRC of the decoded codewords assuming that a failed user has CRC $\neq 0$. The receiver then uses the preamble sequences of users from sub-sets of the above lists to construct the feedback packet.

The feedback packet is broadcasted over the same media and each active user starts processing the received feedback packet. Each active user estimates its channel and the threshold. In case of STF each user correlates its (full or partial) signature with $\bar{\gamma}$ of the feedback packet to infer its status. Active users that detect negative acknowledgment re-transmit in slot $i+1$. Here, for simplicity, we consider a system where each user can re-transmit only once. In general, the proposed system can allow for any number of possible re-transmissions.

Block fading Rayleigh channel model is considered, where the feed-forward and feedback channel are reciprocal for each slot $i$, but are independent from slot to slot. Table III summarizes feed-forward and feedback settings for two types of systems we consider. The System $A$ is designed for smaller numbers of active users, while the System $B$ allows for higher numbers of active users, and applies the standard URA parameters used in [1] widely adopted in URA literature for comparison purposes. We choose the feed-forward $E_b/N_0$ such that the overall PUPE, accounting for re-transmissions, equals $P^{I_c}_e \approx 0.05$. We then calculate the overall SNR $E_b/N_0$ and the average number of new users $K_a$ that transmit at each time slot. The number of experiments for each Monte Carlo simulation equals 300 to ensure the stability of the observed PUPE. We follow the settings given in Table III and vary the feed-forward $E_b/N_0$ (to scan an SNR range with fine granularity) to find the $E_b/N_0$ that leads to $P^I_c \approx 0.05$.

Fig. 5 compares the minimum SNRs required to support various numbers of new users per slot for the proposed STF (green curves), POF (red curves), and NOF (blue curves) approaches for the System $A$, for both full (100%) (with filled markers) and partial (10%) (hollow markers) signature utilization. The dashed curves present the results of the analysis described in Section IV. The feed-forward system performance is given by the black curve. The results demonstrate that, regardless of the type of feedback or level of signature utilization, the feedback provides significantly better performance in terms of required $E_b/N_0$ compared to the feed-forward link. The POF with full signature utilization permits the recovery of almost all failed and miss-detected packets. The same level of performance is provided by the proposed STF with an optimized threshold. The NOF, however, requires 2dB extra power, on average. On the other hand, the NOF can support the highest number of users overall, since it is the most conservative in terms of re-transmissions. Even though the POF allows for recovery of failed and missed users, it suffers from a higher rate of false alarms on the feedback link due to the large number of signatures in the feedback message. That forces several successful users to re-transmit and reduces the number of new users entering the system.

Once the signature length is reduced, the POF performance suffers a serious degradation. The POF contains the highest number of signatures in the feedback message and signature reduction diminishes the processing gain and makes the feedback detection much more challenging. The NOF and STF performance suffer some degradation with STF being the most efficient among all of the techniques. Finally, we note that the proposed analysis approach allows to predict the performance of all the considered systems very accurately.

Fig. 6 plots the PUPE achievable with POF (red circles), NOF (blue squares), and STF (green diamonds) for $E_b/N_0 = 7$ dB. The curves with filled and hollow markers correspond to 100% and 10% signature utilization respectively. The PUPE of the feed-forward link is given by the black curve with hollow diamonds. The results show that all types of feedback provide a significant reduction in terms of PUPE compared to the feed-forward link. The shortened (10%) signature feedback leads to higher PUPE compared to the full signature feedback, where STNF is the least affected feedback type and POF experiences the largest deterioration.

Fig. 7 presents the performance for the System $B$ that is able to support much higher numbers of active users compared to the System $A$. All types of feedback with different signature lengths still require considerably lower $E_b/N_0$ to support...
Fig. 5. The minimum $E_b/N_0$ required to support $K_u$ new users for the System A.

For the case of shortened signatures (10%), the loss experienced by the POF due to the large number of signatures in the feedback message is even more pronounced than for the System A. The performance of the shortened version of the NOF coincides with that of full-signature NOF for smaller numbers of users (up to 350) since small signature density of the feedback message provides nearly perfect feedback detection for smaller numbers of users. For the larger numbers of users the performance of the shortened NOF starts to suffer degradation due to the overwhelmed feedback detection.

The STNF performance suffers almost no degradation due to shortening. The STNF inherits the smaller signature density from the STF and also has the ability to recover most missed and failed users - at the cost of some re-transmission of successful users. These features explain the ability of the STNF to provide the performance which is very close to the performance of the full-signature POF. Again, we note that the results of the performance analysis derived in Section IV are very close to the experimental results for all systems.
Finally we note that the results include the impact of collisions. In case of a collision, the higher-power user typically dominates, can re-transmit and be recovered, while the lower-power user’s packet is lost and contributes to the error.

A. Double-Threshold Feedback

Fig. 8 presents the multi-slot transmission results of the STF and DTF feedback schemes for the System B. Since the all users below the lower threshold \( c_1 \tau \) re-transmit regardless of their status, the number of re-transmitting users in the DTF system is higher than for the STF method. This translates into an increase in the equivalent required \( E_b/N_0 \) compared to the STF approach. As shown in Fig. 8, the gap widens as the number of users increases. Nevertheless the double-threshold approach provides a significant \( E_b/N_0 \) gain over the feed-forward link. The probability \( P_c = C_{dUE}/K_a \) defines the frequency of the utilization of the user’s correlator used for the feedback reception. All active users in the STF, POF, and NOF approaches always have to use their correlator which leads to \( P_c = 1 \). In the STNF approach, the users below the threshold always re-transmit and, therefore, the resulting \( P_c \) is slightly less than 1. For the DTF, however, \( P_c \) and the respective UE cost \( C_{dUE} \) are significantly lower and fall in the range of \([0.4 - 0.6]\). This reduces the overall processing energy by 50%, in average.

VI. CONCLUSION AND FUTURE WORK

In this paper we focus on a systematic approach to feedback design for URA systems. Our proposed feedback format allows the URA users to acquire the knowledge about the status of the packet transmission in a low-complexity fashion, suitable for low-power miniature transceivers. We have also shown that the typical positive-only and negative-only feedback approaches are not suitable for URA systems with large numbers of active users. We have demonstrated that the feedback improves the system performance in terms of the PUPE and the SNR required to support various numbers of active users. Our proposed single-threshold feedback method reduces the number of preambles included into the feedback signal and enables a simple correlation detection receiver at the active user side. Our second proposed algorithm, the double-threshold approach, allows most of the users to skip the processing of the main part of the feedback signal and save the power even further. Finally, we show that the presented analytic framework closely predicts the performance of URA systems with feedback.

REFERENCES

[1] Y. Polyanskiy, “A perspective on massive random-access,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2017, pp. 2523–2527.
[2] V. K. Amalladinne, J.-F. Chamberland, and K. R. Narayanan, “A coded compressed sensing scheme for unsourced multiple access,” IEEE Trans. Inf. Theory, vol. 66, no. 10, pp. 6509–6533, Oct. 2020.
[3] V. K. Amalladinne, A. Kumar Pradhan, C. Rush, J.-F. Chamberland, and K. R. Narayanan, “On approximate message passing for unsourced access with coded compressed sensing,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2020, pp. 2995–3000.

[4] A. Fenger, P. Jung, and G. Caire, “SPARCs and AMP for unsourced random access,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2019, pp. 2843–2847.

[5] A. Pradhan, V. Amalladinne, A. Vern, K. R. Narayanan, and J.-F. Chamberland, “A joint graph based coding scheme for the unsourced random access Gaussian channel,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2019, pp. 1–6.

[6] E. Nassaji, M. Bashir, and D. Truhachev, “Unsourced random access over fading channels via data repetition, permutation, and scrambling,” IEEE Trans. Commun., vol. 70, no. 2, pp. 1029–1042, Feb. 2022.

[7] D. Truhachev, M. Bashir, A. Karami, and E. Nassaji, “Low-complexity coding and spreading for unsourced random access,” IEEE Commun. Lett., vol. 25, no. 3, pp. 774–778, Mar. 2021.

[8] N. H. Mahmood, R. Abreu, R. Böhme, M. Schubert, G. Berardini, and T. H. Jacobsen, “Uplink grant-free access solutions for URLLC services in 5G new radio,” in Proc. 16th Int. Symp. Wireless Commun. Syst. (ISWCS), Aug. 2019, pp. 607–612.

[9] J. Choi and J. Ding, “Network coding for K-repetition in grant-free random access,” IEEE Wireless Commun. Lett., vol. 10, no. 11, pp. 2557–2561, Nov. 2021.

[10] J. Östman, R. Devassy, G. C. Ferrante, and G. Durisi, “Low-latency short-packet transmissions: Fixed length or HARQ?” in Proc. IEEE Globecom Workshops (GC Wkshps), Dec. 2018, pp. 1–6.

[11] J. Choi, “On throughput of compressive random access for one short message delivery in IoT,” IEEE Internet Things J., vol. 7, no. 4, pp. 3499–3508, Apr. 2020.

[12] Q. Zhang, S. Jin, and H. Zhu, “A hybrid-grant random access scheme in massive MIMO systems for IoT,” IEEE Access, vol. 8, pp. 88487–88497, 2020.

[13] B. Ren, T. Ekpenyong, and X. Pan, “Random access feedback method, random access processing method, base station and terminal,” U.S. Patent 10 986 670, Apr. 20, 2021.

[14] S.-H. Lee, B. C. Jung, and S.-W. Jeon, “Successive interference cancellation with feedback for random access networks,” IEEE Commun. Lett., vol. 21, no. 4, pp. 825–828, Apr. 2017.

[15] X. Yang, X. Wang, and J. Zhang, “Compressed sensing based ACK feedback for grant-free uplink data transmission in 5G uMTC,” in Proc. IEEE Vehicular Tech. Conf. (VTC Spring), Jun. 2016, pp. 1–5.

[16] T. Tang and R. W. Heath, “Opportunist feedback for downlink multiuser diversity,” IEEE Commun. Lett., vol. 9, no. 10, pp. 948–950, Oct. 2005.

[17] D. Jung, R. Kim, and H. Lim, “Asynchronous medium access protocol for multilayer MIMO based uplink WLANs,” IEEE Trans. Commun., vol. 60, no. 12, pp. 3745–3754, Dec. 2012.

[18] G. K. Facenda and D. Silva, “Efficient scheduling for the massive random access Gaussian channel,” IEEE Wireless Commun., vol. 19, no. 11, pp. 7598–7609, Nov. 2020.

[19] J. Ding and J. Choi, “Triangular non-orthogonal random access in mMIMO systems,” IEEE Trans. Commun., vol. 68, no. 11, pp. 6918–6931, Nov. 2020.

[20] J. Kang and W. Yu, “Minimum feedback for collision-free scheduling in massive random access,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2020, pp. 2989–2994.

[21] A. E. Kalström R. Kotaba, and P. Popovski, “Common message acknowledgments: Massive ARQ protocols for wireless access,” IEEE Trans. Commun., vol. 70, no. 8, pp. 5258–5270, Aug. 2022.

[22] R. Kotaba et al., “How to identify and authenticate users in massive unsourced random access,” IEEE Commun. Lett., vol. 25, no. 12, pp. 3795–3799, Dec. 2021.

[23] L. Valentinì, M. Chiani, and E. Paolini, “Massive grant-free access with massive MIMO and spatially coupled replicas,” IEEE Trans. Commun., vol. 70, no. 11, pp. 7373–7350, Nov. 2022.

[24] J. R. Ebert, K. R. Narayanan, and J.-F. Chamberland, “HashBeam: Enabling feedback through downlink beamforming in unsourced random access,” in Proc. 56th Asilomar Conf. Signals, Syst., Comput., 2022, pp. 692–697.

[25] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” Proc. Nat. Acad. Sci. USA, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[26] A. Maleki, L. Anitori, Z. Yang, and R. G. Baraniuk, “Asymptotic analysis of complex LASSO via complex approximate message passing (CAMP),” IEEE Trans. Inf. Theory, vol. 59, no. 7, pp. 4290–4308, Jul. 2013.

Murwan Bashir (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Dalhousie University, Canada. He is currently a Post-Doctoral Researcher at Dalhousie University. His research interests include statistical signal processing, machine learning, channel estimation, underwater acoustic communication, and unsourced random access.

Ehsan Nassaji (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Tabriz University, Tabriz, Iran, and the M.S. degree in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2013. He is currently pursuing the Ph.D. degree in electrical engineering with the Department of Electrical and Computer Engineering, Dalhousie University, Canada. His current research interests are wireless communications and fiber-optic communications.

Dmitri Truhachev (Member, IEEE) received the Ph.D. degree in electrical engineering from Lund University, Sweden. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Dalhousie University, Canada. Before joining Dalhousie University, he held postdoctoral and research associate positions at the University of Alberta, Canada. His research interests include error-correction codes, spatial graph coupling, multiuser communications, fiber optical communications, and underwater acoustic communications.

Alireza Bayesteh (Member, IEEE) received the B.Sc. and M.Sc. degrees from the Sharif University of Technology, Iran, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2008. From 2008 to 2009, he was a Post-Doctoral Fellow at the University of Waterloo. From 2009 to 2011, he was a member of the technical staff at Research in Motion (currently BlackBerry Ltd.). Since 2011, he has been with Huawei Canada, Ottawa, where he is currently a Senior Principal Engineer. His research interests include various aspects of air interface design for 6G wireless communications.

Monirosharieh Vameghestahbanati (Member, IEEE) received the Ph.D. degree in electrical engineering from Carleton University, Ottawa, ON, Canada, in 2020. She is currently a Senior Wireless Research Engineer with the Huawei Technologies Canada Research Centre, Ottawa, and also an Active Member of the IEEE Young Professionals. Her research interests include advanced physical-layer technologies for future wireless systems. Her Ph.D. thesis was nominated for the Senate Medal for outstanding academic achievement. She was a recipient of numerous awards, including the Ontario Trillium Scholarship from 2013 to 2017 and the Outstanding Teaching Assistant Award at Carleton University from 2017 to 2018. She is also a co-inventor of more than 12 patents.