User Activity Detection in Massive Random Access: Compressed Sensing vs. Coded Slotted ALOHA

Veljko Boljanović, Dejan Vukobratović
Department of Power, Electronic and Communications Engineering
University of Novi Sad, Serbia
Email: veljkoboljanovic@gmail.com, dejanv@uns.ac.rs

Petar Popovski, Čedomir Stefanović
Department of Electronic Systems
Aalborg University, Denmark
Email: {petarp,cs}@es.aau.dk

Abstract—Machine-type communication services in mobile cellular systems are currently evolving with an aim to efficiently address a massive-scale user access to the system. One of the key problems in this respect is to efficiently identify active users in order to allocate them resources for the subsequent transmissions. In this paper, we examine two recently suggested approaches for user activity detection: compressed-sensing (CS) and coded slotted ALOHA (CSA), and provide their comparison in terms of performance vs resource utilization. Our preliminary results show that CS-based approach is able to provide the target user activity detection performance with less overall system resource utilization. However, this comes at a price of lower energy-efficiency per user, as compared to CSA-based approach.

I. Introduction

Reservation-based access protocols are standardly used in mobile cellular systems, where the first step is detection/identification of the (a priori unknown) set of active users, based on which the assignment of time-frequency resources and the subsequent data transmissions take place, cf. [1], [2]. The user activity detection in mobile cellular networks is slotted ALOHA (SA) based, which is a random access algorithm characterized by the simplicity of implementation, while suffering from low performance. Specifically, the throughput of slotted ALOHA in the basic, collision channel model, is upper bounded by $1/e$. Such performance may prove sufficient when the load of the cellular access network (i.e., the number of accessing devices) is low; however, it will pose a significant bottleneck for the foreseen number of accessing devices in the scenarios pertaining to the Internet-of-Things [3].

The main reason behind the modest throughput of slotted ALOHA is due to the resources wasted on collision slots. In particular, it is assumed that the access point is unable to decode any transmission occurring in a collision slot, making use only of the slots containing a single transmission (singleton slots). Obviously, throughput performance could be boosted through development of reception techniques that are able to exploit collisions, complemented by transmission strategies that foster such operation.

In this paper, we review two advanced approaches for user activity detection in random access scenarios, both being candidate solutions to be used in 5G access networking. The first approach is compressed-sensing (CS) based [4], and the second is coded slotted ALOHA (CSA) [5]. In CS, all active users transmit simultaneously during the activity detection phase (i.e., their transmissions collide/overlap completely), where a transmitted signal uniquely identifies the user transmitting it, and the common receiver uses an CS recovery algorithms for the reconstruction of the set of active users. On the other hand, in CSA, users transmit replicas of their unique identifiers in randomly chosen slots of the activity detection phase. The receiver relies on singleton slots to start decoding process, which is then upheld by application of successive interference cancellation (SIC) that removes replicas of decoded transmissions, potentially reducing collision slot to singletons and enabling new round of decoding. We discuss the key assumptions and characteristics of both frameworks and compare their efficiency in terms of resources and energy-expenditure required to achieve target detection performance. Asymptotic performance limits are reviewed and finite-length numerical simulations are provided for both frameworks. Preliminary results show that compressed-sensing based approaches provide the required user activity detection performance with less overall resource utilization. However, this comes at a price of exploiting a user behavior that leads to higher per-user resource utilization, and thus lower energy-efficiency, as compared to coded slotted ALOHA approach.

The rest of the paper is organized as follows. In Sec. II, we provide background on CS and CSA approaches that underlie the considered user activity detection schemes. In Sec. III, CS and CSA-based approaches for user activity detection are presented and placed within a common framework. Sec. IV reviews asymptotic results, while Sec. V provides numerical results. Finally, Sec. VI concludes the paper.

II. Background

A. Compressed Sensing

The CS framework allows undersampling the signal and then reconstructing it successfully, given that the signal being sampled is sparse, either in its canonical form, or in some transform domain. More precisely, suppose that $\mathbf{x} \in \mathbb{C}^N$ is a k–sparse signal, i.e., it contains k non-zero components, and $\mathbf{y} \in \mathbb{C}^m$ is its sampled version. Then, the ideal (noiseless) CS undersampling process can be expressed in the following way:

$$\mathbf{y} = \mathbf{A}_{cs}\mathbf{x} \qquad (1)$$

where the sampling matrix $\mathbf{A}_{cs} \in \mathbb{C}^{m \times N}$ models the sampling process. The design of the matrix $\mathbf{A}_{cs}$ is the fundamental

problem of CS. Specifically, $\mathbf{A}_{cs}$ should have the properties such as coherence or restricted isometry property. Some simple ensembles of random matrices, e.g., those with real entries sampled from Gaussian or binary entries sampled from Bernoulli distribution, are among suitable choices for $\mathbf{A}_{cs}$.

The expression (1) represents an underdetermined system of linear equations ($m < N$), solvable due to the sparsity of the vector $\mathbf{x}$. Note that the vector $\mathbf{y}$ is a linear combination of k columns of matrix $\mathbf{A}_{cs}$. Thus, to reconstruct $\mathbf{x}$ from its samples $\mathbf{y}$, we need to solve the inverse problem: i) to find the support set of $\mathbf{x}$ (i.e., the indices of the k columns included in the linear combination), and ii) the non-zero values of $\mathbf{x}$ (i.e., the weights that define the linear combination). A basic optimization method for solving (1) is l_0-minimization:

$$\min \left\{ \|\mathbf{z}\|_0 : \mathbf{A}_{cs}\mathbf{z} = \mathbf{y} \right\} \tag{2}$$

where $\|\mathbf{z}\|_0$ is the number of nonzero elements of vector $\mathbf{z}$. Solving (2) is a known NP-hard problem [8]. Alternatively, the relaxation of l_0-minimization is used, called l_1-minimization or basis pursuit, described in [9].

In this paper, for CS reconstruction, we use well-known greedy iterative algorithm called Orthogonal Matching Pursuit (OMP) [10]. OMP aims to find indices of nonzero elements of $\mathbf{x}$ by adding one index to the current support set at each iteration. In addition, proper choice of indices minimizes the residual which eventually leads to the convergence of OMP. Our choice for OMP is motivated by its well-understood performance, including its asymptotic behavior in terms of sufficient number of samples required for reliable detection [11]. However, one can easily replace OMP in our arguments in the rest of the paper with some other CS recovery methods, such as approximate message passing (AMP) [12]. Finally, we note that the model in (1) can be easily extended to a noisy scenario, where additive Gaussian noise vector $\mathbf{n} \in \mathbb{C}^m$, whose components are i.i.d. Gaussian entries with variance σ^2, is added to the right-hand side of (1).

B. Coded Slotted ALOHA

CSA is an umbrella term for SA-based family of random access protocols in which: (i) users contend by transmitting replicas of their packets sent in randomly selected slots, (ii) each packet replica contains pointers to all other replicas of the same packet, (ii) once a replica is decoded, the IC is used to remove the other related replicas, thereby propelling new iterations of decoding and replica removal [5], [13]. In the basic variant of CSA, called irregular repetition slotted ALOHA (IRSA), the slots are organized in frames, and users contend by selecting a random subset of slots in a frame in which replicas are transmitted. It was shown that, for the collision channel model, the described iterative successive interference cancellation (SIC) process is analogous to iterative erasure decoding of low-density parity check (LDPC) codes, motivating the use of modern coding theory tools to design and analyze CSA schemes. In the rest of the paper, we restrict our attention to IRSA. In the idealized scenario, we assume
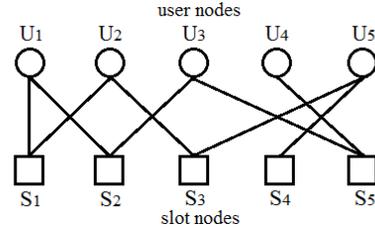

Fig. 1. An example of the bipartite graph with $k = 5$ active users and $M = 5$ slots. The process of decoding starts off with detection of the singleton slot S_4. When replica of user U_5 is successfully decoded, interference from user U_5 can be eliminated from all corresponding slots, i.e. edges from U_5 to S_3 and S_4 can be removed. The next iteration starts similarly – slot S_3 is detected as the singleton slot and replica of user U_2 is decoded. Consequently, interference from U_2 is eliminated from S_1 and S_3. The process iterates until all edges are removed or none of singleton slots can be detected.

that a singleton slot is reliably decoded and that the SIC is ideal.

For the design of IRSA scheme, it is useful to represent the contention process via a bipartite graph, see Fig. 1, analogous to graph representation of LDPC codes [14]. The graph contains user nodes, one for each of the k active users, and slot nodes, one for each of M slots in the frame, and the edges correspond to transmissions of replicas. The number of replicas is determined by a user degree distribution $\Lambda(x) = \sum_i \Lambda_i x^i$, where $\Lambda_i, 0 \le i \le M$, represents the probability that a user sends i replicas. The average number of replicas per user is denoted by $\overline{\Lambda}$. The iterative SIC decoder that "peels off" the graph represents the process of transmission detection and decoding. Note also that, in this paper, we will represent the relationship between replicas and slots via adjacency matrix $\mathbf{A}$, where columns correspond to users and rows correspond to slots. The element $a_{i,j}$ of the matrix $\mathbf{A}$ is equal to one if the j-th user sends a replica in the i-th slot, otherwise, it is equal to zero.

III. USER ACTIVITY DETECTION: CS VS. CSA

A. System Model and Problem Formulation

As already outlined, the role of the user activity detection is to extract the set of active users in a cell in order to allocate them resources. We consider a model with a single receiver and N users, out of which k randomly selected users are active. In the user detection phase, we assume active users communicate to the receiver their unique identification codeword $\mathbf{a}_i, 1 \le i \le N$, where a user-to-codeword association is known to the receiver. Each active user sends its own codeword simultaneously with other active users within the time frame of duration $T_F = m$ symbols. The time frame period is divided into M slots, each slot containing $T_S = m/M$ symbols. We assume ideal synchronization among the users and the receiver is established at the symbol, slot and frame level. The received signal $\mathbf{y}$ at the receiver can be expressed as:

$$\mathbf{y} = \mathbf{A}_t \mathbf{x} + \mathbf{n}, \tag{3}$$

where $\mathbf{A}_t \in \mathbb{C}^{m \times N}$, $t = \{cs, csa\}$, is a matrix defined by the applied framework (explained later in the section), $\mathbf{x} \in$

$\mathbb{C}^N$ is the vector that determines the set of active users, and $\mathbf{n} \in \mathbb{C}^N$ is the additive Gaussian noise. The vector $\mathbf{x}$, which is equivalent in both frameworks, is k-sparse and its nonzero elements describe the active users. Thus, finding the support set of vector $\mathbf{x}$ is equivalent to the user activity detection. The nonzero elements may contain information about the channel gains, but for simplicity, we assume existence of the (ideal) uplink power control, which ensures all nonzero elements of $\mathbf{x}$ are equal. Consequently, in the noiseless scenario, vector $\mathbf{y}$ represents a scaled sum of columns of matrix $\mathbf{A}_t$.

In the following, through the definition of $\mathbf{A}_t$, we investigate CS– and CSA–based approaches for user activity detection. Our goal is to minimize resources required for reliable user activity detection, as detailed in Section IV.

B. CS-based Approach for User Activity Detection

In the CS-based user activity detection, introduced in [15], the model presented in (3) is specialized as follows. The signal at the receiver is given by (3), where $\mathbf{A}_t = \mathbf{A}_{cs} = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_N] \in \mathbb{C}^{m \times N}$ is the matrix containing user identification codewords as columns. The time frame is not slotted, i.e., $M = 1$, and every user's codeword $\mathbf{a}_i, 1 \leq i \leq N$ is of length m symbols. For simplicity, we assume the codewords are randomly-generated binary, BPSK modulated sequences of m BPSK symbols $\{+1, -1\}$. Thus $\mathbf{A}_{cs} \in \mathbb{R}^{m \times N}$ is a Bernoulli matrix with entries ± 1. In this paper, for convenience, the columns are normalized to unit norm. In this setup, the model in (3) becomes equivalent to the noisy version of the CS model in (1). Thus, using CS methodology, we can reconstruct the signal $\mathbf{x}$ and recover the set of active users (i.e., the support set of $\mathbf{x}$) using the CS recovery algorithms such as OMP. Moreover, one can use theoretical results to provide asymptotic analysis on the length of user codewords m required for reliable signal recovery, as reviewed in Section IV.

C. CSA-based Approach for User Activity Detection

In the CSA-based user activity detection, the model presented in (3) assumes different format of the user identification codewords, and consequently, different form of the matrix $\mathbf{A}_t = \mathbf{A}_{csa}$. In order to define the format of user identification codewords $\mathbf{a}_i$'s, we introduce the division of the time frame into M slots, where the length of each slot equals $T_S = \lceil \log_2(N) \rceil$ (binary) symbols (for convenience, we assume T_S divides m). In addition, each user is uniquely identified by its ID sequence $\mathbf{s}_i$ of length $l_s = \lceil \log_2(N) \rceil$ binary symbols that exactly fits a single time slot. To generate $\mathbf{a}_i$, the i-th user applies CSA approach, namely, it first generates the degree d_i from $\Lambda(x)$, randomly samples d_i out of M available slots in the time frame, and transmits d_i replicas of its own ID sequence $\mathbf{s}_i$ within the selected slots. Thus in general, a user identification codeword $\mathbf{a}_i$ consists of length-T_S sub-blocks, where only d_i sub-blocks are BPSK modulated symbol sequences, while the remaining $M - d_i$ represent all-zero sequences. Arranging all the user identification codewords as the columns of the matrix $\mathbf{A}_{csa}$, one obtains the structured
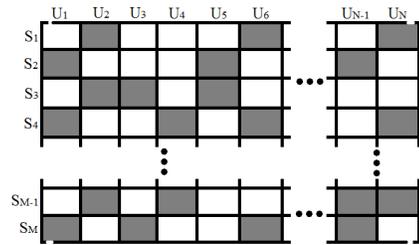

Fig. 2. An example of specific structure of matrix A_{csa}. Replicas of users' ID sequences are represented with grey color. Columns correspond to users and rows correspond to slots.

matrix of the form illustrated in Fig. 2. Finally, upon receiving the frame, the receiver decodes the user transmissions using iterative SIC decoder and eventually recovers user ID sequences from which the set of active users is reconstructed.

D. A Remark on User Activity Detection

We conclude this section with a general remark that detection of the active users can be made by exploitation both of the observations of the activity patterns and the received information. However, in CS, all active users are transmitting throughout the detection phase, i.e., their activity patterns are the same and thus do not carry any specific information. On the other hand, in CSA, the activity pattern of a user can be related to its identity, e.g., its identity can be used as the seed of a random number generator used to select slots in which the user is active. Yet, this prospect is typically not exploited, and the user identity is communicated only through information sent in the packet replicas; this approach is used in the paper.

IV. ASYMPTOTIC RESULTS

In this section, we compare the two frameworks in terms of required system resources needed to achieve asymptotically vanishing probability of error of user activity detection. The required resources are effectively equal to the length (i.e., the number of symbols) of user identification codewords. However, in the CSA framework, from a single user perspective, the majority of the M slots contain no signal, motivating us to consider energy efficiency as a comparison metric, by taking into account number of the actual symbol transmissions. We start by a review of the known asymptotic bounds on the codeword lengths required for the reliable detection.

A. Noiseless Model

CS-based framework: As mentioned in Section II, sampling matrices with i.i.d. Gaussian or Bernoulli entries as suitable choice for the CS framework. Most of results for OMP are based on certain assumptions about coherence among columns of matrix $\mathbf{A}_{cs}$. The coherence threshold, which ensures exact reconstruction of vector $\mathbf{x}$, was evaluated in [16]. However, as noted in [11], OMP would require an enormous number of measurements (symbols) in that case. A useful performance guarantee for OMP follows from the result below [11, Th. 6].

Theorem 4.1: Fix $\delta \in (0, 0.36)$, and choose $m \geq Ck \ln(N/\delta)$ where C is a constant. Suppose that $\mathbf{x}$ is an

arbitrary k-sparse signal in $\mathbb{R}^N$, and draw a random $m \times N$ Gaussian/Bernoulli[1] measurement matrix $\mathbf{A}_{cs}$ independent from the signal. Given the data $\mathbf{y} = \mathbf{A}_{cs}\mathbf{x}$, OMP can reconstruct the signal with probability exceeding $1 - \delta$.

The above result has been further refined in [17] for the case of Gaussian matrices, suggesting that $C \leq 20$. In asymptotic setting, i.e., when $N, k \to \infty$, it is possible to reduce the constant to $C \leq 4 + \eta$, where η is a positive number. Thus, the probability of imprecise support set estimation is reduced below δ if the number of measurements satisfies:

$$m \geq 4k \ln (N/\delta). \tag{4}$$

Numerical simulations in [11] revealed that (4) is slightly pessimistic and it turned out that OMP requires fewer measurements. For example, if $\delta = 0.01$ in the models with $N = 256, 1024$ users, required number of measurements is approximately $m \approx 2k \ln (N)$, instead of $m = 4k (\ln (N) + 4.6)$. Admittedly, as δ decreases, OMP requires more measurements, i.e. the expression for m changes. Furthermore, numerical evidence point out that constants for the Gaussian matrices are closely matched by the case of Bernoulli matrices [11].

CSA-based framework: In the CSA framework, the average number of active user transmissions per slot, also called the average system load, can be defined as $G = k/M$.

Following [13], for a fixed $\Lambda(x)$, an asymptotic ($N, m \to \infty$) threshold G^* is defined such that, if $G \leq G^*$, all transmissions will be successfully detected and decoded with probability approaching one. Otherwise, the probability of successful detection will asymptotically vanish. Therefore, all active users are reliably detected when $\frac{k}{M} \leq G^*$, i.e. when:

$$M \geq \frac{k}{G^*}. \tag{5}$$

By multiplying both sides with the number of symbols in a single ID sequence l_s, the total number of symbols m is obtained:

$$m = Ml_s \geq \frac{k}{G^*}l_s. \tag{6}$$

For example, for the node-oriented distribution $\Lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$, the threshold equals $G^* = 0.892$ and thus, (6) becomes $m \geq 1.121kl_s$.

B. Noisy Model

CS-based framework: Unlike the noiseless model where OMP algorithm iterates until residual $\mathbf{r}$ becomes zero, a different stopping rule for OMP has to be used in the noisy model. Furthermore, the coherence-based results for OMP require additional conditions for the nonzero elements of vector $\mathbf{x}$. A following theorem provides a performance guarantee for the correct detection of the support set of $\mathbf{x}$ in the presence of Gaussian noise [18, Th. 7].

---

[1]Note that the statement of the theorem holds for a more general class of so called admissible matrices defined in [11].

Theorem 4.2: Suppose $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I_n})$ and $\mu < \frac{1}{2k-1}$ and all the nonzero coefficients of $\mathbf{x}$ satisfy

$$|x_i| \geq \frac{2\sigma\sqrt{m + \sqrt{m \log(m)}}}{1 - (2k-1)\mu}. \tag{7}$$

Then OMP algorithm which utilizes the stopping rule $\|\mathbf{r}\|_2 \leq \sigma\sqrt{m + \sqrt{m \log(m)}}$ selects the true support set of $\mathbf{x}$ with probability at least $1 - 1/m$.

CSA-based framework: In general, the CSA framework implicitly assumes that the noise is addressed by (ideal) physical layer forward error correction (FEC) [13]. However, for the sake of counting m, we need to include the FEC rate penalty in our model. Asymptotically, as $N, k \to \infty$, the rate penalty follows directly from Shannon's channel coding theorem. Thus, in the case of binary-input additive white Gaussian noise (BI-AWGN) channel considered here, the additional multiplicative constant $1/C(\sigma)$ needs to be added to equation (6), where $C(\sigma)$ is the capacity of BI-AWGN channel with parameter σ, to account for the physical-layer FEC. For the non-asymptotic scenario, one can apply bounds following recent advents in finite-length information theory, however, we leave that case for future work.

V. Numerical Simulation

The goal of the receiver is to precisely determine the set of active users and thus correctly estimate the support set of the vector $\mathbf{x}$. For the finite number of measurements in the CS framework, the performance of the receiver is analytically intractable and is usually obtained using numerical simulations. Similarly, the performance in the CSA framework deteriorates if the number of slots is finite, with respect to the asymptotic thresholds indicated above.

In the case of limited resources, the reconstruction under OMP algorithm may experience two error events: i) the active user is not detected, and ii) the inactive user is detected as active. Probabilities of these two events are called *missed detection rate* and *false alarm probability*, respectively. On the other hand, in the noiseless CSA framework, there is just a single type of error event – the active user not being detected. This occurs when none of the user's replicas is successfully detected and decoded, i.e., when the user packet is lost, and the related probability is called *packet loss rate* (PLR). The terms PLR and missed detection rate can be used interchangeably, as they refer to the same type of error event.

Missed detection rate and false alarm probability were considered jointly in [11] as probability of unsuccessful detection of the exact support set of vector x. If we were interested in finding the exact support set, we would use that probability in the CS framework, and equivalently, the frame loss rate (FLR) in the CSA framework. However, in access networking, the user activity detection is followed by assignment of resources to the detected users, and the false alarms actually imply resources' loss. Thus, in the paper we separate among false alarms and miss-detections.
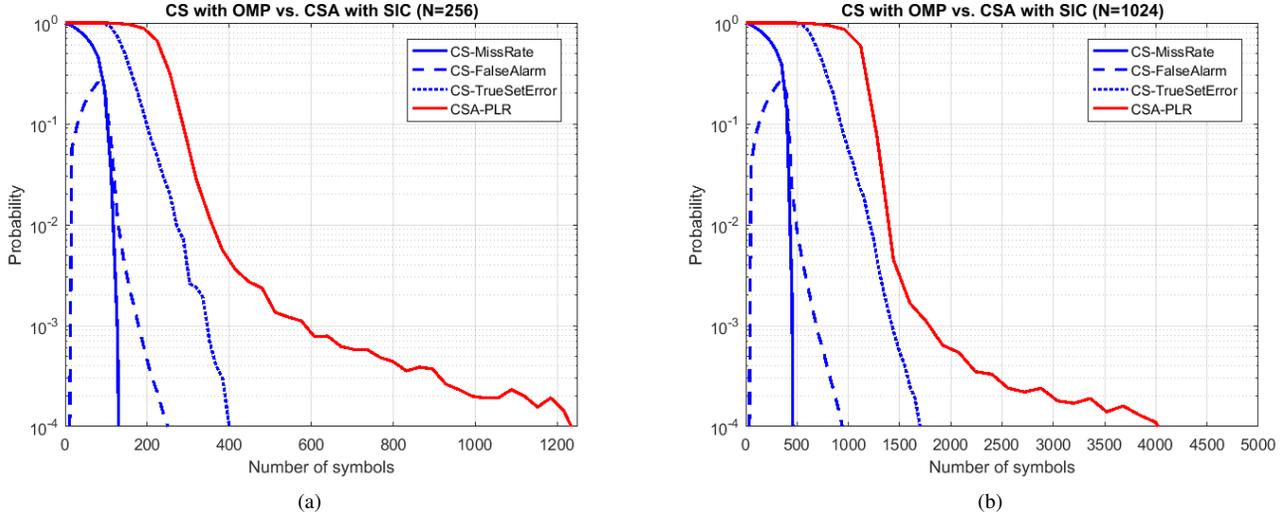
Fig. 3. Performance comparison in the noiseless models with $N = 256$, $k = 25$ (a) and $N = 1024$, $k = 100$ (b), and the user node degree distribution used in the CSA framework is $\Lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$.
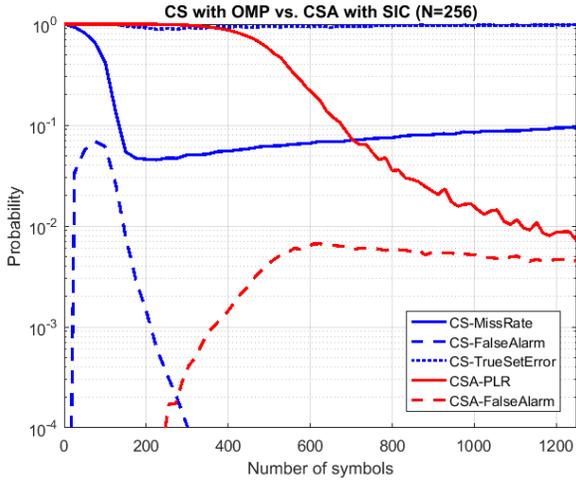

Fig. 4. Performance comparison in the noisy model with $N = 256$, $k = 25$, SNR = 10 dB, and the user node degree distribution used in the CSA framework is $\Lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$.

The numerical analysis is conducted in the models with $N \in \{256, 1024\}$ users, where the number of active users k is 25 and 100, respectively. In the CSA framework, the user degree distribution is set to $\Lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$ [13]. For both frameworks, the main metric we consider is the total number of symbols m in the transmitted frame necessary for the system to reach sufficiently low missed detection rate.

The comparison of two frameworks in the noiseless model when $N = 256$ and $k = 25$ is shown in Fig. 3a, and when $N = 1024$ and $k = 100$ in Fig. 3b; for each case, the results were derived using 10000 independent Monte Carlo trials. The results show that the CS framework with OMP algorithm is more efficient, i.e., it achieves a given miss detection rate (PLR) with fewer number of symbols m. Since miss detection

rate quickly tends to zero, the performance of OMP algorithm becomes solely affected by false alarm probability after certain number of symbols. Consequently, probability that the exact support set of vector x is not detected scales similarly to false alarm probability, as presented in Fig. 3a and Fig. 3b; it is easy to verify that, when false alarm probability is multiplied with $N - k$, i.e. with the number of inactive users, it becomes practically equal to probability of not finding the exact support set. On the other hand, PLR in CSA framework starts to decrease rapidly when the number of transmitted symbols m becomes roughly equal to kl_s, (i.e., when the number of slots M becomes approximately equal to number of the active users k), but then, for lower PLRs, it experiences an error floor. Obviously, the length of users' ID sequences in CSA plays a key role; as it is unknown which users will be active, all users are assigned a unique ID sequence of length $l_s = \lceil \log_2(N) \rceil$. However, when the probability that a user becomes active is small, using ID sequences with $l_s = \lceil \log_2(N) \rceil$ symbols does not seem reasonable, and one can use methods where ID sequence length scales with k instead of N. However, in this case, two or more active users may happen to use a same ID sequence, which necessitates a collision resolution in the space of ID sequences and which is out of the paper scope.

The same system model with $N = 256$ and $k = 25$ is examined in both frameworks in the presence of Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I_n})$ and the results are presented in Fig. 4. The results for $N = 1024$ and $k = 100$ are similar and they are omitted for brevity. In order to fairly compare two frameworks, we define signal-to-noise ratio as SNR per information bit. Since $N = 256$ users can be uniquely described with ID sequences $\mathbf{s}_i$ of length $l_s = 8$ bits, we can obtain mentioned SNR by dividing SNR per transmitted symbol with code rate in the corresponding framework. The code rates are $R_{cs} = \frac{l_s}{m}$ and $R_{csa} = \frac{1}{\Lambda} R_{fec}$ in the CS and the CSA frameworks

respectively. The rate R_{fec} is defined as $R_{fec} = \frac{l_s}{c_s}$ which gives $R_{csa} = \frac{1}{\bar{\Lambda}} \frac{l_s}{c_s}$. The value c_s is the length of ID sequence after FEC coding in the CSA framework (we use Reed-Muller codes with $R_{fec} = 1/2$).

In our simulation, we fix SNR = 10 dB. As Th. 4.2 suggests, the absolute value of the nonzero elements of vector $\mathbf{x}$ (i.e. SNR) needs to be sufficiently high in order to allow reliable user activity detection in the CS, even when the number of measurements (symbols) m goes to infinity. Nevertheless, we present the simulation results for SNR = 10 dB in order to show that insufficiently high SNR causes performance deterioration with respect to the noiseless case. Unlike in the noiseless model, probability of erroneous detection of the exact support is dictated here by high miss detection rate. Indeed, low SNR and a high threshold for the stopping rule from Th. 4.2 prevent some active users from being detected. Moreover, when SNR is fixed, increasing code rate in the CS results in poor performance of OMP algorithm. On the other hand, the CSA code rate remains fixed as m increases and its performance resembles that in the noiseless case. However, noise causes inactive users to be detected as active. Furthermore, the PLR curve is shifted to the right due to the FEC coding exploited in the CSA framework.

It may appear that using the total number of transmitted symbols m as a performance metric to compare CS- and CSA-based framework is, to some extent, unfair. Indeed, in the CS framework, matrix $\mathbf{A}_{cs}$ is a full matrix which means that k active users send a total of km symbols, or m symbols per active user. Therefore, there is a linear relationship between the total number of sent symbols and length of the transmission time frame. In contrast, matrix $\mathbf{A}_{csa}$ from the CSA framework is sparse. E.g., for the used distribution $\Lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$, the average number of replicas per user is $\bar{\Lambda} = 3.5$, thus, the total number of sent bits is $3.5k\lceil \log_2(N) \rceil$, or $3.5\lceil \log_2(N) \rceil$ per active user. Clearly, the total number of symbols increases logarithmically with N (and also with k if we assume that N and k scale linearly). Thus, for an arbitrary but fixed k, as the number of symbols m grows, the total energy expenditure as well as energy expenditure per active user remains constant in the CSA framework, while it grows linearly in the CS framework. This suggests that for low-delay applications, CS-based framework represents superior choice due to considerably smaller number of symbols m required to achieve satisfactory detection rate (PLR) performance. On the other hand, in the case when the energy resources of the reporting devices are limited, significantly reduced energy consumption per user for CSA-based approach may prove advantageous.

VI. DISCUSSION AND CONCLUSIONS

In this paper we examined CS- and CSA-based frameworks for the user activity detection in mobile cellular networks. We showed that CS-based framework is advantageous in terms of the number of system resources used, implying lower delay of the user activity detection phase. On the other hand,

CSA-based framework involves lower number of transmissions during the activity detection phase on the user basis.

Further extensions may include modification of CSA-based framework, such that the activity pattern and the transmitted information jointly determine the user identity. In this case, the length of the slots in CSA can be reduced, which may also reduce the overall length of the activity detection phase. On the other hand, this approach would require a reception algorithm that involves operations more complex than SIC.

REFERENCES

[1] 3GPP, "TS 36.213 - Physical layer procedures".
[2] 3GPP, "TR 36.321 - Medium Access Control (MAC) protocol specification".
[3] C. Bockelmann et al, "Massive Machine-type Communications in 5G: Physical and MAC-layer solutions", *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 59–65, Sep. 2016.
[4] C. Bockelmann, H. Schepker, and A. Dekorsy, "Compressive sensing based multi-user detection for machine-to-machine communication," *Trans. on ETT*, vol. 24, no. 4, pp. 389–400, Jun. 2013.
[5] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded Random Access: How Coding Theory Helps to Build Random Access Protocols," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 144–150, Jun. 2015.
[6] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 1021, Jan. 1949., Reprint (Classic Paper), *Proc. IEEE*, vol. 86, pp. 447–457, Feb. 1998.
[7] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[8] A. Foucart, H. Rauhut, "A Mathematical Introduction to Compressive Sensing," New York: Springer, 2013.
[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
[10] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Proc. 27th Annu. Asilomar Conf. Signals, Syst., and Comput.*, Pacific Grove, CA, Nov. 1993, vol. 1, pp. 40–44.
[11] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[12] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.
[13] G. Liva, "Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.
[14] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: M.I.T. Press, 1963.
[15] A. K. Fletcher, S. Rangan, and V. K. Goyal (2009), *On-Off Random Access Channels: A Compressed Sensing Framework*, [Online]. Available: http://arxiv.org/abs/0903.1022
[16] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
[17] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit: The Gaussian case," Caltech, Pasadena, CA, 2007, ACM Tech. Rep. 2007. [Online]. Available: http://authors.library.caltech.edu/27144/
[18] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, July 2011.