Sign-Compute-Resolve for Random Access

Jasper Goseling*‡, Čedomir Stefanović † and Petar Popovski†

* Stochastic Operations Research, University of Twente, The Netherlands
† Department of Electronic Systems, Aalborg University, Denmark
‡ Delft University of Technology, The Netherlands
j.goseling@utwente.nl, cs@es.aau.dk, petarp@es.aau.dk

***Abstract*—We present an approach to random access that is based on three elements: physical-layer network coding, signature codes and tree splitting. In presence of a collision physical-layer network coding enables the receiver to decode the sum of the information that was transmitted by the individual users. For each user this information consists of the data that the user wants to communicate as well as the user's signature. As long as no more than K users collide, their identities can be recovered from the sum of their signatures. A splitting protocol is used to deal with the case that more than K users collide. We demonstrate that compared to, for instance coded random access, our approach is significantly increasing the performance of the system, both in terms of user resolution rate as well as overall throughput of the system.**

I. Introduction

Uncertainty is the essential element of communication systems. In an information-theoretic setting, uncertainty is associated with noise, while in the context of communication protocols, uncertainty is associated with traffic (packet) arrivals at the users. A canonical example of the latter is seen in random access protocols, used for handling transmissions of users to a common receiver, e.g., a base station, over a shared wireless medium. Random access is necessary when the total number of users associated with the base station is very large, but at a given short time interval, the number of active users that have packets to transmit is small and a priori not known. Such is the case for wide-area networks of sensors, where each sensor has a sporadic traffic pattern. The goal of random access protocols is to enable each of the active users to eventually send her packet successfully.

Traditionally, random access protocols have been designed under the *collision model*: when two or more users transmit at the same time, a collision occurs and all involved transmissions are lost. In other words, collisions are considered as destructive and the information contained in them irrecoverable. Therefore, the objective of classical random access protocols, such as ALOHA [1] or splitting tree [2], is to ensure that each user gets the opportunity to send its packet without collision. Recently, a generalization of the collision model, obtained by including a more elaborate physical-layer model, brings in the possibility for Successive Interference Cancellation (SIC) and gives rise to a new class of protocols, termed coded random access [3]. The main feature of this model is that a collision is treated as a sum of packets and, instead of being discarded, it can be buffered and reused in a SIC-based decoding. We illustrate this through a simple example, in which the received signals in the first two slots are:

$$\begin{aligned} Y_1 &= X_1 + X_2 + Z_1, \\ Y_2 &= X_2 + Z_2, \end{aligned} \tag{1}$$

where X_1 and X_2 are the two useful signals (packets) and Z is the noise. The received signal Y_1 is buffered, and, if the X_2 is successfully decoded from the singleton slot Y_2, it can be subtracted (i.e., cancelled) from Y_1, effectively reducing the first slot to a singleton. The receiver proceeds by attempting to decode X_1 from the noisy signal $Y_1 - X_2 = X_1 + Z$.

An important constraint for protocol operation is that the useful signals must carry embedded pointers that inform the receiver where their replicas occurred. In the above example, the replica of X_2 in Y_2 has a pointer that indicates that another replica occurred in Y_1, so that the receiver, after decoding Y_2, also learns that a replica of X_2 can be cancelled from X_1. Another important aspect of SIC-based coded random access protocols it that the receiver buffers analog signals that contain noise. Hence, the uncertainty brought by the noise persists while the protocol resolves the uncertainty about the set of active users. This is fundamentally changed by applying the ideas of Physical Layer Network Coding (PLNC) to the problem of random access. The key idea in PLNC is to decode a function of multiple received signals, rather than decoding the individual signals.

Such operation is termed denoise-and-forward (DNF) [4], [5] or compute-and-forward (CF) [6]. Let us reuse the example (1) and assume that W_1 and W_2 represent the data bits that are mapped to the baseband signals x_1 and x_2, respectively. Upon receiving y_1 from (1), the base station stores the bitwise XOR $W_1 \oplus W_2$. If x_2 (W_2) is decoded from y_2, then W_1 is recovered by XOR-ing W_2 with the stored signal $W_1 \oplus W_2$. We can, therefore, say that the use of DNF (CF) removes the uncertainty of the noise from the protocol and deals only with the uncertainty of the contending set of users.

One of the main limitations of SIC-based coded random access is that the receiver must wait until it successfully decodes a packet from a singleton slot in order to start and maintain its operation. For example, let the receiver get the following signals in the first three slots:

$$\begin{aligned} Y_1 &= X_1 + X_2 + Z_1, \\ Y_2 &= X_1 + X_3 + Z_2, \end{aligned}$$

$$Y_3 \quad = \quad X_2 + X_3 + Z_3. \tag{2}$$

These are three (noisy) equations with three unknowns X_1, X_2, X_3 and, in principle the receiver should be able to recover all three signals by using the same techniques used to decode multiple streams in a MIMO transmission [7]. However, recall that in a setup with random access, the receiver has no knowledge about the set of transmitting users, e.g., in this case it does not know that X_1 is sent in slots 1 and 2, x_2 in slots 1 and 3, etc. It therefore needs to wait to receive, e.g., $Y_4 = X_1 + Z_4$, decode X_1 and learn from the embedded pointers in which other slots had X_1 been sent, such that it can be canceled. This fundamental limitation of SIC-based random access sets the motivation to introduce PLNC-based random access with signatures [8]. For the example (2) and the simplest case of $\mathbb{F}_2$ functions, in PLNC-based random access the receiver stores the following three (noiseless) digital binary signals:

$$\begin{aligned} V_1 &= W_1 \oplus W_2, \\ V_2 &= W_1 \oplus W_3, \\ V_3 &= W_2 \oplus W_3. \end{aligned} \tag{3}$$

In this way the receiver observes a noiseless XOR multiple access channel. The $\ell-$th user applies the following communication strategy: she prepends a *signature* W_ℓ^s, consisting of predefined number of bits, to the pure data W_ℓ^d in order to obtain W_ℓ. The signature is based on a code that has the following property: if at most K users transmit in a given slot, then from the *integer sum* of the signatures $W_1^s + W_2^s + \cdots + W_L^s$,[1] $L \leq K$, the receiver knows exactly which transmitters have contributed to the XOR-ed data stored in the present slot. In other words, the sum:

$$\sum_{\ell=1}^{L} W_\ell^s \tag{4}$$

is uniquely decodable if $L \leq K$. Referring to the example (3), the receiver will be able to decode the individual data already after the third slot. The receiver also detects if the number of users sending in a slot is larger than K, and the stored XOR combination cannot be used for decoding based on signatures. It may, however, still be used further in the decoding process, as detailed in the sequel.

In this paper we leverage the idea of PLNC-based random access and design a Contention Resolution Algorithm (CRA) that uses signatures. In contrast to collision avoidance protocols [1], contention resolution protocols [2], [10], [11] are efficient in terms of resolving collisions when they occur. The conventional contention resolution algorithms drive the set of contending users towards the state in which each user gets the opportunity to transmit without interference from the others. On the other hand, the use of signatures and PLNC generalizes the concept of collision by allowing the receiver to have a metadata (*i.e.*, the knowledge of the set of colliding users) about the observed collision. This feature fundamentally changes the objective of a CRA: the set of contending users should be driven in a state where the receiver gets a sufficient number of *equations* in the finite field in order to be able to decode the users' data. We provide details on the basic tradeoffs and mechanisms that needs to be considered for a CRA based on PLNC and signatures. The results show that the use of signatures is significantly reducing the average time required to extract useful information from the collisions and therefore improve the overall throughput of the system.

The idea of using SIC in framed ALOHA setting was first proposed in [12]. The analogies of SIC-based ALOHA with erasure-coding theory were identified in [13], establishing the paradigm of the coded random access that was further developed in [14]–[16]. It was shown that coded random access achieves throughputs that asymptotically tend to 1. The highest non-asymptotic throughputs were, so far, reported in [17]; e.g., when number of contending users is 1000, the expected throughput is 0.88.

The use of SIC in the contention resolution framework was first investigated in [18]. Here it was shown that enhancing the original tree-splitting scheme [2] with SIC doubles the asymptotically achievable throughput to 0.693. Another approach was suggested in [19], where SIC was employed over a set of partially split trees, and optimization was performed over the splitting strategy that favors fast SIC evolution. The reported throughputs for the presented design example in [19] are close to 0.8.

Finally, the use of PLNC for random access was studied in [20]–[25] in which it was assumed that the receiver knows which users are active in each slot. The use of physical-layer network coding and signature codes was considered in [26] for broadcast in networks. The combination of physical-layer network coding and signature codes for random access was introduced in [8]. In [26] as well as [8] it was assumed that the number of contending users is bounded. In the current work we leverage this assumption and design a CRA that can deal with any number of contending users by incorporating tree splitting.

The paper is organized as follows. In Section II we introduce our model. In Section III we present some results on PLNC, signature codes and tree splitting that will be used in the remainder. The proposed strategy is presented in detail in Section IV. The performance of the strategy is analyzed in Section V. The discussion and concluding remarks are given in Section VI.

II. Model and problem statement

We consider a system that has a total of M devices (users). Each of the users is assigned a unique identity from the set $\{1, \ldots, M\}$. For notational convenience in the remainder we assume that M is prime. Each user sporadically gets a data packet that needs to be sent to a receiver that is common for all users. The users that have data to transmit wait for a beacon sent by the common receiver, which marks the start of the contention process. In our model, the probability that a user

[1] This integer sum is obtained from the PLNC output $\bigoplus_{\ell=1}^{L} W_\ell^s$, by making use of a result by Nazer [9].

has a packet to transmit when the beacon is sent is p, where p is rather small, *i.e.*, $pM << M$.

Let $\mathcal{L}$ denote the set of contending users and let $L = |\mathcal{L}|$. The receiver does not know $\mathcal{L}$, otherwise the contention problem would have been trivial - the receiver would simply schedule the users from $\mathcal{L}$. The scheduling can be based by asking the users to transmit with rates that correspond to a certain point within the achievable region of an $L-$dimensional multiple access channel. However, the receiver does not know $\mathcal{L}$ and cannot make such a scheduling. In some cases our interest will be in the performance conditioned on a number of active users $L = \mathcal{L}$. Note that in that case only $L = |\mathcal{L}|$ is of importance. Since the packet arrivals across the set of users are independent, L has a binomial distribution: the probability that L users are active is denoted by $q(L) = P(|\mathcal{L}| = L) = \binom{M}{L} p^L (1-p)^{M-L}$. For notational convenience, let $q_0 = q(0) = (1-p)^M$. We will be interested in the probability of having L active users conditioned on the fact there is at least one. We denote this probability by $\hat{q}(L)$ and it readily follows that $\hat{q}(L) = P(|\mathcal{L}| = L||\mathcal{L}| > 0) = q(L)/(1-q_0)$. We will express some of our results in terms of $I_x(a,b)$, the regularized incomplete beta function, which is defined as $I(x;\ a,b) = B(x;\ a,b)/B(1;\ a,b)$ with $B(x;\ a,b) = \int_0^x t^{a-1}(1-t)^{b-1}dt$. The reason is that $\sum_{L=0}^{K} q(L) = I_{1-p}(M-K, K+1)$.

The data packet of each contending user consists of D bits. The channel coefficient between the $m-$th user and the receiver is h_m. Due to reciprocity, the contending device is capable to estimate the channel and *precode* its transmission by transmitting the signal $\frac{X_m}{h_i}$. The time starts at $\tau = 1$ and the $\tau-$th transmitted symbol by the $m-$th user is denoted by $X_m(\tau)$. Hence, at the $\tau-$th channel use, the receiver observes

$$Y(\tau) = \sum_{m \in \mathcal{L}} X_m(\tau) + Z(\tau), \tag{5}$$

where $\mathcal{L}$ is the set of active, contending users. $Z(\tau)$ is the Gaussian noise with unit variance. Each user transmits at the same rate (in bits/channel use) and one packet transmission has a duration of a *slot* that consists of N channel uses. At the end of each slot, the common receiver provides feedback to the users. This feedback is instantaneous, error free and received by all users. We do not impose any constraints on the amount of feedback that can be provided; we will explicitly specify how feedback is used later in the paper.

Further, we will assume that the signal of each user needs to satisfy an average power constraint in each round, *i.e.*,

$$\frac{1}{N} \sum_{\tau=1}^{N} |X_m(\tau)|^2 \leq P, \tag{6}$$

for all $m \in \{1, \ldots, M\}$. We will assume that $P > 1$, such that the Signal-to-Noise Ratio (SNR) is also larger than one, which is required to have a nontrivial computation rate over the multiple access channel, as seen in the next section. The reader may object that the actual transmitted power by the user can be much higher than P, since each user inverts the channel. This can be addressed by assuming that a user that observes a channel with $|h_m|$ lower than a threshold, does not join the set of contending users; in that case the probability p also accounts for the fact that the user channel is sufficiently strong, in addition to the assumption that the user has a packet to send.

For simplicity, as it is common in PLNC schemes, we assume that the channel input/outputs are real, *i.e.*, $X_m(\tau), Z(\tau), Y(\tau) \in \mathbb{R}$. The results are readily transferable to the case of complex symbols, by doubling the number of bits per channel use.

The goal of this paper is to devise a protocol that allows the receiver to retrieve both the identities and the data packets of the all active users. The constituent elements of the protocol are use of contention resolution mechanism across slots, dealing with randomness of the user activity pattern, and use of forward error correcting code within slots, dealing with noise. With respect to the latter, we ignore finite block length effects and assume that forward error correcting codes operate with zero error at any rate up to and including capacity. As a consequence, the task for the receiver is to recover all packets with zero error probability.

We are interested in the following performance parameters. By $S(L)$ we denote the expected number of slots that the protocol uses to resolve L contending users, where the expectation in $S(L)$ is w.r.t. the randomness in the contention resolution mechanism. By $R_{\text{res}}(L) = L/S(L)$ we denote the expected number of users that is resolved per slot. We are also interested in $\overline{R}_{\text{res}}$, obtained by averaging $R_{\text{res}}(L)$ over L, *i.e.*,

$$\overline{R}_{\text{res}} = \mathbb{E}[R_{\text{res}}(L)|L > 0] = \sum_{L=1}^{M} \frac{L}{S(L)} \hat{q}(L). \tag{7}$$

Further, we are interested in the effective number of bits that is transmitted across the channel per channel use (*i.e.*, , net rate), denoted by $\overline{R}_{\text{net}}(L)$. Taking into account that L users each transmit D bits in a total of $S(L)$ slots that each consist of N channel uses we have

$$R_{\text{net}}(L) = \frac{LD}{S(L)N} = R_{\text{res}}(L)\frac{D}{N}. \tag{8}$$

Finally, we are interested in the net rate averaged over L, *i.e.*, $\overline{R}_{\text{net}} = \mathbb{E}[R_{\text{net}}(L)|L > 0]$.

III. Preliminaries

A. Signature codes

We are interested in signature coding for the multiple access adder channel with q-ary inputs and additions over integers, when only up to K random users, out of total M users, are active. So far, there has been a lot of work investigating the case when the signature symbols are binary, *i.e.*, $q = 2$; a summary of the known asymptotic results has been presented in [27]. However, the case of general q has been significantly less studied.

In this paper, we adopt the Lindström's signature coding construction, as presented in [27, pp. 42 - 43]. The construction is designed for the case that the number of users M is a prime number; if M is not prime, one could design signatures

for the smallest prime number larger than M and use just M signatures. For easier exposition, we have assumed in Section II that M is prime. The construction is performed in the following way: choose integers s_i, $i = 1, ..., M$ such that

$$a^{s_i} = a + b_i,\ i = 1, ..., M, \tag{9}$$

where a is a primitive element of $\mathbb{F}_{M^K}$ and b_i, $i = 1, ..., M$, are elements of $\mathbb{F}_M$. It can be shown that: (i) integers s_i, $i = 1, ..., M$, exist, (ii) $0 < s_i < M^K - 1$ and, most importantly, (iii) the sums of subsets of s_i of at most cardinality K have unique values, *i.e.*,

$$\sum_{i \in U_1} s_i \neq \sum_{i \in U_2} s_i, \tag{10}$$

for any $U_1, U_2 \subset \{1, \ldots, M\}$, $|U_1| \leq K$, $|U_2| \leq K$ and $U_1 \neq U_2$.

The signature of user i is denoted by W_i^s. It is a sequence of symbols taking values from $\{0, \ldots, q-1\}$ in which the first symbol has value 1 and the remaining symbols are the q-ary representation of the integer s_i. Recall from Section I that the receiver will be dealing with the symbolwise addition (over the integers) of signatures. Since the first symbol is 1 for all users the receiver can immediately detect how many users are active and, therefore, determine whether the sum of signatures s_i is uniquely decodable. It is shown in [27] that in that case also the symbolwise sum of the W_i^s is uniquely decodable. The number of q-ary symbols in the above signature code is

$$\lceil \log_q(M^K - 1) \rceil + 1 \leq K \log_q M + 2. \tag{11}$$

Since all rates in this paper are expressed in terms of bits per channel use we express the length of W_i^s in terms of the equivalent number of bits. Let N_w denote the length of W_i^s in terms of bits. We have

$$N_w \leq \log_2 \left(K \log_q M + 2 \right) \leq (K + 2) \log_2 M, \tag{12}$$

which holds if $q \leq M$. From above, we have the following result.

Theorem 1 ([27]. pp. 43)**.** *If $q \leq M$ there exist q-ary K-out-of-M signature codes that satisfy*

$$N_w \leq (K + 2) \log_2 M. \tag{13}$$

B. Reliable physical-layer network coding

Another key ingredient of the random access strategy that is proposed in this paper is to employ physical-layer network coding (PLNC), *i.e.*, to organize the physical layer in such a way that the receiver can reliably decode sums of messages that are simultaneously transmitted by users. This requires a suitable choice of the forward error correcting codes as well as the decoding mechanism that is used by the receiver. In this section we provide a short introduction to physical-layer network coding and a result from [6] that will be needed later. There are various angles at which physical-layer network coding be approached, for instance denoise-and-forward [4] or compute-and-forward [6]. A survey of these and other approaches is given in [28] and [29]. In this paper we adopt the compute-and-forward framework as developed by Nazer and Gastpar in [6].

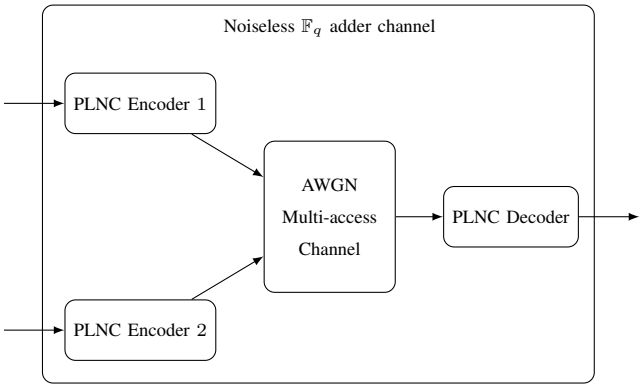

Fig. 1. Physical-layer network coding (PLNC) results in a noiseless $\mathbb{F}_q$ adder channel. ($K = 2$ users)

In order to formulate the result from [6] that we need in the remainder we consider an arbitrary number of L transmitters. User ℓ has data W_ℓ to transmit, where

$$W_\ell = (W_\ell(1), W_\ell(2), \ldots, W_\ell(\kappa)), \tag{14}$$

with $W_\ell(j) \in \mathbb{F}_q$, q prime. Each transmitter uses the same linear code F to encode the data into real-valued channel input of length N (*i.e.*, the length of a slot) that satisfies an average power constraint P. Let $X_\ell = F(W_\ell)$ denote the channel input of user ℓ. The decoder, upon observing $Y = \sum_{\ell=1}^{L} X_\ell + Z$ attempts to decode $(\bigoplus_\ell W_\ell(1), \ldots, \bigoplus_\ell W_\ell(\kappa))$, where $\bigoplus$ denotes adition in $\mathbb{F}_q$. In this sense, the receiver recovers a function (namely, the sum) of the original messages, which is why this approach is referred to as computation coding. In a sense, as illustrated in Figure 1, we turn the AWGN channel in a noiseless $\mathbb{F}_q$ adder channel.

We denote by R_{plnc} the rate of F, *i.e.*, $R_{\text{plnc}} = \kappa N^{-1} \log_2 q$ bits per channel use. We will refer to R_{plnc} as the computation rate and say that it is achievable if the probability of decoding erroneously can be made arbitrarily small by increasing n. The next result follows directly from the main result in [6].

Theorem 2 ([6], Theorem 1)**.** *For the standard AWGN multiple-access channel the following computation rate is achievable:*

$$R_{\text{plnc}} = \frac{1}{2} \log_2^+ (P). \tag{15}$$

The above result does not exactly match the achievable rate as given in [6, Theorem 1], which is $\frac{1}{2} \log_2^+ \left(\frac{1}{L} + P \right)$. Since we will be dealing with an unknown number of active users we use a computation rate that is valid for any number of active users.

The signature codes that we introduced in Section III-A operate over the adder channel with q-ary inputs and additions over the integers. It is important to note that the computation code as described above does not provide an adder channel, but instead provides additions in the finite

field $\mathbb{F}_q$. Therefore, we need an additional result that enables us to lift the computation code result to an integer adder channel. Such a result is provided by Nazer in [9]. The result states that once the receiver has successfully decoded $(\bigoplus_\ell W_\ell(1), \ldots, \bigoplus_\ell W_\ell(\kappa))$, *i.e.*, the sum over $\mathbb{F}_q$, it is also possible to recover the integer sum. To state the result more precisely, we create a mapping between the elements of $\mathbb{F}_q$ and the integers $\{0, 1, \ldots, q-1\}$. Since we consider q prime such a mapping is trivial. Indeed, additions in $\mathbb{F}_q$ are $\bmod q$ operations and the elements of $\mathbb{F}_q$ are naturally identified with the integers $\{0, 1, \ldots, q-1\}$. With slight abuse of notation we denote by $(\sum_\ell W_\ell(1), \ldots, \sum_\ell W_\ell(\kappa))$ the sums of the integers that are identified with the $\mathbb{F}_q$ elements $W_\ell(k)$. It was shown in [9] that at R_{plnc} as given in Theorem 2 the receiver can retrieve the integer sum $\sum_\ell W_\ell$.

C. Tree splitting

We briefly outline the basic binary tree-splitting algorithm under a collision model [2]. Let $\mathcal{L}$ denote the set of active users and $L = |\mathcal{L}|$, $1 \leq L \leq M$, denote the number of active users. In the first slot all L users transmit a packet. If $L = 1$ the receiver succesfully decodes the packet of the user and the contention period ends. If $L \geq 2$ a collision occurs and the receiver does not obtain any useful information. The users probabilistically split into two groups $\mathcal{L}_1$ and $\mathcal{L}_2$. The splitting is uniform at random and independent over users, *i.e.*, each user flips a fair coin to decide on the group to join. Both groups then contend for the medium in the same fashion: first the users from $\mathcal{L}_1$, then the users from $\mathcal{L}_2$. The splitting is done recursively, eventually leading to an instance in which only a single user is active and her transmission is successfully received. The algorithm continues until the transmissions of all active users from $\mathcal{L}$ are successfully received. By means of feedback after each slot the receiver informs the users whether there was a collision, a single or no transmission present, directing the future actions of the contending users.

The above described tree splitting and its variations were thoroughly analyzed in the literature, assessing the performance parameters such as throughput, delay, and stability. The work closest to ours is presented in [10], the most important difference being that we assess a generalized case when the collision occurs when $L > K$, where $K \geq 1$. In other words, the use of signatures in the proposed protocol allows for direct exploitation of the slots containing up to K user transmissions, and not just singleton slots. The related analysis, which also covers the special case $K = 1$, is presented in Section V.

IV. Proposed Strategy

We start with an overview of the proposed random access strategy. The strategy operates in rounds; in each round, the active users transmit the PLNC encoded concatenation of their signatures and payloads. Use of PLNC enables the receiver to reliably obtain the $q-$ary sums of the user transmissions, As long as there are at most K active users, the receiver is able to uniquely decode their signatures, detect which users are active and exploit this information to direct the active users towards solving the linear combination of their payloads.

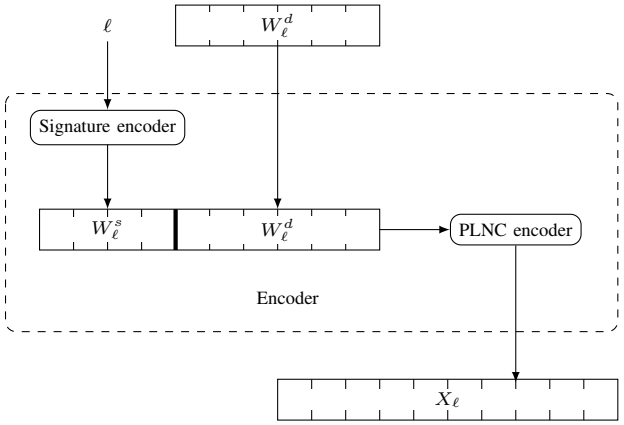

Fig. 2. Illustration of the encoder for user ℓ in one slot.

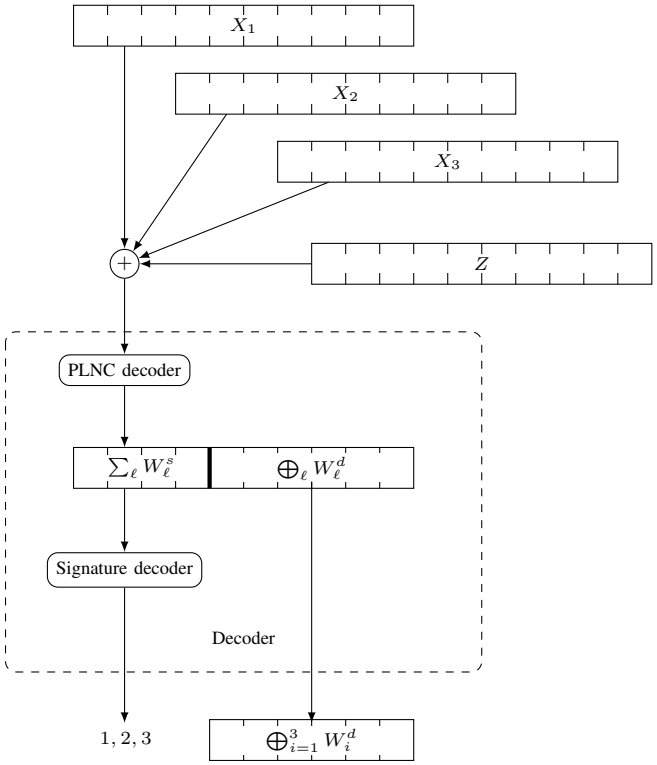

Fig. 3. Illustration of the decoder in one slot. ($L = 3$ users).

The receiver is also able to detect when more then K users are active. As explained in Section III-A this is enabled by the additional symbol that is prepended to the signatures and that is one for each user. By observing the integer sum that is decoded in this position the receiver directly learns the number of active users. If more than K users are active the receiver instructs the users to randomly split in two groups and the strategy is then executed in a recursive fashion for each of these groups. We proceed by presentation of the details.

A. Encoder

Let W_ℓ^s and W_ℓ^d denote the strings representing the signature and the data payload, respectively, of the active user ℓ. According to Theorem 1 the number of bits in a signature is not more than $(K+2)\log_2 M$. The concatenation of signature and payload $W_\ell = W_\ell^s \| W_\ell^d$ is used as the input of a PLNC encoder. Recall from Section III-B that the PLNC encoder applies a linear forward error correcting code, the same code F for all users. The output of the PLNC encoder, denoted by $X_\ell = F(W_\ell) = F(W_\ell^s \| W_\ell^d)$, is a channel input of user ℓ. The operation of the encoder of a single user in one block is illustrated in Figure 2.

B. Decoder

The receiver observes Y, which is a *real* sum sum of X_ℓ, $\ell \in \mathcal{L}$ and additive noise Z,

$$\sum_{\ell \in \mathcal{L}} F(W_\ell) + Z. \tag{16}$$

It uses a PLNC decoder to decode Y and obtain

$$\bigoplus_{\ell \in \mathcal{L}} W_\ell, \tag{17}$$

which decomposes into the sums of the signatures $\bigoplus_{\ell\in\mathcal{L}} W_\ell^s$ and the sums of the codewords $\bigoplus_{\ell\in\mathcal{L}} W_\ell^d$. Recall from Section III-B that, once we have obtained the sum $\sum_{\ell\in\mathcal{L}} W_\ell^s$ over the finite field $\mathbb{F}_q$, we can also interpret the elements of W_ℓ as integers and recover the integer sum $\sum_{\ell\in\mathcal{L}} W_\ell^s$.

Since the first symbol in the signature of all users is 1, we directly obtain $L = |\mathcal{L}|$, the number of active users, from the first symbol in $\sum_{\ell\in\mathcal{L}} W_\ell^s$. If $L \le K$, by the property of the signature code we obtain $\mathcal{L}$ itself. If $L > K$ no information about $\mathcal{L}$ can be obtained in this round. The operation of the decoder is illustrated in Figure 3.

C. User resolution for $L \le K$

If $L = |\mathcal{L}| \le K$ the receiver has exact knowledge of $\mathcal{L}$. Moreover, it has received the sum of the messages $\sum_{\ell\in\mathcal{L}} W_\ell^d$. By making use of the feedback mechanism to the users, the receiver ensures that in the next $L-1$ rounds $L-1$ of the users in $\mathcal{L}$ are individually transmitting their messages. This can be achieved by, for instance, signalling the identity of one the users in the feedback at the end of a round. In that case the feedback acts as an ACK as well as a scheduling mechanism.

D. User resolution for $L > K$

In case $L > K$ the receiver signals this fact via feedback. All users in $\mathcal{L}$ now participate in a splitting protocol with uniform splits in two groups. Each user independently of the other users draws a uniformly distributed random number from $\{1, 2\}$. All users with value 1 enter a new contention resolution phase. The users with value 2 wait until this phase ends and start another contention resolution phase afterwards. If there are more than K users in one of these groups the splitting procedure is applied recursively.

In the next section we analyze the proposed strategy, parametrized on the values of K. Note that case $K = 1$ reduces the scheme to the traditional tree splitting protocol that was discussed in Section III-C.

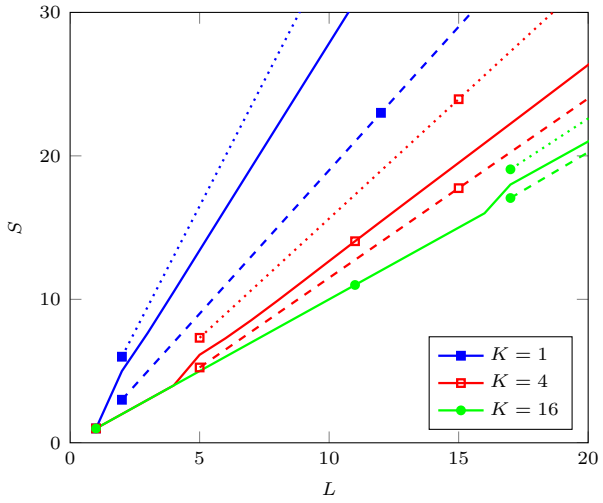

Fig. 4. $S(L)$ and its bounds for various values of K. Upper and lower bounds in dotted and dashed lines, respectively. Exact values of $S(L)$ in solid lines.

K	α^*	β^*
1	2	3.5
2	1.5	2.278
4	1.25	1.663
8	1.125	1.348
16	1.063	1.18

TABLE I
VALUES FOR α^* AND β^* THAT SERVE IN THE BOUNDS ON $S(L)$.

V. ANALYSIS

A. User resolution rate

We provide an analysis in terms of a recursive expression for $S(L)$, the expected number of slots in a contention period in terms of the number of active users L. The analysis is similar to the one by Massey [10]. We start by stating the main result of this section; the proof is given in Appendix A.

Theorem 3. *$S(L) = L$ if $1 \le L \le K$, and, for $L > K$*

$$\alpha^* L - 1 \le S(L) \le \beta^* L - 1, \tag{18}$$

where $\alpha^* = 1 + \frac{1}{K}$ and $\beta^* = 1 + \frac{1}{(K+1)(2^K-1)} + \frac{2}{K+1} + \frac{1}{K}$.

In Figure 4 we have illustrated $S(L)$ as well as the above bounds for various values of K. In Table I we provide a numerical evaluation of the bounds.

From Theorem 3 we derive results on $\overline{R}_{\text{res}}$, the expected number of users that is resolved per slot.

Theorem 4. *The expected number of users that is resolved per round is lower bounded as*

$$\overline{R}_{\text{res}} \ge \frac{\beta^* I_{1-p}(M-K, K+1) + I_p(K+1, M-K) - q_0\beta^*}{(1-q_0)\beta^*}. \tag{19}$$

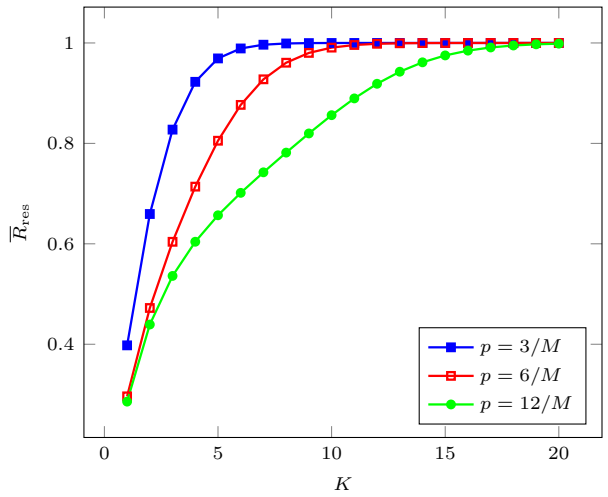

Fig. 5. Lower bounds on $\overline{R}_{\rm res}$, the expected number of users that is resolved per slot. ($M = 1031$)

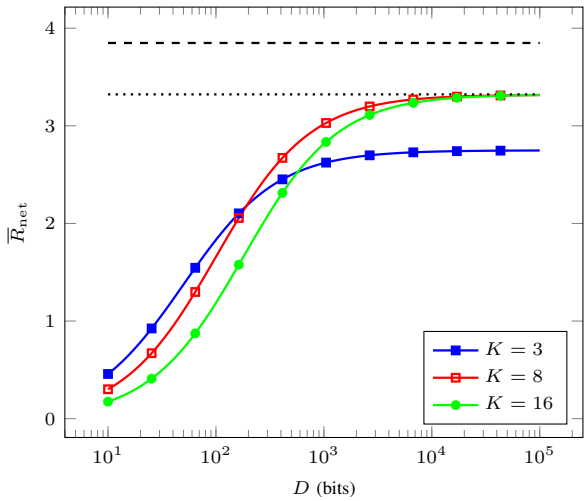

Fig. 6. Lower bounds on $\overline{R}_{\rm net}$. In dashed line the upper bound on $\overline{R}_{\rm net}$. In dotted line the value $1/2\log_2(P)$. ($M = 1031$, $p = 3/M$, $P = 10^2$)

Proof: We have

$$\overline{R}_{\rm res} = \sum_{L=1}^{M} \frac{L}{S(L)} \hat{q}(L) \tag{20}$$

$$\geq \sum_{L=1}^{K} \hat{q}(L) + \sum_{L=K+1}^{M} \frac{L}{\beta^* L - 1} \hat{q}(L) \tag{21}$$

$$\geq (1-q_0)^{-1} \left(\sum_{L=0}^{K} q(L) + \frac{1}{\beta^*} \sum_{L=K+1}^{M} q(L) - q_0 \right) \tag{22}$$

$$= \frac{\beta^* I_{1-p}(M-K, K+1) + I_p(K+1, M-K) - q_0 \beta^*}{(1-q_0)\beta^*}. \tag{23}$$

■

The result is illustrated in Figure 5 as a function of K for various values of p.

B. Rate in bits per channel use

In the previous subsection we analyzed $R_{\rm res}(L)$, the expected number of users that is resolved in a slot. In this section we consider $R_{\rm net}(L)$, which is the overall throughput in bits per channel use that is effectively transmitted.

It is readily verified that from Theorems 1, 2 and 4 it follows that

$$R_{\rm net}(L) \geq R_{\rm res}(L) R_{\rm plnc} \frac{D}{(K+2)\log_2 M + D}. \tag{24}$$

This leads to the following corollary to Theorem 4.

Corollary 1. *The expected number of bits per channel use* $\overline{R}_{\rm net}$ *is at least*

$$\overline{R}_{\rm net} \geq \sum_{L=1}^{M} \binom{M}{L} p^L (1-p)^{M-L} \frac{1}{2} \log_2 (1+LP). \tag{25}$$

Finally, we consider an information-theoretic upper bound on $R_{\rm net}$, *i.e.*, an upper bound that must be satisfied by any protocol. The bound is obtained by assuming that the receiver knows which users are active and that these users can employ a multiuser code which is optimaly decoded by the receiver. Under these assumptions the problem reduces to a standard Gaussian multi-access channel. The sum rate that can be used by L active users is

$$R_{\rm net}(L) \leq \frac{1}{2} \log_2 (1+LP). \tag{26}$$

This immediately leads to

$$\overline{R}_{\rm net} \leq \sum_{L=1}^{M} \binom{M}{L} p^L (1-p)^{M-L} \frac{1}{2} \log_2 (1+LP). \tag{27}$$

In Figure 6 we have illustrated our lower bound on $\overline{R}_{\rm net}$ as a function of D, size the data packet. In addition, Figure 6 illustrates the upper bound (27).

C. Evaluation

Figure 5 shows that as a function of K, $R_{\rm res}$ quickly approaches 1. This performance parameter is a baseline measure of the efficiency of the random access protocols from the system perspective and is usually refered to a throughput. Our results clearly demonstrate the potential of the proposed strategy. Specifically, the proposed strategy outperforms all the state-of-the-art random access schemes in terms of $R_{\rm res}$, cf. Section I. Figure 5 also shows that K should increase as the expected number of the active users pM increases, if high throughputs are to be achieved. Conversely, if K is fixed the expected throughput drops with p. This implies that one should design K to match the expected number of active users. This requirement is similar to the ALOHA-based protocols, where the optimal values of the protocol parameters, like frame lengths or user activation probability, depend on the number of active users [1], [3], [12].

Fig. 6 demonstrates what is the price to pay in information bits per channel use due to: (i) the overhead related to the use

of signatures and (ii) information waste caused by collisions, compared to the idealized solution of aforehand knowing the set of active users and using the optimal multi-user code. Obviously, this loss is pronounced for low payload lengths D and diminishes as D increases. Also, it is reflected in Figure 6 that for large D the loss that is incurred is due to the physical-layer network coding. It is currently an open problem if this loss is an inherent property of physical-layer network coding or an artifact of the computation coding construction that is developed in [6].

It is interesting to observe that the depicted results clearly suggest that due to the counter balancing of effects (i) and (ii) there is an optimal choice of K with respect to D. Finally, we note that the state-of-the-art random access protocols in general suffer from the same limitations; e.g., in SIC-based ALOHA solutions one has to invest overhead in pointers to packet replicas.

VI. Discussion

For the further work we consider extensions dealing with: errors that occur due physical-layer network coding at finite block lengths, more general user activity models (including their absence, as well), sensitivity of the performance parameters to the choice of K and variations of the scheme when the feedback channel is limited.

Acknowledgement

This work was supported in part by the Netherlands Organization for Scientific Research (NWO), grant 612.001.107, and in part by the Danish Council for Independent Research (DFF), grant DFF-4005-00281.

Appendix A
Proof of Theorem 3

Our proof is very similar in spirit to the proof that appears in [10] for the case that $K = 1$.

Let $p_L(\ell)$ denote the probability that a group of L users split into two groups where one of the groups has size ℓ. We have

$$p_L(\ell) = \binom{L}{\ell} 2^{-L}. \tag{28}$$

Note that $p_L(0) > 0$, *i.e.*, it is possible that there are groups with no users. Since our analysis is based on a recursive relation for $S(L)$ we include the case that $L = 0$. Our first result provides this recursive result.

Lemma 1.

$$S(L) = \begin{cases} 1, & \text{if } L = 0, \\ L, & \text{if } 1 \leq L \leq K, \\ \dfrac{1 + 2\sum_{\ell=0}^{L-1} p_L(\ell) S(L)}{1 - 2p_L(L)}, & \text{if } L > K. \end{cases} \tag{29}$$

Proof: Since we have a K out of M signature code, we have $S(0) = 1$, $S(1) = 1$, $S(2) = 2, \ldots, S(K) = K$. For $L > K$ we have the following recursion

$$S(L) = 1 + \sum_{\ell=0}^{L} p_L(\ell) \{S(L) + S(L-\ell)\} \tag{30}$$

$$= 1 + \sum_{\ell=0}^{L} \{p_L(\ell) S(L) + p_L(L-\ell) S(L-\ell)\} \tag{31}$$

$$= 1 + 2\sum_{\ell=0}^{L} p_L(\ell) S(L), \tag{32}$$

which can be rewritten as

$$S(L) = \frac{1 + 2\sum_{\ell=0}^{L-1} p_L(\ell) S(L)}{1 - 2p_L(L)}, \tag{33}$$

by making use of $\binom{L}{L-\ell} = \binom{L}{\ell}$ and $\sum_{\ell=1}^{L} p_L(\ell) S(L-\ell) = \sum_{\ell=1}^{L} p_L(\ell) S(\ell)$. ∎

For notational convenience let $\gamma(L)$ be defined as

$$\gamma(L) = \frac{\sum_{i=0}^{K} (S(i)+1)\, p_L(i)}{\sum_{i=0}^{K} p_L(i) i}. \tag{34}$$

The reason for introducing $\gamma(L)$ is that it can be used to express bounds on $S(L)$. The following result appears in [10].

Lemma 2 ([10]). *If α and β satisfy*

$$\alpha \leq \gamma(L) \leq \beta \tag{35}$$

for all $L \geq K + 1$, then

$$\alpha L - 1 \leq S(L) \leq \beta L - 1 \tag{36}$$

for all $L \geq K + 1$.

Next, we provide an upper and a lower bound on $\gamma(L)$.

Lemma 3.

$$1 + \frac{1}{K} \leq \gamma(L) \leq 1 + \frac{1}{(K+1)(2^K - 1)} + \frac{2}{K+1} + \frac{1}{K}. \tag{37}$$

Proof: For the lower bound we have

$$\gamma(L) = \frac{1 + \sum_{i=1}^{K} \binom{L}{i} i + \sum_{i=0}^{K} \binom{L}{i}}{\sum_{i=0}^{K} \binom{L}{i} i} \tag{38}$$

$$\geq 1 + \frac{\sum_{i=0}^{K} \binom{L}{i}}{K \sum_{i=0}^{K} \binom{L}{i}} \tag{39}$$

$$= 1 + \frac{1}{K}. \tag{40}$$

For the upper bound we start by rewriting $\gamma(L)$ as

$$\gamma(L) = 1 + \frac{1}{\sum_{i=1}^{K} \binom{L}{i} i} + \frac{\sum_{i=0}^{K} \binom{L}{i}}{\sum_{i=1}^{K} \binom{L}{i} i} \tag{41}$$

$$= 1 + \frac{1}{\sum_{i=1}^{K} \binom{L}{i} i} + \frac{2\sum_{i=0}^{K-1} \binom{L-1}{i} + \binom{L-1}{K}}{L \sum_{i=0}^{K-1} \binom{L-1}{i}} \tag{42}$$

$$= 1 + \frac{1}{\sum_{i=1}^{K} \binom{L}{i} i} + \frac{2}{L} + \rho(L), \tag{43}$$

where

$$\rho(L) = \frac{\binom{L-1}{K}}{L\sum_{i=0}^{K-1}\binom{L-1}{i}}. \tag{44}$$

The upper bound follows from the observation that the second and the third term in (43) are decreasing in L and from

$$\rho(L) = \frac{\binom{L-1}{K}}{L\sum_{i=0}^{K-1}\binom{L-1}{i}} = \frac{\frac{1}{K}\binom{L-1}{K}}{\frac{1}{K}L\sum_{i=0}^{K-1}\binom{L-1}{i}} \tag{45}$$

$$= \frac{\frac{1}{K}\binom{L-1}{K}}{\frac{L}{K}\left(\binom{L-1}{K-1} + \sum_{i=0}^{K-2}\binom{L-1}{i}\right)} \tag{46}$$

$$< \frac{\frac{1}{K}\binom{L-1}{K}}{\frac{L}{K}\binom{L-1}{K-1}} = \frac{\frac{1}{K}\binom{L-1}{K}}{\binom{L}{K}} = \frac{\frac{1}{K}\binom{L-1}{K}}{\binom{L-1}{K} + \binom{L-1}{K-1}} \tag{47}$$

$$< \frac{\frac{1}{K}\binom{L-1}{K}}{\binom{L-1}{K}} = \frac{1}{K}. \tag{48}$$

■

The proof of Theorem 3 follows directly from from Lemmas 2 and 3.

REFERENCES

[1] L. G. Roberts, "Aloha packet system with and without slots and capture," *SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.

[2] J. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," *Information Theory, IEEE Transactions on*, vol. 25, no. 5, pp. 505 – 515, sep 1979.

[3] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: How coding theory helps to build random access protocols," *submitted to IEEE Communications Magazine*, available at http://arxiv.org/abs/1205.4208.

[4] P. Popovski and H. Yomo, "Physical network coding in two-way wireless relay channels," in *Proc. of IEEE International Conference on Communications (ICC)*, 2007, pp. 707–712.

[5] ——, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 9, June 2006, pp. 3885–3890.

[6] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, 2011.

[7] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[8] J. Goseling, "A random access scheme with physical-layer network coding and user identification," in *MASSAP*, 2014.

[9] B. Nazer, "Successive compute-and-forward," in *Proc. of the International Zurich Seminar on Communications*, 2012, pp. 103–106.

[10] J. Massey, "Collision-resolution algorithms and random-access communications," in *Multi-User Communication Systems*, ser. International Centre for Mechanical Sciences, G. Longo, Ed. Springer Vienna, 1981, vol. 265, pp. 73–137.

[11] P. Popovski, F. H. Fitzek, and R. Prasad, "A class of algorithms for collision resolution with multiplicity estimation," *Algorithmica*, vol. 49, no. 4, pp. 286–317, 2007.

[12] E. Casini, R. De Gaudenzi, and O. del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.

[13] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted aloha," *Communications, IEEE Transactions on*, vol. 59, no. 2, pp. 477–487, 2011.

[14] E. Paolini, G. Liva, and M. Chiani, "High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA," in *Proc. of IEEE ICC 2011*, Kyoto, Japan, Jun. 2011.

[15] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA protocol for wireless networks," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2087–2090, 2012.

[16] G. Liva, E. Paolini, M. Lentmaier, and M. Chiani, "Spatially-Coupled Random Access on Graphs," in *Proc. of IEEE ISIT 2012*, Boston, MA, USA, Jul. 2012.

[17] C. Stefanovic and P. Popovski, "ALOHA Random Access that Operates as a Rateless Code," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4653–4662, Nov. 2013.

[18] Y. Yu and G. B. Giannakis, "Sicta: a 0.693 contention tree algorithm using successive interference cancellation," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, Mar. 2005, pp. 1908 – 1916 vol. 3.

[19] J. H. Sørensen, C. Stefanovic, and P. Popovski, "Coded Splitting Tree Protocols," in *Proc. of IEEE ISIT 2013*, Istanbul, Turkey, Jul. 2013.

[20] A. ParandehGheibi, J. K. Sundararajan, and M. Médard, "Collision helps-algebraic collision recovery for wireless erasure networks," in *Wireless Network Coding Conference (WiNC)*, 2010, pp. 1–6.

[21] ——, "Acknowledgement design for collision-recovery-enabled wireless erasure networks," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 435–442.

[22] G. Cocco, C. Ibars, D. Gunduz, and O. del Rio Herrero, "Collision resolution in slotted ALOHA with multi-user physical-layer network coding," in *Proc. of 73rd Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–4.

[23] G. Cocco, N. Alagha, C. Ibars, and S. Cioni, "A network-coded diversity protocol for collision recovery in slotted ALOHA networks," *arXiv preprint arXiv:1205.1672*, 2012.

[24] J. Goseling, M. Gastpar, and J. H. Weber, "Random access with physical-layer network coding," in *Information Theory and Applications Workshop (ITA), 2013*. IEEE, 2013, pp. 1–7.

[25] ——, "Physical-layer network coding on the random-access channel," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2339–2343.

[26] K. Censor-Hillel, B. Haeupler, N. Lynch, and M. Médard, "Bounded-contention coding for wireless networks in the high SNR regime," in *Distributed Computing*. Springer, 2012, pp. 91–105.

[27] D. Danyev, B. Laczay, and M. Ruszinko, "Multiple access adder channel," in *Multiple Access Channels*, E. Biglieri and L. Gyorfi, Eds. IOS press, 2007, pp. 26–53.

[28] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 438–460, 2011.

[29] S. C. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: Tutorial, survey, and beyond," *Physical Communication*, vol. 6, no. 0, pp. 4 – 42, 2013.