Multi-Access Spreading over Time: MAST

Sami Akın
sami.akin@ikt.uni-hannover.de
Institute of Communications Technology
Leibniz Universität Hannover
Hannover, Germany

Markus Fidler
markus.fidler@ikt.uni-hannover.de
Institute of Communications Technology
Leibniz Universität Hannover
Hannover, Germany

ABSTRACT

In this paper, we consider a multi-access communication channel with many transmitters that randomly enter a channel and send their data to a receiver. The transmitters are not synchronized and the receiver does not send any feedback to the transmitters. We propose a Medium Access Control (MAC) protocol, which we call Multi-Access Spreading over Time (MAST). In this protocol, in order to mitigate the effects of user interference, each transmitter spreads its access over a time frame that is much larger than its encoded and modulated packet size. In order to perform this operation, the transmitters choose a spreading matrix from a set, which is known by all the transmitters and the receiver. We obtain the packet decoding probability analytically under user interference conditions, and substantiate our results with simulations. We finally compare the symbol-error probability performance of our protocol with the one of the Zig-zag protocol, and show that MAST outperforms the Zig-zag protocol under the same spreading conditions in both low and high signal-to-noise ratio regimes.

CCS CONCEPTS

• **Networks** → **Network performance modeling**; • **Computing methodologies** → *Modeling methodologies.*

KEYWORDS

multi-access communication, medium access control, feedback-less communication, access spreading, zig-zag decoding, Aloha

ACM Reference Format:
Sami Akın and Markus Fidler. 2019. Multi-Access Spreading over Time: MAST. In *22nd Int'l ACM Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'19), Nov. 25–29, 2019, Miami Beach, FL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3345768.3355927

1 INTRODUCTION

Since the proposal of the ALOHA protocol more than four decades ago, data transmission in random multiple access communication settings became one of the research objectives in communication studies. The minimization of user interference on other users became one of the investigation targets. Nevertheless, with changing conditions and advances in emerging technologies, there is still a need for better medium access control (MAC) protocols and transmission methodologies. For instance, increasing network density, steadily growing demand for higher data rates, spectrum scarcity and hidden-node problems may lead to imminent packet collisions [23]. Likewise, the rise of machine-type communications brings up the necessity to support a massive number of transceivers that become active in an uncoordinated manner, and hence requires the use of coding theory and its tools for designing efficient random access protocols [15]. Moreover, the transmission of data packets without error, or with error as small as possible, is another challenge in device-to-device communications because of the constraint of no-feedback messages [21].

In this paper, we focus on a multi-access channel scenario where many transmitters randomly enter a wireless medium to send their data to a common receiver. One can consider *Internet of things* or *cyber-physical systems* where certain units (transmitters) make certain measurements within certain time periods, and send the collected data to a sink (receiver). We assume that collisions of data packets are inevitable and that the receiver cannot send any feedback to many transmitters. Since there is no information flow from the receiver to the transmitters, the transmitters are non-synchronized, which makes the transmission of data packets with error as small as possible very crucial. In order to mitigate the effects of user interference on decoding performance at the receiver, we introduce a MAC protocol, where each transmitter spreads its access over a time frame longer than its encoded and modulated packet size. Particularly, each transmitter converts its data packets into larger transmission packets by multiplying them with a spreading matrix. We call our protocol Multi-Access Spreading over Time (MAST). MAST is easy to implement at the transmitters and the receiver, and it can support the receiver to decode data in case of collisions involving many packets. Moreover, unlike the Zig-zag decoding protocol, which was introduced by Gollakota *et al.* [7] and studied by many others, the receiver is required to detect only one packet rather than many copies of one packet. Given a desired packet error probability constraint, MAST also supports more transmitters than ALOHA does.

In the sequel, we continue with the related work in Section 1.1. Then, we describe our transmission protocol in Section 2. Specifically, we start with simple examples for a smooth introduction of the protocol in Section 2.1, and then define the spreading matrix in Section 2.2. Subsequently, we achieve analytical performance measures when a single spreading matrix is employed, and substantiate them with numerical demonstrations in Sections 2.3 and 2.4, respectively. We further improve the protocol by employing a set

MSWiM '19, November 25–29, 2019, Miami Beach, FL, USA

ACM ISBN 978-1-4503-6904-6/19/11…$15.00
https://doi.org/10.1145/3345768.3355927

of spreading matrices instead of using a single spreading matrix in Section 3. Subsequently, we compare the performance of MAST with the Zig-zag protocol in Section 4. Finally, we conclude the paper in Section 5.

1.1 Related Work

There is a growing body of literature about the Zig-zag decoding protocol, which has been introduced in [7] as a solution to combat user interference in asynchronous multi-access channels. The authors showed that the Zig-zag decoding can attain the same throughput as if the colliding packets were a priori scheduled in separate time slots, while causing no change to the 802.11 protocol and introducing no extra overhead when there are no collisions. Considering the Zig-zag decoding as hard-decision belief propagation, the authors in [25] built a soft-decoding technique on top of the existing Zig-zag protocol, which maintains likelihoods and runs in a loopy manner on the factor graph created by the linear equations formed by collided packets. Moreover, taking into account the possible effects of propagation delays on the performance of the Zig-zag protocol, the authors in [27] proposed a distributed random access MAC protocol named Asynchronous Flipped Diversity ALOHA, which combines a flipped diversity transmission scheme and the Zig-zag protocol.

Regarding the performance, the authors in [19] analyzed the performance of the Zig-zag protocol for the case of two receiving nodes with two simultaneous transmitters in an additive white Gaussian noise (AWGN) channel and showed that the expected length of the error burst is less than two symbols. The authors in [20] proposed an iterative Zig-zag decoding protocol to mitigate the error aggregation, and showed that their approach can effectively defeat the error aggregation. Furthermore, the authors in [12] analyzed and simulated the Zig-zag decoding in idealized multi-access channel models, and showed that the Zig-zag decoding can significantly increase the performance levels when compared to ALOHA and Carrier-sense multiple access protocols. More recently, the authors in [24] proved that the Zig-zag protocol introduces lower encoding and decoding complexities than the other existing techniques at the expense of a slight transmission rate loss.

At the same time, there are techniques proposed as an alternative to the Zig-zag decoding. For instance, the authors in [17, 18] introduced static and dynamic assignment schemes to select transmission slots out of available ones in a frame using group divisible (combinatorial) designs. Moreover, the authors in [2] and the ones in [5] proposed the differential overlap decoding and the iterative collision recovery, respectively. Another approach to deal with the adverse results of random access is based on graph codes. The authors in [11] viewed the iterative collision resolution process as message-passing decoding on an appropriately defined Tanner graph, and showed that the well-known solution distribution is optimal and that the resulting throughput efficiency can be arbitrarily close to one. The authors in [14, 16] exploited a bipartite graph representation of the successive interference cancellation process, resembling iterative decoding of generalized low-density parity-check codes over the erasure channel, to optimize the selection probabilities of the component erasure correcting codes through a density evolution analysis. They derived the component codes that approach the capacity bounds. More recently, the authors in [22] developed a multiple access scheme for machine-to-machine communications based on the capacity-approaching analog fountain code to efficiently minimize the access delay.

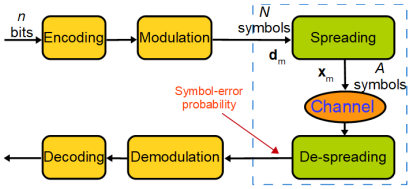


Figure 1: Transmitter block diagram for transmitter m.

2 TRANSMISSION PROTOCOL

We consider a communication scenario, in which many transmitters send data to a receiver. The transmitters are non-synchronized; each transmitter randomly enters the channel and sends its data regardless of the activities of the other transmitters. We assume that, following the reception of each packet, the receiver does not send any acknowledgment to the corresponding transmitter to inform about the status of its packet. Therefore, the transmitters send their packets only once. All the transmitters encode and modulate the same number of bits, n, into one data packet, where a packet is composed of N symbols. Moreover, all the transmitters use the same encoding and modulation techniques. After the composition of each packet, the transmitters spread their packets over time so that the receiver can take advantage of this spreading to decode the data in case of a packet collision. In other words, each transmitter converts an encoded and modulated packet of N symbols into a super packet of A symbols, where $A > N$. The relevant transmitter block diagram is displayed in Figure 1. Specifically, given that $\mathbf{d}_m = [d^*_{m1}, \cdots, d^*_{mN}]^*$ and $\mathbf{x}_m = [x^*_{m1}, \cdots, x^*_{mA}]^*$ are the data packet and the super packet sent by transmitter m, respectively, the spreading process is expressed as follows: $\mathbf{x}_m = \mathbf{G}\mathbf{d}_m$, where $\mathbf{G}$ is the $A \times N$ spreading matrix. Above, $\{*\}$ is the conjugate transpose operator. Notice that when a transmitter enters the channel without spreading its access, its packet takes a space of N symbols in the channel over time. On the other hand, after the spreading process, its packet takes a space of A symbols over time. Therefore, we define the entire process as Multi-Access Spreading over Time and use the term *MAST*. Here, our objective is to decrease the symbol-error probability before the demodulation process as much as possible.

In order to better understand the aforementioned transmission protocol, we first provide a simple example, and then provide a general model definition and substantiate our results with numerical results in the following sub-sections.

2.1 3-Transmitter and 4-Transmitter Collisions

Let us consider the collision cases given in Figure 2 and Figure 3, where transmitter m has $\mathbf{d}_m = [d^*_{m1}\ d^*_{m2}\ d^*_{m3}]^*$ as the 3×1 input vector (data packet) to the spreading process and $\mathbf{x}_m = [x^*_{m1}\ \cdots\ x^*_{m9}]^*$ as the 9×1 channel input vector (super packet) for $m \in \{1, 2, 3\}$. As one can see, the channel input vector, $\mathbf{x}_m$, is composed of symbols formed by re-ordering and adding the symbols of the input vector, $\mathbf{d}_m$, in a specified pattern. For instance,

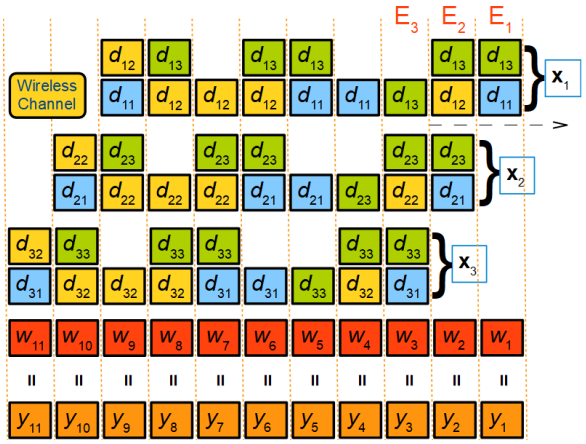


Figure 2: Three packets collide in three entries. E_i: i^{th} entry for $i \in \{1, 2, 3\}$. The dashed arrow indicates the direction.

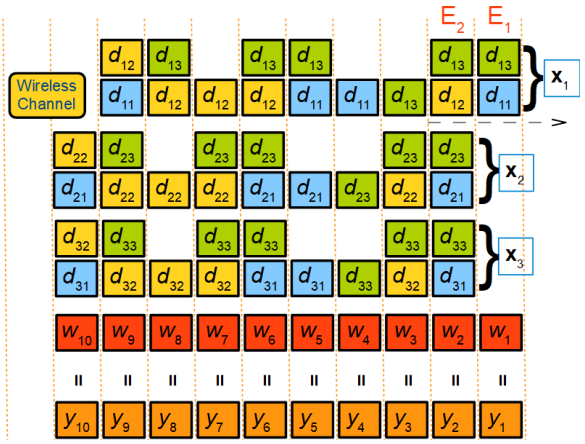


Figure 3: Three packets collide in two entries. E_i: i^{th} entry for $i \in \{1, 2\}$.

the first two symbols sent by transmitter m over the channel are $x_{m1} = d_{m1} + d_{m3}$ and $x_{m2} = d_{m2} + d_{m3}$, respectively. The transmitters enter the channel in different time slots[1] and their packets collide partially with each other in Figure 2, while transmitter 2 and transmitter 3 enter the channel in the same time slot and transmitter 1 enters in a different time slot in Figure 3. Here, if one or more transmitters enter the channel in one time slot, we call it an *entry*. Therefore, we have three entries in Figure 2 and two entries in Figure 3. Furthermore, we have y_i as the channel output in the i^{th} time slot, which contains the noise component, w_i, as well.

The channel input-output relation in Figure 2 is written as

$$\mathbf{y} = \begin{bmatrix} \mathbf{G}\mathbf{d}_1 \\ \mathbf{0}_{2\times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{1\times 1} \\ \mathbf{G}\mathbf{d}_2 \\ \mathbf{0}_{1\times 1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{2\times 1} \\ \mathbf{G}\mathbf{d}_3 \end{bmatrix} + \mathbf{w} = \mathbf{H}\mathbf{d} + \mathbf{w}, \tag{1}$$

where $\mathbf{d} = [\mathbf{d}_1^* \ \mathbf{d}_2^* \ \mathbf{d}_3^*]^*$ is the 9×1 input vector, $\mathbf{y} = [y_1^* \ \cdots \ y_{11}^*]^*$ is the 11×1 received vector, and

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{0\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{2\times 3} \\ \mathbf{G} & \mathbf{G} & \mathbf{G} \\ \mathbf{0}_{2\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{0\times 3} \end{bmatrix} \tag{2}$$

is the 11×9 channel matrix. Above, $\mathbf{0}_{a\times b}$ is the $a \times b$ zero matrix, and the spreading matrix for this specific example is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}^{\dagger}, \tag{3}$$

where $\{\dagger\}$ is the transpose operator. Furthermore, $\mathbf{w} = [w_1^* \ \cdots \ w_R^*]^*$ is the noise vector with zero-mean, and independent and identically distributed samples. While $\mathbf{G}$ is a fixed matrix throughout the entire data transmission process, the channel matrix $\mathbf{H}$ is random and changes element-wise and in size depending on the time slots in which the transmitters enter the channel. Now, re-ordering (1) as

$$(\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{y} = \mathbf{d} + (\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{w}, \tag{4}$$

we can state that if the rank of $\mathbf{H}$ is 9 and $\mathbf{H}^*\mathbf{H}$ is invertible, and if the transmission power is relatively large with respect to the noise power, the receiver can possibly decode[2,3] all the data. Principally, given $\mathbf{G}$ in (3), the receiver can decode all the data in case of a collision of 3 packets or less with any form of $\mathbf{H}$ as long as the transmitters enter the channel in different time slots.

As for the case in Figure 3, one can see that forming the channel input as $\mathbf{d} = [\mathbf{d}_1^* \ \mathbf{d}_2^* \ \mathbf{d}_3^*]^*$ will not help the receiver obtain the messages, because the channel matrix will be

$$\mathbf{H} = \begin{bmatrix} \mathbf{G} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{G} & \mathbf{G} \end{bmatrix}, \tag{5}$$

which has rank less than 9. On the other hand, we can re-define the input as $\mathbf{d} = [\mathbf{d}_1^*, \mathbf{d}_2^* + \mathbf{d}_3^*]^*$, which is a 6×1 vector, and the channel matrix as

$$\mathbf{H} = \begin{bmatrix} \mathbf{G} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & \mathbf{G} \end{bmatrix}, \tag{6}$$

where $\mathbf{H}$ is a 10×6 matrix with rank 6. Hence, after a successful decoding, the receiver obtains $\mathbf{d}_1$ and $\mathbf{d}_2 + \mathbf{d}_3$. However, the $\mathbf{d}_2$-$\mathbf{d}_3$ pair has more than one solution. Basically, in case more than one user enters the channel in the same time slot, the receiver, being able to detect these simultaneous entries, can treat them as one packet during the decoding process and discard them after the decoding process. Similarly, let us consider the case given in Figure 4, where 4 packets from four different users collide, and two of them enter the channel in the same time slot. In this case, the receiver can obtain packets $\mathbf{d}_1$ and $\mathbf{d}_4$, but not $\mathbf{d}_2$ and $\mathbf{d}_3$. Here, we define the channel input as $\mathbf{d} = [\mathbf{d}_1^*, \mathbf{d}_2^* + \mathbf{d}_3^*, \mathbf{d}_4^*]^*$, which is a 9×1 vector, and the channel matrix as given in (2). Specifically, the receiver obtains $\mathbf{d}_1$, $\mathbf{d}_2 + \mathbf{d}_3$ and $\mathbf{d}_4$, and discards $\mathbf{d}_2 + \mathbf{d}_3$ because there is no

[1] One time *slot* is equal to one signal sampling or symbol period. We assume that relative delay and phase offsets of each user in one time slot are estimated at the receiver by employing pilot symbols [19].

[2] Since we focus on interference management in the MAC layer rather than the effects of packet detection and noise on system performance in the physical layer, we assume that the receiver can detect all the collisions correctly, and we keep the noise parameter out of our analysis unless otherwise needed, and assume that the signal-to-noise ratio is high enough to consider the noise negligible. However, we employ the noise parameter in our simulations in low signal-to-noise ratio regimes. As for the detection, we refer interested readers to [1, 6, 8, 26] and references therein.

[3] One can consider (4) as least squares-based zero forcing.

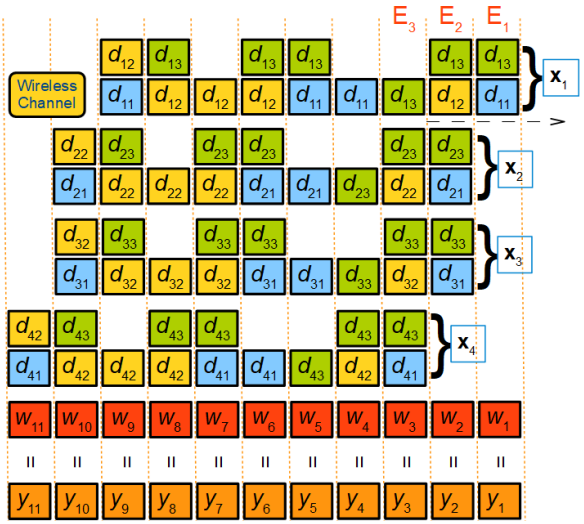


Figure 4: Four packets collide in three entries. E_i**:** i^{th} **entry for** $i \in \{1, 2, 3\}$**.**

single solution for the $\mathbf{d}_2$-$\mathbf{d}_3$ pair. This limitation will be resolved in Section 3.

2.2 Spreading Matrix

Given $\mathbf{G}$ in (3), as long as the number of entries is less than or equal to 3 in any collision case with any $\mathbf{H}$, the receiver can decode all the packets that do not enter the channel in the same time slot because the rank of any random $\mathbf{H}$ is 9, 6 and 3 in the case of 3, 2 entries and 1 entry, respectively, and the inverse of any random $\mathbf{H}^*\mathbf{H}$ exists. Herein, apart from the matrix in (3), we can find any other spreading matrix with rank 3 and size A-by-3 for $A \geq 7$, which will provide the receiver the flexibility to decode at most three packets in case of a collision of three or more packets with maximum three different entries. In more detail, the receiver in the aforementioned example can decode a packet from a user, if the packet enters the channel in a collision with three or less entries and shares its entry with no other user.

Remark 1. *When there is a collision of three entries, the size of* $\mathbf{H}$*, i.e., the number of rows, randomly changes between* $A+2$ *and* $3A-2$*. Because the number of rows in* $\mathbf{H}$ *should be minimum 9 in the given example, we have* $A \geq 7$*.*

Generalizing the aforementioned scenario, given that we want to establish a multi-access communication protocol, in which the receiver can decode up to Q transmitted packets in a collision case of maximum Q entries, what should the spreading matrix, $\mathbf{G}$, be? In the following, we provide a definition for the spreading matrix that we consider in MAST.

Definition 2.1 (Spreading Matrix). Let $\mathbf{G}$ be an $A \times N$ complex matrix, i.e., $\mathbf{G} \in \mathbb{C}^{A \times N}$, where $A \geq (N-1)Q + 1$ and $0 < Q \in \mathbb{N}^+$. Then, if any random channel (collision) matrix $\mathbf{H}$ has rank qN in the case of q entries for $q \leq Q$ and $q \in \mathbb{N}^+$, and if the inverse of any $\mathbf{H}^*\mathbf{H}$ exists, we call $\mathbf{G}$ a *spreading matrix.*

Remark 2. *The Asynchronous Flipped Diversity ALOHA protocol provided in [27] is a special case of MAST as it can be expressed with a spreading matrix;* $\mathbf{G} = \left[\mathbf{I}_N \; \widetilde{\mathbf{I}}_N\right]^\dagger$*, where* $\mathbf{I}_N$ *and* $\widetilde{\mathbf{I}}_N$ *are the* $N \times N$ *identity and anti-diagonal identity matrices, respectively.* $\mathbf{G}$ *supports decoding up-to 2 entries.*

2.3 Performance Analysis

We conduct a performance analysis from a perspective of one transmitter entering the channel along with many other transmitters, given a multi-access communication scenario, where every transmitter employs the same encoding and modulation techniques, and the same spreading matrix, $\mathbf{G}$, which is defined in Definition 2.1. We assume that transmitter m enters the channel in the slot at time t_0. Notice that transmitter m spreads its data packet over a frame of A time slots from t_0 to $t_0 + A - 1$. Moreover, we denote the number of transmitters entering the channel in one time slot by random variable K with probability mass function $\Pr(K = k)$. Particularly, the probability that k transmitters enter the channel in the same time slot is $\Pr(K = k)$. We also consider infinitely many transmitters that can possibly become active at any time.

The receiver will fail decoding the packet of transmitter m, when one or more other transmitters enter the channel in time slot t_0 alongside transmitter m, or when transmitter m enters the channel in a collision with more than Q entries. Hence, we have the following proposition regarding the decoding success probability.

Proposition 2.2. *In the aforementioned multi-access scenario, where each transmitter employs the spreading matrix given in Definition 2.1, the decoding success probability is*

$$\Pr\left\{\begin{array}{c}\textit{Successful decoding}\\ \textit{probability of a packet}\\ \textit{of transmitter } m\\ \textit{at the receiver}\end{array}\right\} = \sum_{q=1}^{Q} q\rho^{2A-1}\left(1-\rho^{A-1}\right)^{q-1}, \qquad (7)$$

where $\rho = \Pr(K = 0)$ *is the probability that there are zero transmitters entering the channel in one time slot.*

Proof: See Appendix A. □

2.4 Numerical Results

We substantiate our analytical results with numerical demonstrations and simulations. Throughout the rest of the paper, unless otherwise stated, we consider the following settings. We assume that the number of transmitters entering the channel in one time slot, K, is Poisson-distributed, i.e., $\Pr(K = k) = \exp(-\lambda)\frac{\lambda^k}{k!}$, where λ is the average number of transmitters that enter the channel in one time slot. We further consider that the symbol rate is $W = 10^6$ *symbols per second.* Hence, the average number of transmitters that enter the channel in one second is $10^6\lambda$, which can be considered as the average number of packets in the channel in one second.

In Figure 5, we plot the average number of transmitters entering the channel in one second as a function of the packet decoding failure probability, i.e., the complement of the probability in (7), for different maximum number of entries to be supported in one collision case, Q. Given any Q, we employ $\mathbf{G}$ with size $A \times N$, and set $A = (N-1)Q + 1$. Notice that $Q = 1$ refers to the ALOHA protocol. We have the packet size 50 and 100 in the upper and lower figures, respectively, i.e., $N = 50$ and $N = 100$. We compare the simulation results with the analytical results. There is a performance increase

with increasing Q; the system can support more transmitters in the channel for a desired packet decoding failure probability. Moreover, the performance gap increases with the decreasing packet decoding failure probability. The reason behind this increase is the fact that the probability of collisions with more entries than the entries that the spreading matrix can support decreases with the increasing spreading size.

We further plot the decoding success probability as a function of the maximum number of entries to be supported, Q, in Figure 6 for different packet sizes, i.e., $N = 50$, 75 and 100. We set the average number of transmitters per second to 1000 and 2000 in the upper and lower figures, respectively. The decoding success probability is high when the packet size is smaller. This is because we have a smaller super packet size when the packet size is smaller, which leads to collisions with fewer entries. Therefore, the probability of decoding failure decreases with the decreasing packet size. However, the average number of bits transmitted reliably without error, i.e., the packet size times the decoding success probability, shows a different tendency. As seen in Figure 7, the reliable transmission performance is better with higher N in certain Q ranges. We display the results in Figure 7 because the transmitters may have to transmit as much data as possible in certain circumstances.

3 TRANSMISSION PROTOCOL WITH A SET OF SPREADING MATRICES

In Section 2, we consider MAST with a fixed spreading matrix, $\mathbf{G}$, i.e., all the transmitters employ the same spreading matrix. Although there is an increase in the performance with the increasing spreading matrix size, i.e., increasing Q, as seen in Figure 5, the receiver is not able to decode the packets of the transmitters that enter the channel in the same time slot even if the number of entries in one collision case is less than the maximum number of entries to be supported. On the other hand, if two or more transmitters send their data packets after spreading their access over time by employing different spreading matrices, the receiver will possibly be able to decode the packets of the transmitters even if they enter the channel in the same time slot. For instance, let us consider the example given in Figure 3, and assume that the transmitters choose a spreading matrix from a defined set, $\mathcal{G}$:

$$\mathcal{G} = \left\{ \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}}_{\mathbf{G}_1}, \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}}_{\mathbf{G}_2}, \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}}_{\mathbf{G}_3} \right\}. \tag{8}$$

Recall that $\mathbf{G}_1$ is given also in (3). Now, let us take into account the example that each transmitter picks one of the three matrices separately, i.e., the transmitters do not choose the same matrix. Particularly, let us assume that transmitter m chooses $\mathbf{G}_m$ for $m \in \{1, 2, 3\}$. Then, we can re-write the channel matrix given in (5) as $\mathbf{H} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{G}_2 & \mathbf{G}_3 \end{bmatrix}$, which is 10×9 matrix with rank 9. Now, one can easily see that the inverse of $\mathbf{H}^*\mathbf{H}$ exists and that (1) has one solution. Even if all the transmitters in the same example enter the channel in the same time slot, the receiver will be able to obtain the transmitted symbols correctly as long as the transmitters employ different spreading matrices.

Regarding a scenario with more transmitters than the cardinality of the spreading matrix set, which is 3 in the example, one can design a system in which the transmitters randomly choose a spreading matrix from the set[4]. Now, for instance, given that two transmitters enter the channel in the same time slot, the probability that the transmitters choose the same spreading matrix is $\frac{1}{3}$. Notice that with the increasing cardinality of the given set, the probability that two or more transmitters entering the channel in the same time slot choose the same spreading matrix decreases. We note that we display only three spreading matrices as examples in $\mathcal{G}$; however, one can easily extend the set into a set with more than three matrices by finding the matrices that guarantee that the inverse of $\mathbf{H}^*\mathbf{H}$ exists in any collision case of up-to 3 entries.

As for the packet decoding probability, it is not straightforward to obtain an analytical expression given a set of spreading matrices since we have to consider all of the possible collision cases and take the expectation over all the available spreading matrices in the defined set. Therefore, we rather take advantage of simulation tools and compare the performance levels of the aforementioned protocols, i.e., the protocol with the single spreading matrix given in (3), $\mathbf{G}$, and the protocol with the set of spreading matrices given in (8), $\mathcal{G}$. Regarding high signal-to-noise ratio and considering that the binary-phase shift keying (BPSK) modulation is employed, we plot the symbol-error probability as a function of the average number of transmitters per time slot in Figure 8. The blue solid line is the performance obtained when the transmitters employ only $\mathbf{G}$, and the red dotted line indicates the performance levels obtained after employing $\mathcal{G}$. Clearly, we observe a significant increase in the decoding performance.

Moreover, we employ larger sets, $\mathcal{G}$, and calculate the symbol-error probability when the packet size is 50 and 100. We have composed three different sets for each packet size setting. Specifically, we have employed the following sets:

(1) Set 1 has $M = 20$ matrices with size 1000×50,
(2) Set 2 has $M = 30$ matrices with size 1500×50,
(3) Set 3 has $M = 40$ matrices with size 2000×50,

when $N = 50$, and

(1) Set 1 has $M = 20$ matrices with size 2000×100,
(2) Set 2 has $M = 30$ matrices with size 3000×100,
(3) Set 3 has $M = 40$ matrices with size 4000×100,

when $N = 100$. In order to obtain the aforementioned sets, we find matrices by running exhaustive search methods, with which we can obtain the inverse of $\mathbf{H}^*\mathbf{H}$ in any collision case of up-to M entries[5].

[4] In a system where there is no information flow from the receiver to the transmitters, one can embed the spreading matrix set to the transmitters and the receiver during the system initialization process.

[5] In some cases, especially when the data packets are larger, even if the inverse of $\mathbf{H}^*\mathbf{H}$ exists, $\mathbf{H}^*\mathbf{H}$ may be ill-conditioned, and hence amplifies the noise. Here, several regularization techniques, e.g., ridge regression, can be considered at the receiver.

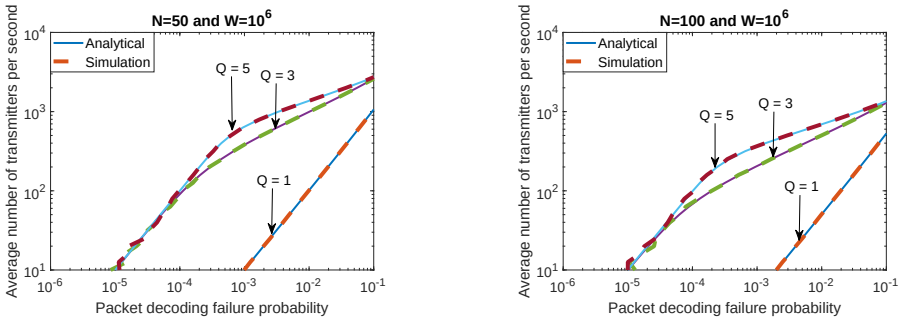


Figure 5: Average number of transmitters per second vs. packet decoding failure probability. The solid and dashed lines indicate the analytical and simulation results, respectively.

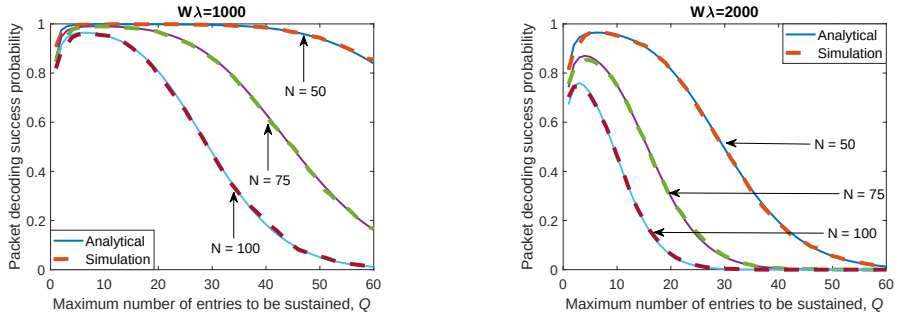


Figure 6: Decoding success probability vs. number of entries to be supported.

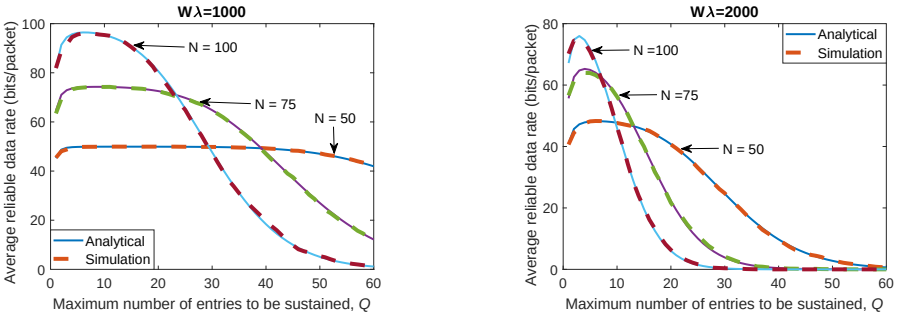


Figure 7: Average reliable data rate vs. number of entries to be supported.

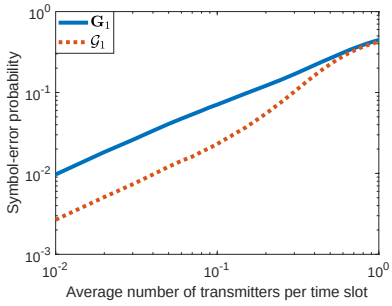


Figure 8: Symbol-error probability vs. average number of transmitters per time slot.

We set the maximum number of entries to be supported to $Q = M$. We assume that there are infinitely many potential transmitters and that the number of transmitters entering the channel in one time slot is Poisson-distributed. We set the sampling rate to $W = 10^6$ symbols per second. Hence, the average number of transmitters entering the channel in one second is $W\lambda$. We further note that because we employ the BPSK modulation, the symbol-error probability is equal to the uncoded bit error probability after the demodulation process. However, our analysis can easily be extended to scenarios with other modulation techniques.

In Figure 9, we plot the symbol-error probability as a function of the average number of transmitters entering the channel per second. We consider a high signal-to-noise ratio regime; therefore, the noise is negligible. The reason behind this assumption is to

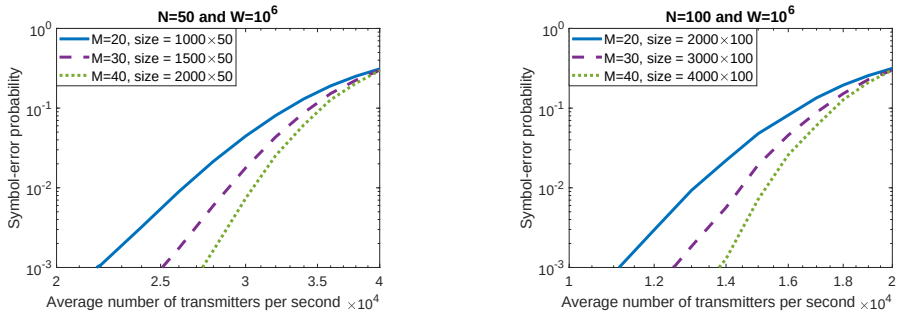


Figure 9: Symbol-error probability vs. average number of transmitters per time second. M is the cardinality of a given set.

understand the effects of user interference on performance levels. We can observe that the decoding performance gets better with the increasing cardinality of the sets, and the increasing access spreading (i.e., the increasing spreading matrix size). We also emphasize that, as seen in Figure 9, the symbol-error probability decreases to 10^{-3} when the average number of transmitters is around 2.5×10^4 and 1.25×10^4 given that the packet size is 50 and 100 symbols, respectively, in a channel sampled with rate 10^6 symbols per second. This means that one can easily reach a complete channel utilization with the choice of appropriate error-control coding techniques.

Moving on now to understand the decoding performance in low signal-to-noise ratio regimes, we set the average number of transmitters per second to 3×10^4 and 1.5×10^4 when the packet size is 50 and 100, respectively. Then, we plot the symbol-error probability as a function of the signal-to-noise ratio in Figure 10. We define the signal-to-noise ratio at transmitter m as

$$\text{SNR}_m^{\text{MAST}} = \frac{\mathbb{E}\{||\mathbf{x}_m||^2\}}{N\sigma_w^2} = \frac{\mathbb{E}\{||\mathbf{G}\mathbf{d}_m||^2\}}{N\sigma_w^2}, \tag{9}$$

where σ_w^2 is the noise variance and $\mathbb{E}\{\cdot\}$ is the expectation operator. Notice in Figure 9 that the symbol-error probability is 0.0445, 0.0178, and 0.0074, when the cardinality of the spreading matrix set is 20, 30, and 40, respectively, given that the packet size is 50 and the average number of transmitters is 3×10^4. Likewise, the symbol-error probability is 0.0479, 0.0194, and 0.0072, when the cardinality of the spreading matrix set is 20, 30, and 40, respectively, given that the packet size is 100 and the average number of transmitters is 1.5×10^4. Specifically as seen in Figure 10, we can observe that with the increasing signal-to-noise ratio, the symbol-error probability decreases to the values that are obtained in Figure 9. Finally, the decoding performance gets better in low signal-to-noise ratio regimes with the increasing cardinality and spreading size.

4 COMPARISON WITH THE ZIG-ZAG PROTOCOL

As far as asynchronous multi-access protocols are concerned, the Zig-zag protocol is one leading technique in the literature that can mitigate the effects of user interference. However, there is an implementation cost following the deployment of the Zig-zag protocol. Specifically, receivers have to detect all the copies of the transmitted packets, or they have to detect as many copies as possible to alleviate user interference. In addition, each copy of a packet necessitates the use of a header and preambles in order to be detected over a channel. On the contrary, MAST spreads a data packet over a larger time frame and sends one copy only; hence, receivers have to detect only once. Since there is only one big packet, the cost of headers and preambles is decreased as well. Hence, in order to compare MAST with the Zig-zag protocol, we translate the Zig-zag protocol into our framework.

Remark 3. *In the Zig-zag protocol, we can express the channel input with a spreading matrix. Particularly, we can formulate the spreading matrix as follows:* $\mathbf{G} = \left[\mathbf{I}_N\ \mathbf{0}_{N\times\alpha_1}\ \mathbf{I}_N \dots \mathbf{0}_{N\times\alpha_{Z-1}}\ \mathbf{I}_N\right]^\dagger$, *where Z is the number of packet repetitions, and α_z for $z \in \{1, \cdots, Z-1\}$ is a random variable, which indicates the time gap between the z^{th} and $(z+1)^{th}$ copies of a packet. The matrix, $\mathbf{G}$, is composed of Z identity matrices and $Z-1$ zero matrices. In addition, the size of $\mathbf{G}$ varies randomly. For more information on the Zig-zag protocol and other repetition-based packet transmissions, we refer interested readers to [3, 4, 7, 9, 10, 13]. We also note that one can easily compose a set in MAST, which has spreading matrices with varying column size although we have sets with fixed-size matrices in our example in Figure 9 and Figure 10.*

We compare the performance levels of MAST and the Zig-zag protocol under the conditions of spreading range and packet size. With spreading range, we refer to the period that the copies of a packet spans over time in the Zig-zag protocol and the access spreading in MAST that we describe in Section 2. As for the Zig-zag protocol, we have two implementations, namely Zig-zag 10 and Zig-zag 20, which refer to the Zig-zag protocol with 10 and 20 packet repetitions, respectively. Each transmitter waits for a random amount of time between the copies of a packet, where the waiting duration is uniformly distributed over a range between one time slot and three times the packet size, i.e., between one time slot and $3 \times N$ time slots. Notice that when a packet of 50 symbols is transmitted in Zig-zag 10 and Zig-zag 20, its copies span a range up-to 1850 and 3850 time slots, respectively, and that when a packet of 100 symbols is transmitted in Zig-zag 10 and Zig-zag 20, its copies span a range up-to 3700 and 7700 time slots, respectively. In the aforementioned MAST protocols, a message spans 1000, 1500 and 2000 time slots when the packet size is 50, and 2000, 3000 and 4000 time slots when the packet size is 100.

In both MAST and the Zig-zag protocol, since we are interested in understanding the system performance associated with asynchronous user interference management, we do not focus on the

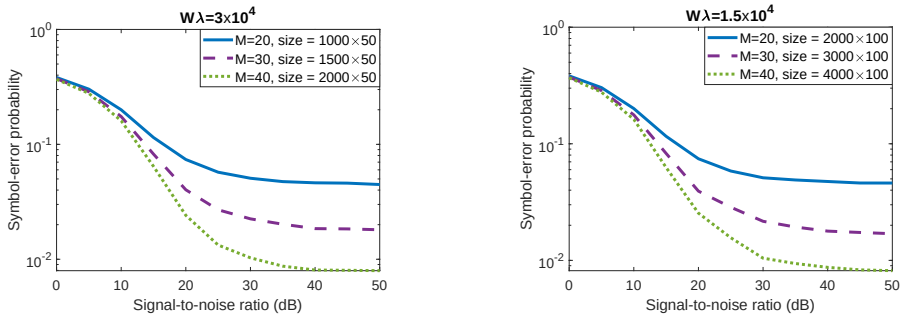


Figure 10: Symbol-error probability vs. signal-to-noise ratio.

detection of transmitted packets by the receiver; therefore, we assume that the start and end of data packets in terms of time slots are detected correctly and available at the receiver. We refer interested readers to [7, Sec. 5] for more information on packet collisions and detections. Moreover, we define the signal-to-noise ratio in the Zig-zag protocol as

$$\mathrm{SNR}_m^{\text{Zig-zag}} = \frac{10\mathbb{E}\{||\mathbf{d}_m||^2\}}{N\sigma_w^2} \text{ and } \mathrm{SNR}_m^{\text{Zig-zag}} = \frac{20\mathbb{E}\{||\mathbf{d}_m||^2\}}{N\sigma_w^2},$$

when the packet repetition is 10 and 20, respectively. Particularly, when we have $\mathrm{SNR}_m^{\text{Zig-zag}} = \mathrm{SNR}_m^{\text{MAST}}$, the amount of energy spent for the transmission of one data packet is same in all the protocols. This is an important constraint in low signal-to-noise ratio regimes in order to perform a fair comparison.

As seen in Figure 11 and Figure 12, the red solid and light blue dashed lines with plus ('+') marker indicate the performance levels when Zig-zag 10 and Zig-zag 20 are employed, respectively. As seen in the figures, the performance of MAST with spreading size 1500 and 2000 outperforms Zig-zag 10 with spreading size 1850 when $N = 50$, and similarly, the performance of MAST with spreading size 3000 and 4000 outperforms Zig-zag 20 with spreading size 3700 when $N = 100$. The performance of Zig-zag 20 is better, but it takes an effort of 20 copies of a packet with a span over a range up-to 3850 and 7700 time slots given $N = 50$ and $N = 100$, respectively. Particularly, the Zig-zag protocol needs to span a packet on a time frame more than MAST needs in order to catch the performance of MAST. Considering that we do not have the preambles, which increase the size of transmitted packets, and the detection errors in these simulations, we expect that the performance of MAST will be better than the performance of the Zig-zag protocol in practice. Note that the transmitters send preambles only once in MAST, while they have to send a group of preambles for each copy in the Zig-zag protocol. Moreover, the receiver has to detect one packet in MAST, whereas it has to detect more than one packets in the Zig-zag protocol.

5 CONCLUSION

We have proposed a MAC protocol in multi-access communication scenarios. We have called it MAST. Our proposed technique is simple to implement at the transmitter and receiver. We have introduced MAST with simple examples, and then formulated the performance measures. We have performed simulations to substantiate our results. We have showed that MAST can serve more transmitters than ALOHA does for a desired minimum packet decoding failure probability. Our results also indicate that one can easily reach a complete channel utilization using MAST because the symbol-error probability can be decreased to 10^{-3}. Finally, we have compared the performance of MAST with the Zig-zag protocol, and showed that our protocol outperforms the Zig-zag protocol when we consider the spreading of a message over time in both protocols.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) – FeelMaTyc (FI 1236/6-1).

REFERENCES

[1] Yihenew Dagne Beyene, Riku Jäntti, and Kalle Ruttik. 2017. Random access scheme for sporadic users in 5G. *IEEE Transactions on Wireless Communications* 16, 3 (2017), 1823–1833.

[2] Jingye Cao, Feng Yang, Lianghui Ding, Liang Qian, and Cheng Zhi. 2013. Differential Overlap Decoding: Combating hidden terminals in OFDM systems. In *Wireless Communications and Networking Conference (WCNC)*. IEEE, Shanghai, China, 3732–3736.

[3] Enrico Casini, Riccardo De Gaudenzi, and Oscar Del Rio Herrero. 2007. Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access schemefor satellite access packet networks. *IEEE Transactions on Wireless Communications* 6, 4 (2007), 1408 – 1419.

[4] Federico Clazzer and Mario Marchese. 2014. Layer 3 throughput analysis for advanced ALOHA protocols. In *IEEE International Communications Conference (ICC)*. IEEE, Sydney, NSW, Australia, 533–538.

[5] Yan Du, Liang Qian, Lianghui Ding, Cheng Zhi, and Feng Yang. 2013. Iterative Collision Recovery for OFDM based WLAN. In *International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, Hangzhou, China, 1–6.

[6] Alyson K Fletcher, Sundeep Rangan, and Vivek K Goyal. 2009. A sparsity detection framework for on-off random access channels. In *IEEE International Symposium on Information Theory (ISIT)*. IEEE, Seoul, Korea, 169–173.

[7] Shyamnath Gollakota and Dina Katabi. 2008. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*. ACM, Seattle, WA, USA, 159–170.

[8] Gabor Hannak, Martin Mayer, Gerald Matz, and Norbert Goertz. 2016. Bayesian QAM demodulation and activity detection for multiuser communication systems. In *IEEE International Communications Conference (ICC)*. IEEE, Kuala Lumpur, Malaysia, 596–601.

[9] Christian Kissling. 2011. Performance enhancements for asynchronous random access protocols over satellite. In *IEEE International Communications Conference (ICC)*. IEEE, Kyoto, Japan, 1–6.

[10] Gianluigi Liva. 2011. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. *IEEE Transactions on Communications* 59, 2 (2011), 477–487.

[11] Krishna R Narayanan and Henry D Pfister. 2012. Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy. In *International*

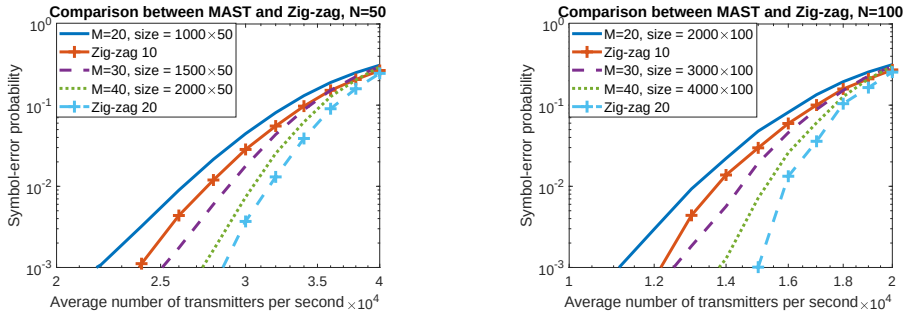


Figure 11: Symbol-error probability vs. average number of transmitters per second.

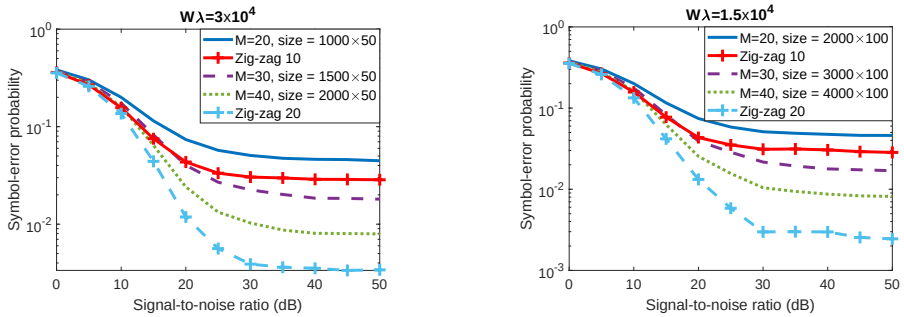


Figure 12: Symbol-error probability vs. signal-to-noise ratio.

Symposium on Turbo Codes and Iterative Information Processing (ISTC). IEEE, Gothenburg, Sweden, 136–139.

[12] Jeongyeup Paek and Michael J Neely. 2011. Mathematical analysis of throughput bounds in random access with ZIGZAG decoding. *Mobile Networks and Applications* 16, 2 (2011), 255–266.

[13] Enrico Paolini, Gianluigi Liva, and Marco Chiani. 2011. High throughput random access via codes on graphs: Coded slotted ALOHA. In *IEEE International Communications Conference (ICC)*. IEEE, Kyoto, Japan, 1–6.

[14] Enrico Paolini, Gianluigi Liva, and Marco Chiani. 2015. Coded slotted ALOHA: A graph-based method for uncoordinated multiple access. *IEEE Transactions on Information Theory* 61, 12 (2015), 6815–6832.

[15] Enrico Paolini, Cedomir Stefanovic, Gianluigi Liva, and Petar Popovski. 2015. Coded random access: applying codes on graphs to design random access protocols. *IEEE Communications Magazine* 53, 6 (2015), 144–150.

[16] Enrico Paolini, Cedomir Stefanovic, Gianluigi Liva, and Petar Popovski. 2015. Coded random access: applying codes on graphs to design random access protocols. *IEEE Communications Magazine* 53, 6 (2015), 144–150.

[17] Gino T Peeters, R Bocklandt, and Benny Van Houdt. 2009. Multiple access algorithms without feedback using combinatorial designs. *IEEE Transactions on Communications* 57, 9 (2009), 2724–2733.

[18] Gino T Peeters and Benny Van Houdt. 2010. Design and analysis of multi-carrier multiple access systems without feedback. In *2010 22nd International Teletraffic Congress (ITC 22)*. IEEE, Amsterdam, Netherlands, 1–8.

[19] David Qiu, Frederick J Block, and Everest W Huang. 2010. Analysis and extension of zigzag multiuser detection. In *IEEE Military Communications Conference (MILCOM)*. IEEE, Shanghai, China, 1771–1776.

[20] Md Shahriar Rahman, Yonghui Li, and Branka Vucetic. 2010. An iterative ZigZag decoding for combating collisions in wireless networks. *IEEE Communications Letters* 14, 3 (2010), 242–244.

[21] Mei-Ju Shih, Guan-Yu Lin, and Hung-Yu Wei. 2015. A distributed multi-channel feedbackless MAC protocol for D2D broadcast communications. *IEEE Wireless Communications Letters* 4, 1 (2015), 102–105.

[22] Mahyar Shirvanimoghaddam, Yonghui Li, Mischa Dohler, Branka Vucetic, and Shulan Feng. 2015. Probabilistic rateless multiple access for machine-to-machine communication. *IEEE Transactions on Wireless Communications* 14, 12 (2015), 6815–6826.

[23] Hossein Shokri-Ghadikolaei, Carlo Fischione, Petar Popovski, and Michele Zorzi. 2016. Design aspects of short-range millimeter-wave networks: A MAC layer perspective. *IEEE Network* 30, 3 (2016), 88–96.

[24] Chi Wan Sung and Xueqing Gong. 2014. Combination network coding: Alphabet size and zigzag decoding. In *International Symposium on Information Theory and its Applications (ISITA)*. IEEE, Melbourne, VIC, Australia, 699–703.

[25] Arash Saber Tehrani, Alexandros G Dimakis, and Michael J Neely. 2011. Sigsag: Iterative detection through soft message-passing. *IEEE Journal of Selected Topics in Signal Processing* 5, 8 (2011), 1512–1523.

[26] Qiwei Wang, Guangliang Ren, and Jueying Wu. 2015. A multiuser detection algorithm for random access procedure with the presence of carrier frequency offsets in LTE systems. *IEEE Transactions on Communications* 63, 9 (2015), 3299–3312.

[27] Lei Zheng and Lin Cai. 2015. AFDA: Asynchronous flipped diversity ALOHA for emerging wireless networks with long and heterogeneous delay. *IEEE Transactions on Emerging Topics in Computing* 3, 1 (2015), 64–73.

A PROOF OF PROPOSITION 2.2

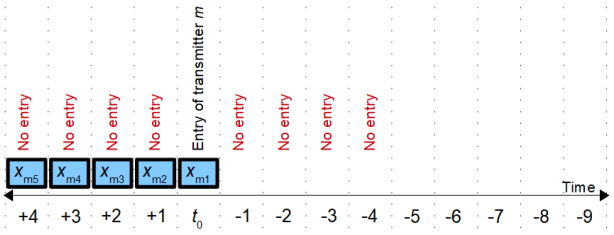
We can easily calculate the decoding failure of the packet of transmitter m due to the entries of others at t_0 as

$$\Pr\left\{\begin{array}{c}\text{One or more transmitters other than}\\ \text{transmitter } m \text{ enter the channel at } t_0\\ \text{given that transmitter } m \text{ is in the channel}\end{array}\right\} = 1 - \underbrace{\Pr(K=0)}_{\rho}. \quad (10)$$

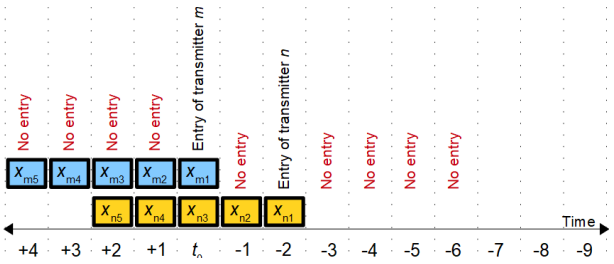
Notice that when there is another transmitter entering the channel at the time transmitter m enters, it is not important for transmitter m if there are other transmitters entering the channel within the same collision or not. As for the probability that there are q number of entries in the same collision around t_0 but there are zero transmitters entering the channel at time t_0, we have

$$\Pr\left\{\begin{array}{c}q \text{ number of entries in one}\\ \text{collision case but zero other}\\ \text{transmitters at } t_0\\ \text{given that transmitter } m \text{ is}\\ \text{in the channel at } t_0\end{array}\right\} = q\rho^{2A-1}\left(1-\rho^{A-1}\right)^{q-1}. \quad (11)$$

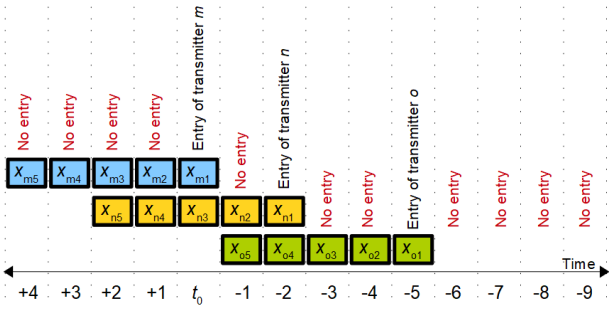
Note that one of the entries in (11) is the entry of transmitter m at



(a) Transmitter m only



(b) Transmitter m plus one entry (one or more transmitters entering at the same time)



(c) Transmitter m plus two entries (one or more transmitters in each entry)

Figure 13: No collision case, i.e., transmitter m is not interfered, and two collision cases with two entries and three entries, respectively.

t_0, and transmitter m is the only transmitter entering the channel at t_0. In order to obtain (11), we first remark that given a collision case, we know that there are no entries in the first $A-1$ time slots after the last entry in the collision, and in the last $A-1$ time slots before the first entry in the collision. As seen in Figure 13, where we have $A=5$ as an example, there are no entries in the first 4 time slots after the last entry and in the last 4 time slots before the first entry in all three cases. Now, let us follow an induction approach, calculate the probability that there is only one entry, which is the entry of transmitter m, given that transmitter m enters the channel at t_0, as seen in Figure 13(a). It is easily seen that the probability that transmitter m is the only transmitter in the channel between time instants t_0-4 and t_0+4 is equal to the probability that there are no transmitters entering the channel between time instants t_0-4 and t_0+4, which is expressed as $\rho^9=\rho^{2A-1}$. As for the case in Figure 13(b) with two entries, we initially note that transmitter(s) n can enter the channel[6] either before or after transmitter m. However, we can consider the case transmitter n enters the channel before transmitter m, and multiply the probability by two. We also note that we do not consider the case, when both transmitter m and transmitter n enter the channel at t_0 because this case is already included in (10). Now, noting that transmitter n may enter the channel in time slots between t_0-1 and t_0-4 (i.e., t_0-1 and t_0-A+1), we can easily calculate the probability that there are two number of entries in one collision case but zero other transmitters at t_0 given that transmitter m enters the channel at t_0 as

$$\begin{aligned}&2\rho^9\left[(1-\rho)+(1-\rho)\rho+(1-\rho)\rho^2+(1-\rho)\rho^3\right]\\&=2\rho^9(1-\rho)\left[1+\rho+\rho^2+\rho^3\right]\\&=2\rho^{2A-1}(1-\rho)\left[1+\rho+\rho^2+\rho^{A-2}\right]=2\rho^{2A-1}(1-\rho^{A-1}),\end{aligned}$$

where 2 in front of the equation comes from the fact that transmitter n may enter the channel before or after transmitter m. Furthermore, $(1-\rho)$ refers to the probability that one or more transmitters enter the channel at the same time, and ρ^9 comes from the fact that there are no other transmitters entering the channel at t_0 and that there are no transmitters in the first $A-1$ time slots after t_0 and the last $A-1$ time slots before the collision starts. Now, considering Figure 13(c), we can again see that transmitter m may enter the channel first, second or last. Therefore, because all cases are equally likely and have the same probability, we will only consider the case transmitter m enters the channel last. Initially, let us consider that the second transmitter or group of transmitters (i.e., transmitter n) enter the channel at time instant t_0-2 is given, and then write the probability as

$$\begin{aligned}&\rho^9\rho(1-\rho)\left[(1-\rho)+(1-\rho)\rho+(1-\rho)\rho^2+(1-\rho)\rho^3\right]\\&=\rho^9\rho(1-\rho)(1-\rho^4),\end{aligned}$$

where ρ^9 comes again due to that there are no entries at t_0 other than transmitter m and that there are no transmitters in the first $A-1$ time slots after t_0 and the last $A-1$ time slots before the collision starts. Furthermore, $\rho(1-\rho)$ is due to the position of transmitter n, and the rest is the sum of the probabilities of all the possible positions of transmitter o. Furthermore, considering all the possible positions of transmitter n and generalizing the result, we obtain the probability that there are three number of entries in one collision case but zero other transmitters at t_0 given that transmitter m enters the channel at t_0 as

$$\begin{aligned}&3\rho^9\left[(1-\rho)+(1-\rho)\rho+(1-\rho)\rho^2+(1-\rho)\rho^3\right](1-\rho^4)\\&=3\rho^9(1-\rho)\left[1+\rho+\rho^2+\rho^3\right](1-\rho^4)\\&=3\rho^9(1-\rho^4)^2=3\rho^{2A-1}(1-\rho^{A-1})^2,\end{aligned}$$

where 3 in front of the equation comes from the fact that transmitter m may enter the channel first, second or last. Now, generalizing the aforementioned probabilities for q number of entries in one collision case, we obtain the result in (11). Hence, noting the cases the receiver succeeds the decoding of a packet, we reach the result in Proposition 2.2.

[6]With transmitter(s) n, we refer to one entry, i.e., one or more transmitters entering the channel in the same time slot. However, we show only one transmitter in the examples in Figure 13.