CHALMERS

Chalmers Publication Library

Probabilistic Handshake in All-to-all Broadcast Coded Slotted ALOHA

(article starts on next page)

Probabilistic Handshake in All-to-all Broadcast Coded Slotted ALOHA

Mikhail Ivanov, Petar Popovski[§], Fredrik Brännström, Alexandre Graell i Amat, and Čedomir Stefanović[§]

Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

[§]Department of Electronic Systems, Aalborg University, Aalborg, Denmark

{*mikhail.ivanov, fredrik.brannstrom, alexandre.graell*}@*chalmers.se*, {*petarp, cs*}@*es.aau.dk*

Abstract—We propose a probabilistic handshake mechanism for all-to-all broadcast coded slotted ALOHA. We consider a fully connected network where each user acts as both transmitter and receiver in a half-duplex mode. Users attempt to exchange messages with each other and to establish one-to-one handshakes, in the sense that each user decides whether its packet was successfully received by the other users: After performing decoding, each user estimates in which slots the resolved users transmitted their packets and, based on that, decides if these users successfully received its packet. The simulation results show that the proposed handshake algorithm allows the users to reliably perform the handshake. The paper also provides some analytical bounds on the performance of the proposed algorithm which are in good agreement with the simulation results.

I. INTRODUCTION

Vehicular communications (VCs) is presently one of the most challenging problems of communication engineering. Its deployment will enable numerous applications, such as intelligent transportation systems, autonomous driving, and, most importantly, traffic safety. VCs entails a number of challenges, such as all-to-all communication, high mobility networks with rapidly changing topologies and a large number of users, and poor channel quality. These challenges require new ideas and designs at the physical and the medium access control (MAC) layers. The main requirements for VCs are high reliability and low latency. Furthermore, the aforementioned challenges prohibit the classical use of acknowledgements in the form of additional signaling.

A novel MAC protocol called all-to-all broadcast coded slotted ALOHA (B-CSA) was proposed by the authors in [1], which was shown to be able to satisfy the reliability and latency requirements in rough conditions of vehicular networks under a set of idealized assumptions, such as perfect interference cancellation. Originally proposed for a unicast scenario, coded slotted ALOHA (CSA) can provide large throughputs close to those of coordinated schemes [2], [3]. Different versions of CSA have been proposed (see [4] for the most recent review). All of them share a slotted structure borrowed from the original slotted ALOHA [5] and the use of

successive interference cancellation. The contending users introduce redundancy by encoding their messages into multiple packets, which are transmitted in randomly chosen slots. In the unicast scenario, the base station (BS) buffers the received signal, decodes the packets from the slots with no collision and attempts to reconstruct the packets in collision exploiting the introduced redundancy. A packet that is reconstructed is subtracted from the buffered signal and the BS proceeds with another decoding round.

In contrast to classical CSA, where a BS is the intended recipient of the messages, in B-CSA each user acts as both receiver and transmitter. Every user is equipped with a half-duplex transceiver, so that a user cannot receive packets in the slots it uses for transmission. This can be modeled as a packet erasure channel [6] and it affects the design and the performance analysis as compared to classical CSA.

Whereas B-CSA can provide high reliability, the rear communication failure events may be extremely costly in safety applications. Since providing error-free communication under the described conditions of VCs is not possible, one may attempt to detect communication failure events to use this information in the application level. For instance, consider the scenario where two users A and B are heading towards each other. If A receives a message from B and obtains the information that B failed to receive its message, A can take extra precautions to avoid collision with B. In this paper, we propose an algorithm to obtain this information based on the by-product of decoding, i.e., no extra signaling is used by the users. In particular, A uses the knowledge of B's transmissions to detect that B failed to resolve A.

The problem studied in this paper resembles the one of handshake used for establishing connections in connection-oriented protocols. In transmission control protocol (TCP), a three-way handshake is used by a pair of users to exchange their messages and to acknowledge that the messages were received [7]. If the messages in the TCP level are exchanged, then the handshake is always performed successfully. In the proposed algorithm, however, the decision about successful handshake can be in error, which indicates its probabilistic nature.
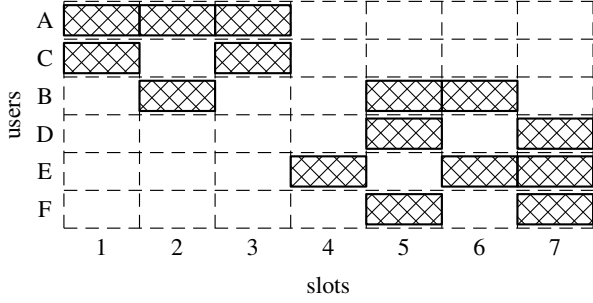
Fig. 1: Users' transmissions in a B-CSA system within one frame. Shaded rectangles represent transmitted packets.

II. SYSTEM MODEL

In this section, we first describe how encoding and decoding are performed in B-CSA. Based on that, we describe the proposed handshake algorithm.

A. Coded Slotted ALOHA

We consider a fully connected network with m users that want to communicate between each other over a shared medium. We focus on the exchange of cooperative awareness messages (CAMs) [8] used for safety applications, which are transmitted periodically by each user. The transmission period is called *frame* and it is divided into n slots of equal duration. Users are assumed to be frame-synchronized by means of Global Positioning System (GPS). Each user maps its message to a physical layer packet and repeats it l times (l is a random number chosen based on a predefined distribution) in randomly chosen slots, as shown in Fig. 1 for a system with 6 users and 7 slots. Such a user is called a degree-l user. Every packet contains pointers to its copies, so that, once a packet is successfully decoded, full information about the location of the copies is available.

Under the assumptions described in the following, the system can be analyzed using the theory of codes on graphs on the binary erasure channel (BEC). Each user corresponds to a variable node (VN) in the bipartite graph and represents a repetition code, whereas slots correspond to check nodes (CNs) and can be seen as single parity-check codes. In the following, users and VNs are used interchangeably. An edge connects the jth VN to the ith CN if the jth user transmits in the ith slot. For the example in Fig. 1, the corresponding bipartite graph is shown in Fig. 2(a). A bipartite graph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{C}, \mathcal{E}\}$, where $\mathcal{V}$, $\mathcal{C}$, and $\mathcal{E}$ represent the sets of VNs, CNs, and edges, respectively.

The performance of the system depends on the distribution that users use to choose the degree l or, using graph terminology, on the VN degree distribution

$$\lambda(x) = \sum_{l=0}^{q} \lambda_l x^l, \qquad (1)$$

where x is a dummy variable, λ_l is the probability of choosing degree l, and q is the maximum degree, which is often bounded due to implementation constraints.

(a) Original "unicast" graph $\mathcal{G}$.

(b) Induced graph $\mathcal{G}_A$ for user A. $\mathcal{G}_A(B) = 1$.

(c) Intermediate graph $\mathcal{A}''$ reconstructed by user A and slots with residual interference (marked with gray).

(d) Intermediate graph $\mathcal{A}'$ reconstructed by user A excluding slots with residual interference.

(e) Graph $\mathcal{A}$ reconstructed by user A (including user A's slots).

(f) Induced graph $\mathcal{A}_B$ for user B based on $\mathcal{A}$. $\mathcal{A}_B(A) = 1$.

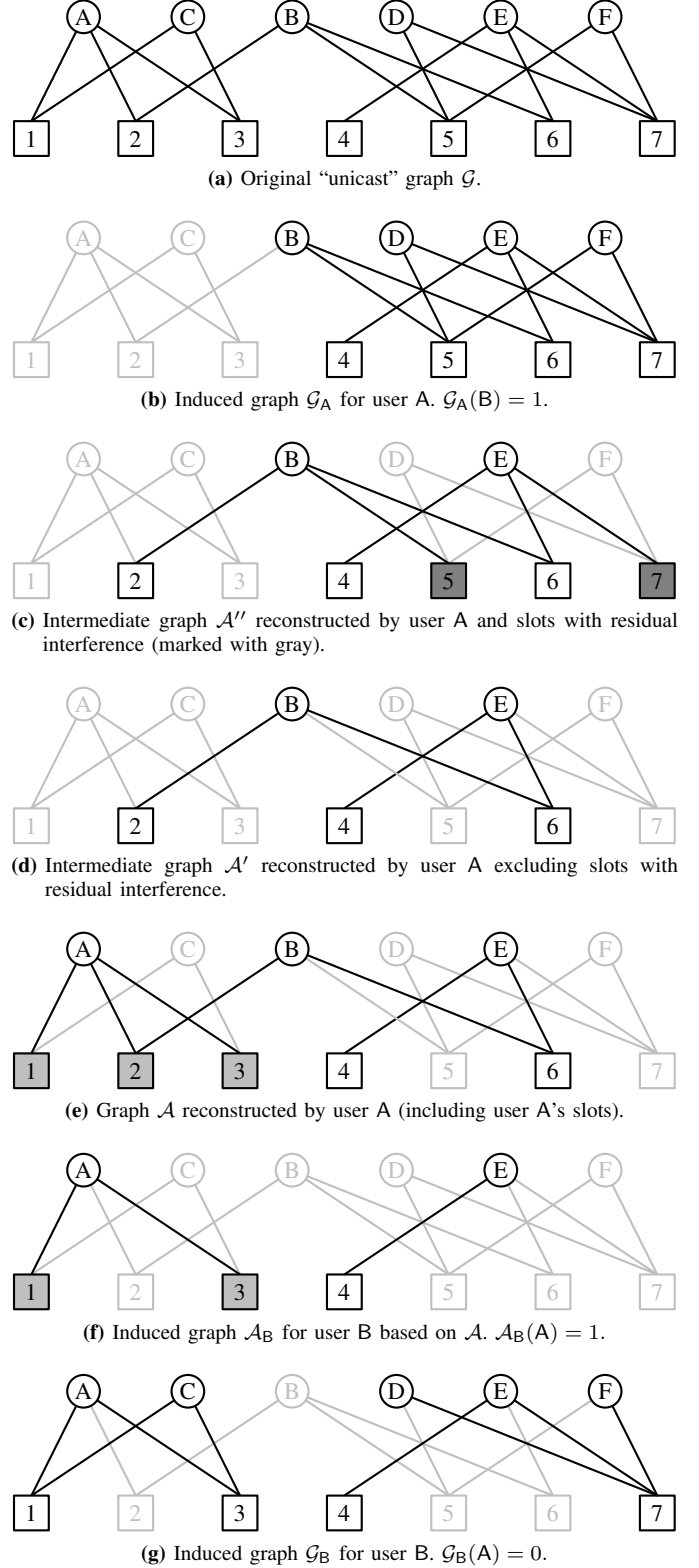**(g)** Induced graph $\mathcal{G}_B$ for user B. $\mathcal{G}_B(A) = 0$.

Fig. 2: The graph evolution over the course of the handshake algorithm. Circles represent users (VNs) and squares represent slots (CNs). The example corresponds to the outcome $\boldsymbol{g} = [1, 1, 0]$ (see Section II-C).

Users buffer the received signal whenever they are not transmitting. The difference between classical CSA and B-CSA is illustrated in the example of Fig. 2. The entire graph $\mathcal{G}$ is available to the BS in CSA (Fig. 2(a)). For a generic user A in B-CSA, the part of the graph that corresponds to user A's transmissions is not available to it due to the half-duplex mode. Thus, this part of the graph shown with gray in Fig. 2(b). The available part of the graph is called the *induced* graph and is denoted by $\mathcal{G}_A$. The received signal buffered by user A in slot i is

$$y_i = \sum_{j \in \mathcal{U}_i} h_{i,j} a_j, \tag{2}$$

where $\mathcal{U}_i \subset \mathcal{U}$ is the set of users that transmit in the ith slot, $\mathcal{U}$ is the set of all users, a_j is a packet of the jth user in $\mathcal{U}_i$, and $h_{i,j} > 0$ is the channel coefficient. A slot is called a *singleton* slot if it contains only one packet. If it contains more packets, we say that a collision occurs.

When decoding, user A first decodes the packets in singleton slots and obtains the location of their copies. Using data-aided methods, the channel coefficients corresponding to the copies are then estimated. After subtracting the interference caused by the identified copies, decoding proceeds until no further singleton slots are found.

The performance parameters of B-CSA are defined as follows. The channel load is defined as $g = m/n$. The average number of users that are not successfully resolved by user A, termed *unresolved users*, is denoted by $\bar{w}$. The reliability is measured by means of the average packet loss rate (PLR), $\bar{p} = \bar{w}/(m-1)$, which is the probability of a user to be unresolved by user A.

B. Probabilistic Handshake

One of the main differences of CSA compared to actual codes on graphs is that, in CSA, the graph is not known to the decoder a priori and the decoder reconstructs it while decoding. We use this reconstructed graph to perform the handshake. Without loss of generality, we concentrate on the handshake between users A and B.

We first introduce the necessary notation to describe the proposed handshake algorithm. Given a particular realization of the graph $\mathcal{G}$ with m VNs and n CNs generated randomly using the distribution $\lambda(x)$, the reconstruction of the graph $\mathcal{G}$ obtained by user A after decoding is denoted by $\mathcal{A}$. We recall that the induced graph for user A is denoted by $\mathcal{G}_A$. With a slight abuse of notation, if user B is resolvable by user A based on $\mathcal{G}_A$, we write $\mathcal{G}_A(B) = 1$ and we write $\mathcal{G}_A(B) = 0$ otherwise. Using this notation, the PLR can be written as $\bar{p} = \Pr\{\mathcal{G}_A(B) = 0\}$. $\mathcal{A}_B$ denotes the graph that user A obtains after removing user B's slots from the reconstructed graph $\mathcal{A}$. Hence, we write $\mathcal{A}_B(A) = 1$ if user A concludes that user B resolves it using the reconstructed graph $\mathcal{A}$.

For the example in Fig. 2, user A uses the induced graph $\mathcal{G}_A$ shown in Fig. 2(b) for decoding. However, user A may not be able to fully reconstruct $\mathcal{G}$. In fact, user A can only resolve users B and E. Users D and F cannot be resolved because they form a so-called *stopping set*, a harmful graph structure

that makes decoding fail. A stopping set is a subset of VNs of non-zero degrees $\mathcal{S} \subset \mathcal{V}$ where all neighboring CNs of $\mathcal{S}$ are connected to $\mathcal{S}$ at least twice [9].

After decoding, user A reconstructs the graph $\mathcal{A}''$ shown in Fig. 2(c). Additionally, user A obtains the knowledge that slots 5 and 7 belong to a stopping set. It is worth noting that this is a stopping set from user A's perspective and not necessarily a stopping set in the original graph $\mathcal{G}$. Nonetheless, user A assumes that these slots cannot be used for decoding by any other user. Therefore, user A removes these slots, as well as all the edges connected to them, to obtain the graph $\mathcal{A}'$ shown in Fig. 2(d) (the removed slots and edges are shaded).

As the the last step to reconstruct $\mathcal{G}$, user A adds itself as a VN to the graph $\mathcal{A}'$ and connects it to the corresponding CNs. The reconstruction of the graph $\mathcal{A}$ is shown in Fig. 2(e). Since user A does not know who exactly transmitted in slots 1–3, these slots are shown with gray. User A uses this graph to run the decoding on behalf of other users, e.g., Fig. 2(f) shows the graph $\mathcal{A}_B$ that user A uses for decoding on behalf of user B. In this case, $\mathcal{A}_B(A) = 1$. However, in reality user B uses $\mathcal{G}_B$ for decoding and its true outcome is $\mathcal{G}_B(A) = 0$. Therefore, user A makes an erroneous decision about user B's awareness of user A, which happens due to partial knowledge about the slots user A uses for transmission.

C. Handshake Outcomes

If user A is in a stopping set contained in the original graph $\mathcal{G}$, then it will not be resolvable by any other user in the network. User A has no means to learn about this since this information is contained in the slots that it uses for transmission. Hence, user A can never be sure about its successful handshake decision. To describe the possible outcomes of the handshake algorithm and analyze their probabilities, we introduce the vector $\boldsymbol{g} = [\mathcal{G}_A(B), \mathcal{A}_B(A), \mathcal{G}_B(A)]$. If user A fails to resolve user B, i.e., $\mathcal{G}_A(B) = 0$, with a slight abuse of notation we say that $\mathcal{A}_B(A) = \mathsf{x}$, meaning that A cannot perform a handshake with B. All possible outcomes with the corresponding probabilities are summarized in Table I. In the table, p_1 is the probability that user A successfully detects communication failure at user B's side. p_2 is the probability that user A fails to detect communication failure at user B's side and erroneously assumes that user B successfully receives its packet. p_5 is the probability of correct handshake. p_3 and p_4 are auxiliary probabilities, where $p_3 = \Pr\{\mathcal{G}_B(A) = 0, \mathcal{G}_A(B) = 0\}$ is the probability that users A and B do resolve each other simultaneously. The sum $p_3 + p_4$ equals the probability that user A does not resolve user B, i.e., $p_3 + p_4 = \Pr\{\mathcal{G}_A(B) = 0\} = \bar{p}$.

Interestingly, the outcome $\boldsymbol{g} = [1, 0, 1]$ can not occur, which is formally proven in the following theorem.

Theorem 1. $\Pr\{\boldsymbol{g} = [1, 0, 1]\} = 0$.

Proof: $\mathcal{G}_A(B) = 1$ and $\mathcal{A}_B(A) = 0$ imply that user A is in a stopping set $\mathcal{S}$ of the graph $\mathcal{A}_B$. This stopping set $\mathcal{S}$ has to be present in the graph $\mathcal{G}_B$ as well since $\mathcal{A}_B$ is a subgraph of $\mathcal{G}_B$. Hence, $\mathcal{G}_B(A) = 0$, which completes the proof. $\square$

TABLE I: Possible outcomes of the handshake algorithm. $\boldsymbol{g} = [\mathcal{G}_\mathsf{A}(\mathsf{B}),\ \mathcal{A}_\mathsf{B}(\mathsf{A}),\ \mathcal{G}_\mathsf{B}(\mathsf{A})]$.

$\mathcal{G}_\mathsf{A}(\mathsf{B})$	$\mathcal{A}_\mathsf{B}(\mathsf{A})$	$\mathcal{G}_\mathsf{B}(\mathsf{A})$	$\Pr\{\boldsymbol{g}\}$	Event
1	0	0	p_1	Failure detected
1	1	0	p_2	False handshake
0	×	0	p_3	Auxiliary
0	×	1	p_4	Auxiliary
1	1	1	p_5	Successful handshake

III. PERFORMANCE ANALYSIS

A. Induced Distribution

The performance of CSA exhibits a threshold behavior, i.e., all users are successfully resolved if the channel load is below a certain threshold value when $n \to \infty$. The threshold depends only on the degree distribution and is obtained via density evolution [10]. A finite number of slots, however, gives rise to an error floor in the PLR performance. In [1] it was shown that the error floor can be accurately predicted based on the induced distribution observed by the receiver. The induced distribution for a degree-k receiver can be expressed similarly to (1), where

$$\lambda_d^{(k)} = \sum_{l=d}^{\min\{q,k+d\}} \frac{\binom{n-k}{d}\binom{k}{l-d}}{\binom{n}{l}} \lambda_l \tag{3}$$

is the fraction of users of degree d as observed by user A if it chooses degree k.

We define the PLR for a degree-d user as observed by a degree-k receiver as

$$\bar{p}_d^{(k)} = \frac{\bar{w}_d^{(k)}}{\bar{m}_d^{(k)}} = \frac{\bar{w}_d^{(k)}}{m\lambda_d^{(k)}}, \tag{4}$$

where $\bar{m}_d^{(k)}$ and $\bar{w}_d^{(k)}$ are the average number of all and unresolved degree-d users for a degree-k receiver, respectively. For degree-0 users, $p_0^{(k)} = 1$ for all k.

B. Analytical Results

In this section, we are interested in describing the probabilities p_1 and p_2. First, we observe that

$$p_1 + p_2 + p_3 = \Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0\} = \bar{p}. \tag{5}$$

From (5), we can immediately write that

$$p_1 + p_2 \le \bar{p}. \tag{6}$$

In the following, we tighten the bound in (6). The probability p_3 can be written as

$$p_3 = \Pr\{\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\} \Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\}. \tag{7}$$

The fact that the induced graphs for all users arise from the same original graph $\mathcal{G}$ gives rise to dependency between users' performance. This is expressed as the conditional probability in the right-hand side of (7).

Examining (7), we conjecture that

$$p_3 \ge \bar{p}^2. \tag{8}$$

For the asymptotic case, when $n \to \infty$, it is easy to show that $p_3 = \bar{p}^2$ since the probability for users A and B to use the same slots for transmission is zero.

For finite frame lengths, the rationale behind this conjecture is as follows. Let us take a closer look at the conditional probability in (7). We conjecture that

$$\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\} \ge \Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0\}. \tag{9}$$

Using the law of total probability, we can write

$$\begin{aligned}&\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\} \Pr\{\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\}\\&+\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 1\} \Pr\{\mathcal{G}_\mathsf{A}(\mathsf{B}) = 1\}\\&=\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0\}.\end{aligned}$$

Exploiting $\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0\} = \Pr\{\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\} = \bar{p}$ gives

$$\begin{aligned}&\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 1\}\\&= \frac{\bar{p}}{1-\bar{p}}\left(1 - \Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 0\}\right).\end{aligned} \tag{10}$$

Therefore, showing (9) is equivalent to showing

$$\Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0|\mathcal{G}_\mathsf{A}(\mathsf{B}) = 1\} \le \Pr\{\mathcal{G}_\mathsf{B}(\mathsf{A}) = 0\}. \tag{11}$$

Assuming an unequal error protection (UEP) property [6], i.e., $\bar{p}_{l+1}^{(k)} < \bar{p}_l^{(k)}$ for a given k, it can be shown that (11) and, hence, (8)–(9) hold. The derivations are omitted due to lack of space. The UEP property does hold when $n \to \infty$. It is also easy to find a counterexample for extremely short frame lengths, when it does not hold. For instance, consider a unicast system ($k = 0$) with two users, two slots, and the distribution $\lambda(x) = 0.5x + 0.5x^2$. For this toy example, the PLR for users of different degrees can be found by hand, yielding $\bar{p}_1^{(0)} = 0.25$ and $\bar{p}_2^{(0)} = 0.5$. However, from our extensive simulations, we conjecture that the UEP property does hold for sufficiently large values of n. Proving it rigorously and characterizing sufficient frame lengths is subject of ongoing work.

Using the conjecture in (8) together with (5), we can write a tighter version of (6) as

$$p_1 + p_2 \le \bar{p}(1 - \bar{p}). \tag{12}$$

Next section presents numerical results and confirms the conjectures made in this section.

IV. NUMERICAL RESULTS

In Fig. 3, we show simulation results for two different distributions and frame length $n = 200$. The distribution $\lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$ is taken from [10], where it was optimized for classical CSA based on the threshold obtained via density evolution. The fraction of degree-2 users was limited to 0.25 to yield low error floor. The second distribution, $\lambda(x) = 0.86x^3 + 0.14x^8$, was optimized in [1] based on error floor approximations to provide low error floor for B-CSA.

The red and the green curves show the probabilities p_1 and p_2, respectively, and characterize the handshake performance.
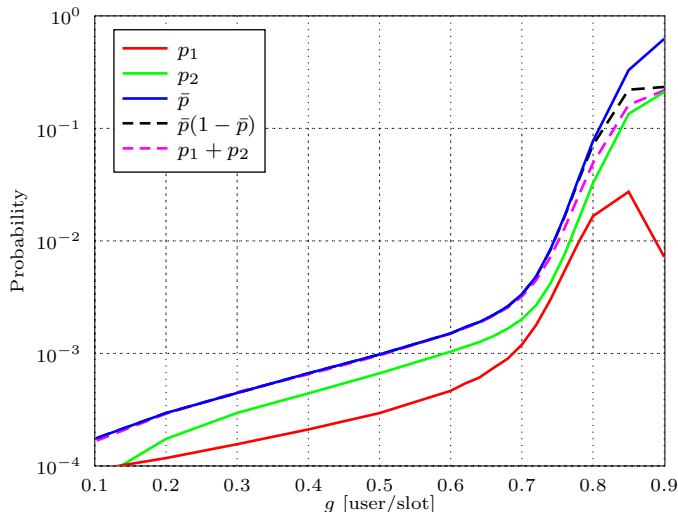
(a) $\lambda(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$.
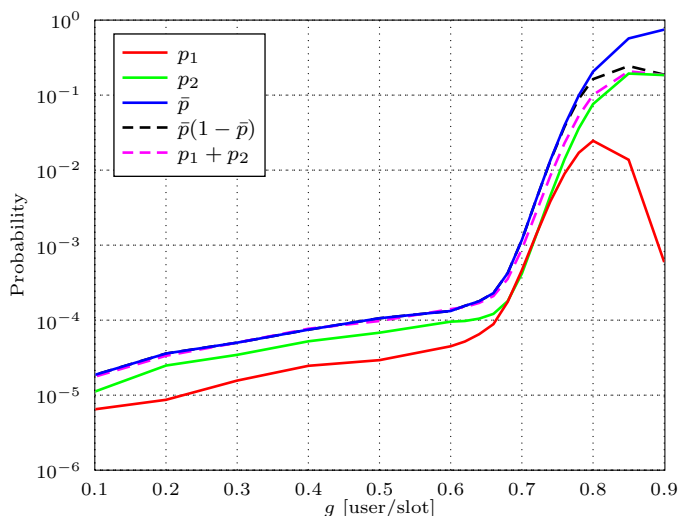
(b) $\lambda(x) = 0.86x^3 + 0.14x^8$.

Fig. 3: Handshake performance of user A in B-CSA for two different distributions and the frame length of $n = 200$ slots.

The blue curves show the PLR. From Fig. 3 we observe that for low to moderate channel load

$$p_1 + p_2 \approx \bar{p}(1 - \bar{p}). \tag{13}$$

Using (5), we conclude that $p_3 \approx \bar{p}^2$, which, using (7), leads to

$$\Pr\left\{\mathcal{G}_{\mathsf{B}}(\mathsf{A}) = 0 | \mathcal{G}_{\mathsf{A}}(\mathsf{B}) = 0\right\} \approx \bar{p}. \tag{14}$$

Therefore, for low to moderate channel load, the probability for users to overlap in some slots is very small, which explains that the correlation between users' performance is negligible.

Bearing in mind that $\bar{p} \ll 1$ in the error floor region, we

can further simplify (13) and write

$$p_1 + p_2 \approx \bar{p}. \tag{15}$$

In other words, the probability $\Pr\left\{\mathcal{G}_{\mathsf{B}}(\mathsf{A}) = 0\right\} = \bar{p}$ consists of p_1 and p_2, so that user A manages to detect communication failure events $\mathcal{G}_{\mathsf{B}}(\mathsf{A}) = 0$ in $p_1/\bar{p} \approx 30\%$ of the cases for both distributions. We remark, however, that this ratio may change depending on the distribution. This fact suggests a new design criterion for optimizing the degree distribution, i.e., the minimization of the probability of false handshake p_2. We also remark that the sum of $p_1 + p_2$, shown with dashed purple curves, is strictly smaller than $\bar{p}(1 - \bar{p})$ (black dashed curves), which is in agreement with the bound in (12).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a probabilistic handshake algorithm for vehicular communications based on B-CSA. In the rare cases of communication failure between two users, this event can be detected by one of the users. The simulation results show that around 30% of such events can be detected for the considered distributions. We also proposed analytical bounds on the performance of the handshake algorithm, which match well the simulation results. The analytical bounds rely on the UEP property of CSA, for which a rigorous proof for finite frame lengths is left for future work.

REFERENCES

[1] M. Ivanov, F. Brännström, A. Graell i Amat, and P. Popovski, "All-to-all broadcast for vehicular networks based on coded slotted ALOHA," in *Proc. IEEE Int. Conf. Commun. Workshop*, London, UK, June 2015.

[2] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded slotted ALOHA," in *Proc. IEEE Int. Conf. Commun.*, Kyoto, Japan, June 2011.

[3] 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.

[4] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: Applying codes on graphs to design random access protocols," *IEEE Commun. Mag.*, 2015 (to appear), available at http://arxiv.org/abs/1405.4127.

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

[6] M. Ivanov, F. Brännström, A. Graell i Amat, and P. Popovski, "Error floor analysis of coded slotted ALOHA over packet erasure channels," *IEEE Commun. Lett.*, vol. 19, no. 3, pp. 419–422, Mar. 2015.

[7] C. A. Sunshine and Y. K. Dalal, "Connection management in transport protocols," *Computer Networks*, vol. 2, no. 6, pp. 454–473, Dec. 1978.

[8] ETSI EN 302 637-2: Draft V.0.0.5, "Intelligent transport systems (ITS); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service," Tech. Rep., June 2012.

[9] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1459–1473, June 2002.

[10] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.