Codes for the Noisy V2V Random Access Channel without Feedback

Abstract

I. Introduction

Growing interest to the device-to-device (D2D) communication, that is, direct communication between devices without involving base station motivates a study of new network scenarios. In particular, in the vehicle-to-vehicle (V2V) communication, geographical location data, direction of movement, as well as some emergency information about fast changes in the traffic situation should be transmitted and kept regularly updated. The main specific requirements to V2V data transmission networks are

a) no relays and central stations,
b) no feedback is used,
c) users cannon cooperate but all of them can decode information sent over the channel by other users,
d) low delivery delay (short life time of messages)
e) reliability, i.e. high probability of successful delivery before the deadline.
f) no inter-layer coding and decoding are allowed. Standard physical layer equipment is used. Any user whose communication device is compliant with the standard should be able to participate in the network. However, input data of the encoder can be changed, for example, auxiliary information (header) can be added to the codeword.

The requirements listed above limit the ability to directly apply existing communications protocols for V2V communications. Next, we overview the existing approaches to constructing V2V networks. We split the overview in two parts. In the first part, we discuss solutions based on multiple access to the communication channel applicable to V2V networks. The second part is devoted to error correcting code based techniques in this application.

A. Channel access strategies

Almost all well-analyzed random access strategies used in D2D communications (see, for example, [1] and references therein) imply centralized communication controlled by the base station (see [2] for recent results in this area).

In V2V communications, typically, a common communication channel is used for random access by involved users without base station. Two main decentralized random access based scenarios are carrier sense multiple access (CSMA) [3] and a self-organizing time division multiple access (STDMA) [4], [5]. In particular, an STDMA-based protocol for V2V communication is considered in [6].

Another distinguishing property of V2V networks is that there are severe restrictions on the delivery delay. This limits application of CSMA to V2V communication since this technique can lead to unbounded delays. The STDMA approach provides finite transmission delay and is used for ship-to-ship communication. Delay restrictions also imply that algorithms based on retransmissions of lost packets cannot be used and only multiple access channels without feedback can be considered in decentralized V2V networks.

Pioneering papers in multiple access based communication area are [7]–[9]. They study both capacity of the multiple access noiseless and noisy channels and coding techniques achieving their capacities. Random access schemes based on packet loss (erasure) correcting codes for recovering parts of the packets lost in collisions and on successive interference cancellation (CSI) method [10], [11] are analyzed in [12]. The idea behind CSI approach is that interference caused by collided packets can be reduced after detecting an uncollided copy of the packet participating in collision. However, this method makes strong requirements for network synchronization and requires knowledge of precise signal powers of network users. Long low-density parity-check (LDPC) codes approaching capacity in the random access schemes with CSI are considered in [12].

In Table I we present a short overview of the known multiple access techniques and characterize them from the point of view of applicability to V2V communication networks.

B. *Codes for reducing packet losses due to collisions*

Both centralized and decentralized random access based scenarios imply that two or more users can send their packets simultaneously which leads to a collision, that is, loss of the corresponding fully or partially conflicted packets. Two main approaches to resolving collisions are retransmissions and recovering lost packets by using error-correcting codes. Typically, (see, for example, [7], [8], [12]), packet collisions are considered as the only source of packet losses and non-conflicted packets are assumed to be error free. In [18] channel noise is also taken into account.

Starting with [20], [7] a number of works consider so-called *conflict-avoiding codes* whose codewords can be used as protocol sequences for active users. However, they are based on different assumptions and, consequently, optimize codes according to different criteria. For example, in [7] packets are allowed to conflict. It is assumed that the corresponding lost packet (erasure) is then recovered (corrected) at the receiver side by using an erasure correcting code. Both numbers of allowed packet conflicts for a user and parameters of the erasure correcting code are chosen in a way as to guarantee correction of any pattern of erased packets that can occur if all users are active. However, this approach requires rather long codes to be used. In [21] a generalization of code construction in [7] is suggested which leads to significantly shorter protocol

TABLE I
MULTIPLE ACCESS TECHNIQUES WITH APPLICATION TO V2V(D2D) AD HOC NETWORKS

Class of algorithms	Contention resolution	Examples of coding techniques	Applicability to V2V
Random access with or without relays or central stations	Retransmission of colliding packets. Feedback-based collision resolution	ALOHA [13], distributed queuing (DQ) [14], Successive interference cancellation tree algorithm SICTA [11]	Low applicability: feedback is assumed. Packet delay is not controlled
	Listening channel and waiting for idle frames. Either random back-offs or scheduling algorithm is used for reducing probability of collisions	IEEE 802-11p CSMA [5], [15], VeMAC [6], STDMA [4], [5],	Low applicability: Low packet delay cannot be guaranteed. Collided packets are lost
Multiple access with error correcting codes for reconstructing colliding packets	Block erasure correcting codes used for recovering lost packets	Massey et al [7], Tsybakov et al [8], Hui [9],	Limited applicability: capacity is upper-bounded by $1/e$
	Both retransmissions and coding, either block or convolutional	Network-coded TCP [16]	Not applicable: encoder computes linear combinations of packets from different users
Joint network-channel coding schemes	Recovering data from collided packets after successful reconstruction of some of collided packets	LDPC codes with SIC [12], Non-Binary joint network-channel codeng (JNCC) [17], Slotted coding random access multiplexing (SCRAM) [18]	Limitations: Inter-layer coding is not always possible, delays are large
Low-delay joint coding and scheduling	Low-memory convolutional code are used for recovering packet losses both at physical and network layers. Code parameters are matched with access scheduling	This study and [19].	Applicable. Delay is low and controlled. Capacity is close to SIC capacity.

sequences than those studied in [7]. In [22], the protocol sequences which guarantee successful delivering a certain number of packets for a given maximum number of active users (intersections) in a slot are constructed. A few code constructions with and without additional zeros at the end of codewords are considered. In particular, two constructions of protocol sequence for two-user case are presented. A generalization to more-than-two-user case is suggested. Recovering of the lost packets are not assumed. A code suitable for three active users is constructed in [23]. It has a property that each from any three active users can transmit a packet of information in one of three attempts to do it during a given number of time slots equal to the codelength without a collision with other active users. The constructed code has the maximal total number of users (code cardinality) for given parameters. Similar problem statement is considered in [24], where conflict-avoiding codes for more than three active users are studied.

In this paper, we study a decentralized random access based V2V system functioning in small to moderate load conditions (typical number of network users is a few dozens). We suggest a new communication model without feedback where packet losses (erasures) are resulted from two sources: collisions of packets at the network level and noise corruptions at the physical level. We suggest a new class of protocol sequences which guarantee that each user can have at most a predetermined number of conflicts during the transmission frame. The corresponding number of packet losses are corrected by packet loss (erasure) correcting code of the user. In such a way, in case of noiseless channel at the physical layer error-free transmission is provided.

In particular, we suggest a new graph-based code which we call *confict-lost packet correcting code*. This code is tailored to a decentralized V2V communication protocol. The corresponding protocol sequences are chosen from the incidence matrix of the regular graph. The occurred due to conflicts packet losses together with packet losses detected at physical layer of the network are corrected by using erasure correcting convolutional code with sliding-window decoding. Parameters of the latter code are matched to the estimated probability distribution of the possible conflicts at the MAC layer. It is guaranteed *conflict-resolvable* transmission if the number of active users M does not exceed half of the girth g of the underlying graph. For $M > g/2$ the probability of a given number of conflicts is estimated.

Two classes of non-binary error-correcting codes for correcting erasures are analyzed. Important feature of V2V communications is a short lifetime of the transmitted information. This restriction narrows down the number of decoding techniques suitable for this application. Since low delay, low decoding complexity, and probability of successful delivering the entire message are main requirements in the V2V communication, we modify the decoding algorithm for these two classes of codes to tailor it to the system needs.

II. SYSTEM MODEL

Before formulating the new data transmission strategy we make some assumption about the system.

- *Physical layer assumptions*.
  We assume that an AWGN channel with physical layer signaling according to one of the wireless communication standards, for example, WiFi standards IEEE 802.11, is used for random multiple access. Although, mobile users can be located on different distances from each other and channel conditions for them can differ we assume that the same physical level code is used for all of them.
  There are two sources of packet losses: packet collisions caused by choosing the same slot by more than one user and packet losses caused by corruption the packets by the channel noise. It is assumed that both types of packet losses are detected at the physical level with probability 1. This assumption is supported by the fact that error correcting codes from the wireless standards have low probability of undetected error. Moreover, physical level protocols additionally use cyclic redundancy checks (CRCs) to protect data. Based on this assumption,

we consider erasure-only correcting codes applied for correcting both types of packet losses (erasures).
We assume that all users presented in the network can receive and decode information from all other users.

- *User model.*
In each time moment M mobile users are broadcast or multicast or communicate with each other on this channel, where M is a random number with a given probability distribution which will be specified later. Each user can address messages to one or more users. In order to avoid synchronization problems and taking into account a specific of transmitted information in this scenario, we suggest that each user broadcasts a message to all other users. In such a problem statement we do not lose generality since if a message is addressed to a single user or a group of users it can be additionally encrypted. In such a scenario, a specification of the recipient is contained in the packets overheads.
- *Messages and packets*
Network users send messages consisting of one or more packets. The packet sizes are the same for all users and match physical layer formats (codelengths, coderates, CRC, etc.). A message is considered as delivered successfully if all its packets are delivered successfully. Each packet is transmitted during one time slot and the users are aware about starts and ends of the slots. In each time slot a user can either send a packet or be silent.
- *Timing.*
The time is discrete, duration of one slot is considered as one time unit. The entire time axis is split into frames of length T time units and subframes of length τ units, such that number of subframes in each frame $w = T/\tau$ is an integer number. There is no frame synchronization in the system but there is subframe synchronization. It means that the starts of subframes are known to all users. This assumption does not constrain generality since the time offset value fro the packet can be included into the header of the packets at the cost of negligible loss in data transmission rate.
- *Packet loss correcting codes and redundant packets*
Messages of users are split into non-overlapped blocks of K packets and each block is encoded into the codeword of length $N \leq w$ packets by using systematic (N, K) erasure correcting code. It means that each block of K message packets is followed by $(N - K)$ redundant packets.
For so-called *block* codes, the block length N is equal to the codelength. *Convolutional* codes may have an arbitrary codelength determined by the message length. For this class of codes, N and K denote the length of codeword subblock in packets and the number of message packets corresponding to one subblock, respectively.
For the lock (N, K) code up to $N - K$ packet losses among N transmitted coded packets can be recovered. The block code is called *maximum distance separable*

(MDS) if any combination of $N-K$ lost packets (erasures) out of N packets can be corrected. Erasure correcting capability of convolutional code will be discussed below in Section.

- *Protocol sequences.*
  Each user transmits packets according to a schedule which is represented by a binary sequence of length equal to the frame length. This sequence we call *protocol sequence*. Ones in the protocol sequence correspond to time slots where packets are to be sent. If the number of ones in the protocol sequence is less than the number of packets in the message, the sequence is repeated periodically as many times as needed for sending the entire message. ??? In the multiple access protocol below, the Hamming weight of the protocol sequences is equal to the codelength or to the subblock length in packets N if a packet loss correcting block code or a packet loss correcting convolutional code is used, respectively.
  The full list of allowed protocol sequences (schedules) is known to all users. It is represented as a binary (nonlinear) code with $L \gg M$ codewords of length T and constant Hamming weight N.
- *Headers.*
  In order to make it possible to use the protocol below the following information is to be included into the header of each transmitted packet:
  1) Protocol sequence number in the list of protocol sequences.
  2) The message size measured in number of slots (packets).
  3) The packet number in the message.

  This set of numbers we call *packet ID* (PID). Typically, 8 bits is enough to transmit each of three numbers, 3 bytes in total. Therefore overhead data size added to the packet is negligible.

For convenience, we collected all notations for variables used for the system description in Table II.

Summarizing the description of the system, we can say that messages of users are split into blocks of K packets and each block (subblock in case of convolutional coding) is encoded into the codeword of length N packets by using systematic (N, K) erasure correcting code. Users are considered as active if they started transmission of a codeword of length N or non-active otherwise. Active users are not aware about starts of length T frames of the other active users but know starts of subframes of length τ. Protocol sequences are chosen in such a way that with high probability not more than $s \leq N-K$ out of N packets of each active user are conflicted with packets of other active users.

Each packet represents one of a few codewords of the (n, Rn) error correcting code $\mathcal{C}$ at the physical layer, where n and R are codelength and coderate of $\mathcal{C}$, respectively. It is assumed that codes from the wireless standards (e.g., IEEE 802-11) are used.

At the decoder side packets of active users are decoded by the decoders of the code $\mathcal{C}$. We assume that the decoder either makes correct decision or detects an error and erases the corresponding packet. Erasers can appear due to channel noise of collision

TABLE II
NOTATIONS

Variable	Meaning
T	Frame length in slots
τ	Subrame length in slots
$w = T/\tau$	Number of subframes in each frame
M	Random number of users presented in the network
(N, K)	Length and dimension of the linear block code used for packet loss (erasures) correction / length of the subblock and number of message packets in each subblock of convolutional code
(n, k)	Length and dimension of the physical layer error-correcting code
L	Number of pre-constructed protocol sequences of Hamming weight N each
K_m, N_m	Number of packets and number of encoded packets corresponding to mth message
$K_{\max}$	Maximum number of packets in one message
$F_{\max}$	Maximum number of frames used for transmitting one message

caused by simultaneous using the same slot by more than one user. The sequence of packet delivery successes/losses for each active user is decoded by the (N, K) code in order to correct erasures (packet losses).

III. Multiple access protocol

In this section we formulate the message exchange protocol. Taking into account specific of V2V communication system we suggest a random access protocol without feedback. In [7] a multiple-access protocol without feedback, providing the same capacity $1/e$ as the slotted ALOHA (see [25] and references therein), was analyzed. Later it was shown [26] that capacity of the slot synchronized ALOHA system can be increased by using the so-called reservation protocol (R-ALOHA). In this system conflicts can occurre only at the reservation stage. After reservation of slots conflict-free transmission is assumed. The R-ALOHA protocol implies that time slots are organized into frames, all active users listen one frame and then in the next frame use a time slot according to its status in the previous frame. The R-ALOHA capacity is lowerbounded by $1/e$ and tends to 1 when message length grows. Notice, that R-ALOHA for V2V communication was suggested and analyzed in [27]. In this paper we aim at developing and analyzing a multiple access protocol without feedback whose capacity close to 1. Similar result was demonstrated in [12] for SIC based system. However, as mentioned before, SIC approach is not applicable for V2V communication since it leads to large delays and requires inter-layered decoding.

In the protocol below we assume that the block code is used. In this case consequtive frames are processed independently. Modification of some steps for the case of convolutional coding will be presented in Section below.

A. *Idle state*
The receiver is decoding all data broadcasted in the network. By using information in PIDs of the received packets a user can compute for all future frames which slots are occupied by the other active users.

B. *Transmission initialization*
- *Forming packets*
When a message is ready for transmitting the user splits the message into packets. Then PID is computed and attached to each packet. Let K_i be the number of packets to be transmitted by the ith user.
- *Encoding packets by the packet-loss correcting code.*
The K_i packets are divided into $\lceil K_i/K \rceil$ blocks. Each block is appended by $N-K$ redundant packets computed using the encoder of chosen erasure correcting code (see Section **??**).
- *Selecting a protocol sequence* (example)
The transmitter i
  - computes the number M of users presented in the network and the indices $i_1, ..., i_M$ of currently used protocol sequences;
  - for each slot of F_i future frames ($F_i T$ slots) the transmitter i computes the number of users s_t sending packets in slot t. Denote by $\boldsymbol{s} = (s_1, s_2, ...)$ the computed occupation sequence;
  - for each protocol sequence (including those which are currently used) the risk value R_i is computed as a function of $\boldsymbol{s}$. For example, R_i can be computed as a predicted number of collisions corresponding to the sequence number i;
  - selects a protocol sequence at random among those protocol sequences for which $R_i < R_{\max}$ where $R_{\max}$ is a predetemined threshold.

C. *Transmission*
Packets of the encoded sequence are transmitted according to the selected schedule (protocol sequence).

D. *Receiving and reconstructing of received data*
- Decompose a sequence of packets at the output of physical layer decoder into M sequences corresponding to M active protocol sequences according to the PIDs of the decoded packets. In case of receiving unrecoverable packet, mark as erased the packets in the positions of ones in this slot in the active protocol sequences.
- For each protocol sequence split a sequence of the received packets into code blocks using information from PIDs. Some positions will be marked as erased due to channel noise or collisions.
- Recover the erased packets if the number of erasures in the block does not exceed $N-K$.

To complete formalization of the protocol we have to provide additionally
- An erasure correcting code

- A set of protocol sequences (scheduling code)
- A strategy of selecting a protocol sequence from the scheduling code for a new-coming message or user.

In more detail the decoding procedure can be explained as follows.

The physical layer decoder decodes each packet and either outputs successfully decoded packet or erases the packet if error is detected. The successfully decoded packets are identified by using their PID and enter the next level decoder whose number is determined by the packet PID. The decoder of the next level checks the header of the obtained packet and if the packet number is equal to the decoder packet counter then the decoder outputs this packet to the recipient and store it in the buffer of size at most N packets. If a packet is erased then the next level decoder discovers this fact by observing that the obtained packet number and the decoder packet counter do not match. In this case the decoder tries to recover the erased packet from the other successfully decoded packets of the same codeword stored in the buffer.

Notice that the presented above description of the message exchange protocol implies using a block (N, K) erasure correcting code. If convolutional rate K/N code is used then the buffer size is equal to the size of decoding window.

Let us consider how the packet loss (erasure) correcting codes can be used in V2V network.

Denote by (N, K) a systematic code of length N with K information symbols over alphabet of size $Q = 2^k$, where k is the number of bits in one packet (symbol of (N, K) code). We also can split the packet into small parts (up to 1 bit long) and apply coding to parts of packets. In the systematic code, first K symbols are message symbols and the last $N - K$ symbols are redundant (check) symbols used for erasure correction. Typically (for so-called maximum distance separable (MDS) codes the entire codeword can be recovered from any K code symbols out of N symbols of a codeword. This means that any combination of $N - K$ packet losses will be successfully recovered.

Using packet loss (erasure) correcting codes increases amount of transmitted data due to redundant symbols which should be transmitted. On the other hand, these codes allow successful delivery of the message even if some message packets are lost due to the channel noise or collisions at the MAC layer. Thus, more intensive traffic can be allowed than in the case if any non-recoverable packet will destroy (make senseless) the entire message consisting of several packets.

IV. Protocol code. Examples and random ensemble

We first consider only block code. Let $N \leq w = T/\tau$, where N is the codelength. Thus, one codeword corresponds to one frame and one user occupies one of τ slots in N of w subrames.

In this case the total number of possible protocol sequences is

$$\mathcal{N} = \binom{w}{N} \tau^N.$$

Certainly, some combinations of protocols should be avoided. Assume that the packet loss correcting (N, K)-code is able to correct all combinations of $N-K$ packet losses in the frame. Then the protocol sequences overlapping in more than $N-K$ slots will lead to loss of one or more packets of the corresponding users.

One of possible solutions is to pre-construct a good set of allowed protocol sequences. The user which start transmission makes his choice based on the analysis of the last received subframe. Notice that choice should be random to avoid choosing the same protocol by two or more users starting transmission in the same subframe.

For purposes of analysis we will consider random protocol code. Two coding strategies will be studied, completely random strategy (Strategy R) and greedy random strategy (Strategy G) :

Strategy R:

1) Among $\binom{w}{N}$ equiprobable possibilities choose one set of N subframes where packets will be transmitted. Select at random N least loaded subframes by assigning equal probabilities to equally loaded subframes.
2) At each of N chosen subframes let Λ_i, $i = 0, 1$ be the number of slots occupied by i users. Let γ_i be a predetermined probability distribution preference weight function corresponding to slots with i users in the slot.
   assign equal probabilities p_i slots, which are occupied by i users. let
   choose one of τ available subframes with equal probabilities $1/\tau$.

Strategy G:

1) Use PIDs of users which were active at the last frame to compute occupation vector $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_w)$, where $\boldsymbol{x}_t = (x_{t1}, ..., x_{t\tau})$ and $x_{ij} = 1$ if slot is occupied and 0 otherwise.
2) Among $\binom{w}{N}$ equiprobable possibilities choose one set of N subframes where packets will be transmitted
3) At each of N chosen subframes choose one of τ available subframes with equal probabilities $1/\tau$.

V. Packet-loss recovering convolutional codes

In this application we consider nonbinary rate $R = (n-1)/n$ convolutional codes. The so-called Reed-Solomon (RS)-convolutional code was first presented in [28] and analyzed when applied to recovering lost packets in [19]. In this section, we describe this class of codes and briefly recall the properties of these codes, which are important for understanding the functioning of the system as a whole.

Any rate $R = (n-1)/n$ time-invariant convolutional code is determined by a semi-infinite parity-check matrix

$$H = \begin{pmatrix} H_0 & \mathbf{0} & \cdots & & & & & \\ H_1 & H_0 & \mathbf{0} & \cdots & & & & \\ H_2 & H_1 & H_0 & \mathbf{0} & \cdots & & & \\ \cdots & \cdots & \cdots & \cdots & & & & \\ H_m & H_{m-1} & H_{m-2} & H_1 & H_0 & \mathbf{0} & \cdots & \\ \mathbf{0} & H_m & H_{m-1} & \cdots & H_1 & H_0 & \mathbf{0} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}, \tag{1}$$

where m denotes the code syndrome memory [29] and H_i, $i = 0, 1, ..., m$ is of size $(n-1) \times n$. For the RS-convolutional code block-column submatrix $H^{(m)}$ has the form

$$H^{(m)} = \begin{pmatrix} H_0 \\ H_1 \\ H_2 \\ \cdots \\ H_m \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \alpha^m & \cdots & \alpha^{m(n-1)} \end{pmatrix} \tag{2}$$

and α denotes a primitive element of the Galois field GF(2^c) whose elements can be represented as c-tuples of bits.

Encoding for this class of codes can be implemented by using parity-check matrix (1). We construct a codeword in the systematic form calculating a single parity-check symbol of each subblock recursively. The encoding procedure can be simplified by rewriting columns of $H^{(m)}$ in (2) in the reverse order. For example, for $R = 3/4$ and $n = 4$ reordered matrix has form

$$H^{(m)} = \begin{pmatrix} H_0 \\ H_1 \\ H_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \alpha^3 & \alpha^2 & \alpha & 1 \\ \alpha^6 & \alpha^4 & \alpha^2 & 1 \end{pmatrix} \tag{3}$$

Denote by

$$\tilde{H} = \begin{pmatrix} 1 & 1 & 1 \\ \alpha^3 & \alpha^2 & \alpha \\ \alpha^6 & \alpha^4 & \alpha^2 \end{pmatrix}$$

the information part of $H^{(m)}$. The codeword of the RS-convolutional code over GF(2^c) can be written in the form $\boldsymbol{u}_1, v_1, \boldsymbol{u}_2, v_2, ...$ where $\boldsymbol{u}_i = (u_{i1}, u_{i2}, u_{i3})$ are message blocks consisting of three information symbols from GF(2^c) and v_i are the corresponding parity-check symbols. Denote by $\boldsymbol{s}_i = (s_{i1}, s_{i2}, s_{i3}) = \boldsymbol{u}_i \tilde{H}^{\mathrm{T}}$ a partial syndrome computed for the information block. It is easy to see that the first check symbol $v_1 = s_{11}$ and the next one depends on s_{12}, etc. The following recurrent equations describe encoding for the RS-convolutional code.

$$v_1 = s_{1,1},\ v_2 = v_1 + s_{1,2} + s_{2,1},$$
$$v_i = v_{i-2} + v_{i-1} + s_{i-2,3} + s_{i-1,2} + s_{i,1},\ i = 3, 4, ... \tag{4}$$

Encoding delay is equal to the encoder block length nc since each parity-check symbol is computed immediately after receiving a new block of $(n-1)c$ latest information bits.

Decoding of RS-convolutional codes can be performed by using either sliding-window (SW)-decoding or its simplified version sliding-window belief propagation (SWBP)-decoding (see for details [19]). In general terms, decoding is reduced to solving the following system of linear equations

$$\boldsymbol{z} H_{I(\boldsymbol{e})}^{\mathrm{T}} = \boldsymbol{s}(\boldsymbol{e}) \ , \tag{5}$$

where $I(\boldsymbol{e})$ is the set of positions (inside the decoding window) of ν lost packets, $\boldsymbol{z} = (z_1, z_2, ..., z_\nu)$ is the vector of unknowns located on positions $I(\boldsymbol{e})$, $H_{I(\boldsymbol{e})}$ denotes columns of parity-check matrix corresponding to $I(\boldsymbol{e})$ and $\boldsymbol{s}(\boldsymbol{e})$ is the syndrome vector computed by using non-erased (successesfully received inside the decoding widow) packets.

Erasure correcting capability of the RS-convolutional codes was analyzed in [19]. The minimum Hamming distance of the rate $(n-1)/n$ RS code is equal to 4, that is recovering 3 lost packets is guaranteed. Moreover, SW-decoding of these codes can correct almost all patterns of 4 lost packets.

VI. Throughput analysis

In this section we obtain upper bound on the achievable system capacity. In order to simplify analysis we do the following assumptions:

- M users are active
- Message of each user consists of K packets which are transmitted on K slots whose numbers are determined by a protocol sequence.
- Protocol code is a set of orthogonal (non-intersecting) sequences of length K and the number of protocol sequences is much larger than the number of users.
- Each packet is transmitted during one time slot

Denote by p_i a probability that a given protocol sequence is occupied by the ith user. The probability of conflict-free message transmission is upper-bounded by

$$P_{\text{success}} \leq p_i \prod_{j=1, j\neq i}^{M} (1-p_j) \tag{6}$$

$$\leq p(1-p)^{M-1}, \tag{7}$$

where $p_i = p$, $i = 1, 2, ..., M$. The inequality in (6) means that the probability of conflict-free transmission is less than or equal to the probability that none of M users except the ith user chose the ith protocol sequence and the inequality (7) follows from the fact that the maximum in the right hand side of (6) is achieved on equal probabilities.

Now we estimate the probability p. Taking into account that the maximal utilization of the network, that, 1 packet/time slot is obtained if all M users transmit K packets each and each of the users transmit its K packets successfully with probability p we obtain that p is upper-bounded by

$$MKp \leq 1,\, p \leq \frac{1}{MK}.$$

Finally, by inserting $p = 1(MK)$ in (7) we obtain

$$P_{\text{success}} \leq \frac{1}{MK}\left(1 - \frac{1}{MK}\right)^{M-1}.$$

Then the average number of transmitted packets is upper-bounded by

$$MKP_{\text{success}} \leq \left(1 - \frac{1}{MK}\right)^{M-1}$$

When number of users M tends to infinity we obtain that system capacity is upper-bounded as

$$C_s \leq e^{-1/K}$$

. It is easy to see that the upper bound on capacity coincides with $1/e$ for $K = 1$ and tends to 1 when K grows.

VII. Example

In this section we demonstrate efficiency of the proposed protocol compared to time-sharing. Let $M = 4$ users use the following set of orthogonal protocol sequences

$$\mathcal{P} = \left(\begin{array}{cccc|cccc|cccc|cccc}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right)$$

to transmit $K = 4$ packets each. Assuming a noiseless physical layer channel, in such conflict-free system $MK = 16$ packets will be successfully delivered. However, if more than M users transmit their packets, some of packets collide and will be lost.

In the proposed system with erasure correcting we allow to use the following intersecting protocol sequences

$$\mathcal{P}_s = \left(\begin{array}{cccc|cccc|cccc|cccc}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right).$$

In this case $M = 5$ users can transmit $MK = 20$ packets. However, since protocol sequences intersect with each other in 1-4 positions the corresponding packets will be lost. At the colliding positions users can flip a coin, that is, use a position with probability 1/2. This reduces the number of lost packets to 1-2. In order to be able to recover these packets we use erasure correcting code of rate $R = 3/4$ with minimum distance 3. In other words, among 20 transmitted packets there are $MK(1-R) = 5$ redundant packets. By using this code for each user lost packets can be recovered. Thus, we can increase the number of network users keeping the efficiency of the system.

Taking into account erasure correcting capability of SW-decoding of RS-convolutional code we can use longer protocol sequences

$$\mathcal{P} = (\mathcal{P}_s|\mathcal{P}_s)\,,$$

and correct most of patterns of more than 2 lost packets.

Notice that by jointly selecting protocol sequences and erasure correcting codes the number of network users can be significantly increased compared to the number of users in network with time-division strategy.

The presented example is illustrative. Further analysis and design of protocol sequences should be done.

VIII. Simulations

IX. Conclusions

References

[1] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? a survey of alternatives," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 4–16, 2013.

[2] Z. Alavikia and A. Ghasemi, "Collision-aware resource access scheme for LTE-based machine-to-machine communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4683–4688, 2018.

[3] T. V. Nguyen, F. Baccelli, K. Zhu, S. Subramanian, and X. Wu, "A performance analysis of CSMA based broadcast protocol in VANETs," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2805–2813.

[4] T. Gaugel, J. Mittag, H. Hartenstein, S. Papanastasiou, and E. G. Ström, "In-depth analysis and evaluation of self-organizing TDMA," in *2013 IEEE vehicular networking conference*. IEEE, 2013, pp. 79–86.

[5] K. Bilstrup, E. Uhlemann, E. Ström, and U. Bilstrup, "On the ability of the 802.11 p MAC method and STDMA to support real-time vehicle-to-vehicle communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 902414, 2009.

[6] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-based MAC protocol for reliable broadcast in vanets," *IEEE transactions on mobile computing*, vol. 12, no. 9, pp. 1724–1736, 2012.

[7] J. Massey and P. Mathys, "The collision channel without feedback," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 192–204, 1985.

[8] B. S. Tsybakov and N. Likhanov, "Packet switching in a channel without feedback," *Problemy Peredachi Informatsii*, vol. 19, no. 2, pp. 69–84, 1983.

[9] J. Y. Hui, "Multiple accessing for the collision channel without feedback," *IEEE transactions on vehicular technology*, vol. 33, no. 3, pp. 191–198, 1984.

[10] E. Casini, R. De Gaudenzi, and O. D. R. Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access schemefor satellite access packet networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, 2007.

[11] Y. Yu and G. B. Giannakis, "High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4628–4639, 2007.

[12] E. Paolini, G. Liva, and M. Chiani, "Graph-based random access for the collision channel without feedback: Capacity bound," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–5.

[13] N. Abramson, "The ALOHA system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*. ACM, 1970, pp. 281–285.

[14] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, aloha!" *IEEE access*, vol. 4, pp. 2029–2044, 2016.

[15] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in *VTC Spring 2008-IEEE Vehicular Technology Conference*. IEEE, 2008, pp. 2036–2040.

[16] J. K. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network coding meets tcp," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 280–288.

[17] Z. Guo, J. Huang, B. Wang, S. Zhou, J.-H. Cui, and P. Willett, "A practical joint network-channel coding scheme for reliable communication in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2084–2094, 2012.

[18] S. Nafie, J. Robert, and A. Heuberger, "SCRAM: A Novel Approach for Reliable Ultra-Low Latency M2M Applications," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[19] I. Bocharova, B. Kudryashov, N. Lyamin, E. Frick, M. Rabi, and A. Vinel, "Low delay inter-packet coding in vehicular networks," *MDPI*, p. in press, 2020.

[20] L. A. Bassalygo and M. S. Pinsker, "Limited multiple-access of a non-synchronous channel," *Problemy Peredachi Informatsii*, vol. 19, no. 4, pp. 92–96, 1983.

[21] V. da Rocha, "Short protocol sequences for the collision channel without feedback," in *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No. 01CH37252)*. IEEE, 2001, p. 206.

[22] B. S. Tsybakov and A. R. Rubinov, "Some constructions of conflict-avoiding codes," *Problems of Information Transmission*, vol. 38, no. 4, pp. 268–279, 2002.

[23] V. I. Levenshtein and V. D. Tonchev, "Optimal conflict-avoiding codes for three active users," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.* IEEE, 2005, pp. 535–537.

[24] T. Baicheva and S. Topalova, "Optimal conflict-avoiding codes for 3, 4 and 5 active users," *Problems of Information Transmission*, vol. 53, no. 1, pp. 42–50, 2017.

[25] I. Gitman, "On the capacity of slotted ALOHA networks and some design problems," *IEEE Transactions on Communications*, vol. 23, no. 3, pp. 305–317, 1975.

[26] S. S. Lam, "Packet broadcast networks — A performance analysis of the R-ALOHA protocol," *IEEE transactions on Computers*, no. 7, pp. 596–603, 1980.

[27] Y.-A. Kim and M. Nakagawa, "R-ALOHA protocol for SS inter-vehicle communication network using head spacing information," *IEICE transactions on communications*, vol. 79, no. 9, pp. 1309–1315, 1996.

[28] P. Ebert and S. Tong, "Convolutional Reed-Solomon codes," *Bell Labs Technical Journal*, vol. 48, no. 3, pp. 729–742, 1969.

[29] R. Johannesson and K. S. Zigangirov, *Fundamentals of convolutional coding*. John Wiley & Sons, 2015, vol. 15.