Lectures on Multiple Access Channels

László Györfi Sándor Győri Bálint Laczay
Miklós Ruszinkó

Preface

We wish to thank ... for commenting on an earlier version of the manuscript and for the many valuable suggestions. We would also like to thank ... for fruitful discussions from which the book has benefited greatly.

This work relates to Department of the Navy Grant N00014-04-1-4034 issued by the Office of Naval Research International Field Office, and was sponsored also by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-05-1-3017. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.

Contents

Chapter 1

Introduction

From the viewpoint of information theory the **multiple-access channel** is a black-box operating in discrete time with a fixed number of inputs and one output. There are also extended models, with multiple outputs, the so called interference channels, but we do not deal with them now (c.f. Shannon (1961) and Ahlswede (1971)). We consider that one user “sits” at each input, so instead of inputs we usually refer to users. Let us denote the number of users with t. The input and output alphabets of the channel are denoted by I and O, respectively.

In the information theory one deals only with the case of memoryless channel, so to fully describe the channel, it is enough to give the channel transition probabilities $p(y|x_1x_2\ldots x_T)$:

$$\mathbf{P}\left(Y = y\middle|X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T\right) = p(y|x_1x_2\ldots x_T) \qquad \forall(x_1, x_2, \ldots, x_T) \in I^T, \forall y \in O.$$

Here $X_1, X_2, \ldots, X_T$ denote the T inputs of the channel, while Y denotes the output.

Each user of the channel has a so called **component code**. A component code is a set of fixed codewords, one for each possible message of the user. We assume, that all these codewords of all users have a common length n. So the component code of the i^{th} user can be written as

$$C_i = \left\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{|C_i|}^{(i)}\right\} \subseteq I^n.$$

The **code** itself is the set of the component codes defined above:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_T\}.$$

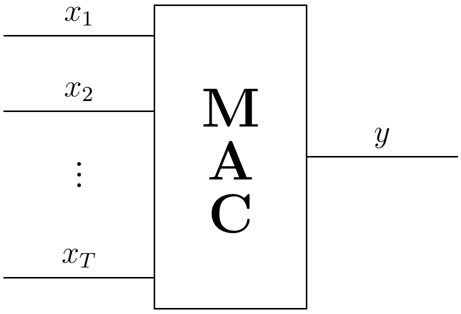

Figure 1.1: The Multiple Access Channel

The **message** user i wants to send is denoted by the random variable $K_i \in \{1, 2, \dots, |C_i|\}$. To send this message, the user transmits $\mathbf{x}^{(i)}_{K_i}$ through the channel. We will use a further restriction, that the codewords sent by the users are bit and block synchronized. This means, that at a given instant, all users are sending the same component of their codewords. Say, when user i is sending the m^{th} component of his codeword ($[\mathbf{x}^{(i)}_{K_i}]_m$), then user j is also sending the m^{th} component ($[\mathbf{x}^{(j)}_{K_j}]_m$). So we can treat the channel output as a vector of length n.

Since the channel is memoryless, we can write a simple formula for the distribution of the vectorial channel output, conditionally on that user i sends its k_i^{th} codeword:

$$\begin{aligned}
&\mathbf{P}\left(\mathbf{Y} = \mathbf{y} \middle| K_1 = k_1, K_2 = k_2, \dots, K_T = k_T\right) \\
&\quad = \mathbf{P}\left(\mathbf{Y} = \mathbf{y} \middle| \mathbf{X}_1 = \mathbf{x}^{(1)}_{k_1}, \mathbf{X}_2 = \mathbf{x}^{(2)}_{k_2}, \dots, \mathbf{X}_T = \mathbf{x}^{(T)}_{k_T}\right) \\
&\quad = \prod_{m=1}^{n} \mathbf{P}\left(Y = [\mathbf{y}]_m \middle| X_1 = [\mathbf{x}^{(1)}_{k_1}]_m, X_2 = [\mathbf{x}^{(2)}_{k_2}]_m, \dots X_T = [\mathbf{x}^{(T)}_{k_T}]_m\right).
\end{aligned}$$

To define the **error probability** of a given code $\mathcal{C}$, we must have a decoding function for each user:

$$d_i \colon O^n \to \{1, 2, \dots, |C_i|\} \qquad \forall i \in [T].$$

(Here $[T]$ denotes $\{1, 2, \dots, T\}$.) The aim of the decoding function d_i is to recover the message K_i of the i^{th} user from the channel output vector ($\mathbf{Y}$). The error probability (P_e) of a given code is defined as the probability of making a mistake for at least one user, considering the optimal decoding functions:

$$P_e(\mathcal{C}) = \inf_{d_1, d_2, \dots, d_T} \mathbf{P}\left(\{\exists i \in [T] \colon d_i(\mathbf{Y}) \neq K_i\}\right).$$

Here we consider, that the random variables $K_1, K_2, \ldots, K_T$ are independent and uniformly distributed over the set of possible messages ($\{1, 2, \ldots, |C_i|\}$):

$$\mathbf{P}\left(K_i = k\right) = \frac{1}{|C_i|} \qquad \forall i \in [T], \forall k \in \{1, 2, \ldots, |C_i|\}.$$

The **code rate** of a given code for user i is defined as

$$r_i(\mathcal{C}) = \frac{\log |C_i|}{n},$$

while the rate vector of a given code is formed by arranging these quantities into a vector:

$$\mathbf{r}(\mathcal{C}) = (r_1, r_2, \ldots, r_T).$$

The **rate region** of a channel is the set of rate vectors that can be reached by arbitrarily small error:

$$\mathcal{R} = \big\{\mathbf{r} \colon \forall \varepsilon > 0 \colon \exists \mathcal{C} \colon P_e(\mathcal{C}) \leq \varepsilon \text{ and } \forall i \in [T] \colon r_i(\mathcal{C}) \geq r_i\big\}.$$

Ahlswede (1971) and van der Meulen (1971) have determined the rate region for the case $t = 2$. Liao (1972) has formulated the rate region for the general t user case.

In contrast to the multi-user information theory, in the models of multiple access communications the users are partially active, which is formulated such that in a given time instant at most M out of the T can be active. Moreover, usually the models allow the asynchronous access. For the current practical solutions in mobil communications the active users initiate a login procedure during which get codes for the actual session. Here we are interested in the problems, where the users have codes forever, therefore three tasks should be solved:

- detection (identification) of active users,
- synchronization of their code words, and
- decoding the messages.

There is an important special case of this general problem, where the users have no messages at all, their activity is the only "information" to transmit. It is called **signature coding**, which means just to solve the identification and synchronization.

In this note we consider just **deterministic multiple access channels**:

- OR channel,
- collision channel,
- slow frequency hopping,
- collision channel with ternary feedback,
- ADDER channel,
- collision channel with multiplicity feedback,
- Euclidean channel.

Chapter 2

OR channel: synchronous access

2.1 Channel model

Cohen, Heller and Viterbi (1971) introduced the model of the noiseless deterministic OR channel for multiple access communication. If there are T users in the system such that the inputs x_i $(1 \leq i \leq T)$ and the output y are binary, then the output is 0 iff all inputs are 0, so the output is the Boolean sum of the inputs (cf. Figure 2.1):

$$y = \bigvee_{i=1}^{T} x_i.$$

A possible example of communication scheme where this simple model is suitable is the on/off keying (OOK) modulation, where the bit 1 corresponds

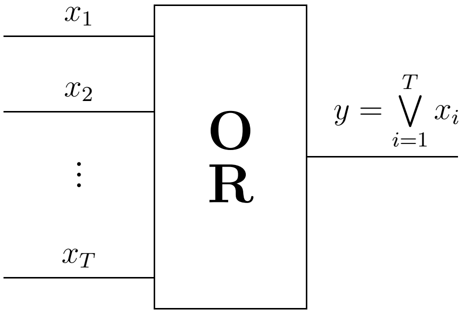

Figure 2.1: Multiple access OR channel

to a waveform and the bit 0 corresponds to the waveform constant 0. The receiver consists of an envelope detector followed by a threshold detector, so the demodulation is just a decision whether all users sent the 0 waveform.

We are investigating the identification (and in Chapter 3 also the synchronization) problem which is called *signature coding* via a multiple access OR channel. There are T users in the communication system, and each of them has only one own code word. Becoming active a user sends his code word into the channel, and otherwise does nothing, formally sends the all-zero code word into the channel. The decoder, from the Boolean sum of the code words of the active users, should reconstruct the set of active users.

For permanent activity of the users this channel is trivial, with time sharing the maximum utility 1 can be achieved. For partial activity, however, the problem is hard and is far from being solved. It is usually assumed that at most M users communicate on the channel simultaneously, where M is a fixed number. The signature coding problem is to find a code of minimum length $n(T, M)$ such that if at most M active out of T total users send their code words, then from the output vector of the OR channel the set of active users can be identified.

Kautz and Singleton (1964) introduced the concept of UD and ZFD codes.

Definition 2.1 (UD code). *A code which has T code words of length n is Uniquely Decipherable of order M (UD(T, M, n)), if every Boolean sum of up to M different code words is distinct from every other sum of M or fewer code words.*

Formally, let $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ be a code. The UD(T, M, n) property means that for any subsets $A, B \subset \{1, 2, \dots, T\}, |A| \leq M, |B| \leq M, A \neq B$ we have

$$\bigvee_{i \in A} \mathbf{x}_i \neq \bigvee_{i \in B} \mathbf{x}_i.$$

Example: Retrieval files. Assume a library of documents (files) such that each document has attributes, called descriptors. The information retrieval can be done according to a descriptor such that we have to decide whether a given document has this descriptor. This inquiry can be organized by a head of the document which is a list of descriptors. If the total number of descriptors is T and a given document may have at most M descriptors then this head can be encoded into a binary vector of length $n = \log \sum_{m=0}^{M} \binom{T}{m} \simeq M \log T$.

If a document and its descriptors are changing from time to time then the head is changing, too, therefore for generating the head we have an alternative way by UD codes. Let $\mathbf{x}_i$ be a binary vector assigned to descriptor i, and if $i_1 < \cdots < i_m$ are the descriptors then generate the head by the Boolean sum:

$$\mathbf{y} = \bigvee_{j \in \{i_1, \ldots, i_m\}} \mathbf{x}_j.$$

Thus for UD(T, M, n) code, from the head we can identify the descriptors (cf. Kautz and Singleton (1964), Chien and Frazer (1966)).

Given two binary sequence $\mathbf{x}$ and $\mathbf{y}$ of the same length n, the superposition sum of these sequences is a binary sequence $\mathbf{z}$ of length n

$$\mathbf{z} = \mathbf{x} \vee \mathbf{y},$$

where

$$z_i = \begin{cases} 0 & \text{if } x_i = y_i = 0, \\ 1 & \text{otherwise} \end{cases}$$

$(i = 1, 2, \ldots, n)$.

We say that $\mathbf{z} = (z_1, \ldots, z_n)$ covers $\mathbf{y} = (y_1, \ldots, y_n)$

$$\mathbf{z} \geq \mathbf{y}$$

if $z_i \geq y_i$ $(i = 1, 2, \ldots, n)$.

In the previous example the coding of the head was easy and fast. Maybe want an easy decoding, too. Notice that for $j \in \{i_1, i_2, \ldots, i_m\}$

$$\mathbf{y} \geq \mathbf{x}_j.$$

This remark leads to a special case of UD code:

Definition 2.2 (ZFD code). *We call a code which has T code words of length n Zero False Drop of order M (ZFD(T, M, n)), if every Boolean sum of up to M different code words logically includes no code word other than those used to form the sum.*

It means that if for a k

$$\mathbf{y} \geq \mathbf{x}_k$$

then

$$\mathbf{x}_k = \mathbf{x}_{i_j}$$

for some i_j. In the example of information retrieval it means that for a document with head $\mathbf{y}$ and for the descriptor k if

$$\mathbf{y} \geq \mathbf{x}_k$$

then the document has the descriptor k. This is really a fast decoding rule.

The ZFD property is defined by a decoding rule, therefore a ZFD(T, M, n) is a UD(T, M, n). The question is that what is the loss with respect to UD if we need ZFD.

Theorem 2.1 (Kautz and Singleton (1964)). *A UD(T, M, n) is ZFD$(T, M-1, n)$ and a ZFD(T, M, n) is UD(T, M, n).*

Proof.

1. Assume that $\mathcal{C}$ is UD(T, M, n). If $\mathcal{C}$ were not ZFD$(T, M-1, n)$ then there would be $\mathbf{x}_M \notin \{\mathbf{x}_1, \dots, \mathbf{x}_{M-1}\}$ such that
$$\mathbf{x}_M \leq \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_{M-1},$$
i.e.
$$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_{M-1} \vee \mathbf{x}_M = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_{M-1},$$
which contradicts that $\mathcal{C}$ is UD(T, M, n).

2. Suppose that $\mathcal{C}$ is ZFD(T, M, n) but not UD(T, M, n), then there exist sets of code words $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\} \neq \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$, $K, L \leq M$ such that
$$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_K = \mathbf{y}_1 \vee \mathbf{y}_2 \vee \cdots \vee \mathbf{y}_L.$$
Since the two sets are not equal there exists a code word $\mathbf{x}_i$ which is not in $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$. However,
$$\mathbf{x}_i \leq \mathbf{y}_1 \vee \mathbf{y}_2 \vee \cdots \vee \mathbf{y}_L$$
is a contradiction. □

Corollary 2.1. *The relationship between ZFD(T, M, n) and UD(T, M, n) codes is as follows.*

$$\textit{ZFD}(T, M, n) \subseteq \textit{UD}(T, M, n) \subseteq \textit{ZFD}(T, M-1, n) \subseteq \cdots$$

Put $\mathcal{C}_1 = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ and let $\mathcal{C}_k, k = 2, 3, \dots$ be the set of all superposition sum of exactly k vectors of $\mathcal{C}_1$. Thus, the set $\mathcal{C}_k$ contains $\binom{T}{k}$ vectors, which are not necessarily all different.

In considering the sequence of sets $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k, \dots$ we are interested in the value of k at which duplicate vectors first appear, either within the same set $\mathcal{C}_k$, or between $\mathcal{C}_k$ and some earlier set.

Lemma 2.1 (Kautz and Singleton (1964)). *If the sets $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{M+1}$ are disjoint then $\mathcal{C}_k$ contains exactly $\binom{T}{k}$ different vectors ($k = 1, 2, \dots, M$).*

Proof. Suppose that two of the $\binom{T}{k}$ vectors in $\mathcal{C}_k$ were equal:

$$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k = \mathbf{y}_1 \vee \mathbf{y}_2 \vee \cdots \vee \mathbf{y}_k$$

where $\mathbf{x}_i, \mathbf{y}_i \in \mathcal{C}_1, i = 1, 2, \ldots, k$. Then

$$\mathbf{y}_j \vee \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k$$

for every $j = 1, 2, \ldots, k$. But $\mathcal{C}_{k+1}$ and $\mathcal{C}_k$ are disjoint, therefore each of the code words $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ must belong to the set of code words $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$, so there are no duplicates in $\mathcal{C}_k$. □

Lemma 2.2 (Kautz and Singleton (1964)). *A code $\mathcal{C}_1$ is ZFD(T, M, n) iff the sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{M+1}$ are disjoint.*

Proof. We use an indirect way of proof in both direction of the statement.

1. Suppose that $\mathcal{C}_1$ is not ZFD, so there is a $\mathbf{y} \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}, k \leq M$ such that $\mathbf{y} \leq \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k$, i.e.

   $$\mathbf{y} \vee \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_k$$

   then $\mathcal{C}_{k+1}$ and $\mathcal{C}_k$ would not be disjoint.

2. Suppose that $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{M+1}$ are not disjoint, so there are $\mathcal{C}_j$ and $\mathcal{C}_k$ for some $1 \leq j < k \leq M + 1$ having common element

   $$\mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_j = \mathbf{y}_1 \vee \mathbf{y}_2 \vee \cdots \vee \mathbf{y}_k.$$

   Because of $j < k$ there is a $\mathbf{y}_i \notin \{\mathbf{x}_1, \ldots, \mathbf{x}_j\}$, and

   $$\mathbf{y}_i \vee \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_j = \mathbf{x}_1 \vee \mathbf{x}_2 \vee \cdots \vee \mathbf{x}_j,$$

   therefore $\mathcal{C}_1$ is not ZFD. □

Lemma 2.3 (Kautz and Singleton (1964)). *A code $\mathcal{C}_1$ is UD(T, M, n) iff the sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ are disjoint and $\mathcal{C}_M$ contains $\binom{T}{M}$ different vectors.*

Proof.

1. Suppose that the sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ are disjoint and $\mathcal{C}_M$ contains $\binom{T}{M}$ different vectors. Then because of Lemma 2.1 each set $\mathcal{C}_k$ for $1 \leq k \leq M$ contains $\binom{T}{k}$ different vectors, therefore no two superposition sum vectors of at most M code words can be equal without contradicting either the condition that $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ be disjoint, or that $\mathcal{C}_k$ contains $\binom{T}{k}$ different elements for $1 \leq k \leq M$.

2. Suppose that the code $\mathcal{C}_1$ is UD(T, M, n). Then any two superposition sum of at most M code vectors are different, therefore $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ are disjoint and $\mathcal{C}_M$ contains $\binom{T}{M}$ different elements.

□

Lemma 2.4 (Kautz and Singleton (1964)). *Let the $T \times n$ matrix $\mathbf{A}$ consist of the code vectors of $\mathcal{C}_1$. The code $\mathcal{C}_1$ is ZFD(T, M, n) iff every subset of $M+1$ rows of $\mathbf{A}$ contains an $(M+1)$-columned identity submatrix.*

Proof. The condition that $\mathcal{C}_1$ be ZFD(T, M, n) is equivalent to the requirement that in each subset of $M+1$ rows of $\mathbf{A}$, no one row may be covered by the superposition sum of the other M. This will be the case iff each row of this $(M+1)$-rowed submatrix has a 1 in some column in which all other rows have a 0. Conversely, if every subset of $(M+1)$ rows contains an identity submatrix of order $(M+1)$, then no one of these rows may be covered by the sum of the other M. Hence, $\mathcal{C}_1$ is ZFD(T, M, n). □

Example: Signature coding for multiple access OR channel. Consider a T user multiple access OR channel. Each user has an n length binary vector (code word), and if a user is active then it sends its code word. From the output of the OR channel, i.e., from the superposition sum of the active code words one has to identify the set of active users. If at most M users can be active then it is a UD(T, M, n) problem. If, moreover, we want to have an easy decoding by covering then it is a ZFD(T, M, n) problem.

Example: Monitoring. Let us assume a public transportation company which has a lot of buses. Each bus broadcasts its code word periodically. There is a receiver in a heavy-traffic junction. If there is only one bus in the range of the receiver, the problem is easy. If there are many buses, then suppose, that the modulation is OOK (on/off keying). Since in case of many simultaneous transmission the signal in the receiver can be modelled by the output of an OR channel, the received signal is the Boolean sum of those identifiers which buses are in the range of the receiver.

Example: Alarming. Let us chain as many as T fire-alarm stations to one wire. Should an alarm station become active, it sends its own code word. If the number of simultaneous outbreaks of fire is not more than M, then the active stations can be identified from the signal on the wire. Existent alarming systems usually apply a 1 bit output which only tells there is fire *somewhere* in the system. Advantage of using a multiple access channel is to be able to know which rooms or locations are catching fire at the moment and where the fire spreads.

Example: Login. Consider a communications system which has lots of low-duty mobile users, but just a limited number of channels. Becoming active, a user may send his code word over a radio channel to a central control unit, and from the output of the channel the central control unit may detect the set of active users and assign dedicated channels to them. Nowadays, mobile telecommunications systems use random access with feedback, so that users can log in to the system. This procedure can be replaced by signature coding for multiple access channel, where the advantage is that there is no need to process the acknowledgements.

Example: Collection of measurment data. We would like to collect, for example, electric energy consumption data of customers in a power line network. The power line can be used as a multiple access OR channel (cf. Dostert (2001)). The measuring instrument of a user sends its unique code word to this common channel if a user has consumed a unit (e.g., 1 kWh) of electric energy.

Example: Sending packets without error correction. Consider the collision channel without feedback (time hopping) with the restriction that there is no error correction over the packets, a user just repeats its packet several times, and needs at least one successful transmission. The sending is according to protocol sequences, user i has an n-length binary vector $\mathbf{x}_i$ which is its protocol sequence. Assume that user i is active, i.e., has a packet to send. Then it has at least one successful transmission, if $\mathbf{x}_i$ is not covered by the superposition sum of the protocol sequences of the other active users, which means that the protocol sequence set should be $\mathrm{ZFD}(T, M-1, n)$, where M is the number of active users.

Example: Non-adaptive hypergeometric group testing. The problem of group testing firstly appeared in administering syphilis tests to millions of people being inducted into the U.S. military services during World War II. The test for syphilis was a blood test called the Wasserman test. Dorfman (1943) suggested pooling the blood samples from a number of persons and applying the Wasserman test to a sample from the resultant pool. The Wasserman test had sufficient sensitivity that the test would yield a negative result if and only if none of the individual samples in the pooled sample were diseased. Dorfman's paper was the beginning of a research area which has become known as group testing. Assume T individuals which contain at most M defectives. In a test step one can ask whether a subset of T contains some defectives. The task is to identify the positive individuals using the minimum number of needed tests (in the worst case). A test plan is a sequence of tests such that, at its completion, the outcomes of these tests uniquely determine the states of all individuals. In the classical (adaptive) group testing there is a feedback, when selecting a set A in a step we know the results of the previous steps. One can see that the number of steps required is less than $M \log T$, and for binomial model of the defectives (when each individual is defective with the same probability) there are efficient strategies (cf. Hwang (1972), Wolf (1985), Sterrett (1957), Sobel and Groll (1959)). In the problem of non-adaptive group testing (cf. Hwang and T. Sós (1987), Du and Hwang (1993), Knill et al. (1998)) we don't assume feedback, choose a priori a sequence of test sets $(A_1, A_2, \ldots, A_n)$. The trivial solution corresponds to the time sharing, when $A_i = \{i\}, i = 1, \ldots, T$, so $n = T$.

The testing can be formulated in another way: the i^{th} individual has the binary code word (test sequence) $\vec{x}_i$ and

$$A_j = \{i : x_{i,j} = 1\}.$$

If the individuals $i_1, i_2, \ldots, i_m$ are positive, then the result of the test is

$$\vec{y} = \bigvee_{i \in \{i_1, \ldots, i_m\}} \vec{x}_i,$$

where $y_j = 1$ iff $\{i_1, i_2, \ldots, i_m\} \cap A_j \neq \emptyset$, and from $\vec{y}$ we should identify $i_1, \ldots, i_m$.

As we show in the sequel, the number of steps needed is at least $c \, \frac{M^2}{\log M} \, \log T$ for some constant c.

Let U be an n-element set (called underlying set). We denote by $\binom{U}{k}$ the set of the k-element subsets of U $(0 \leq k \leq n)$, while 2^U denotes the power set of U $(2^U = \bigcup_{k=0}^{n} \binom{U}{k})$. A family $\mathcal{F}$ of subsets of U is a subset of the power set $(\mathcal{F} \subseteq 2^U)$.

Definition 2.3 (Cover-free family, cf. Füredi (1996), Erdős et al. (1985)). *A family of sets $\mathcal{F}$ is called M-cover-free if*

$$F_0 \not\subseteq F_1 \cup \cdots \cup F_M$$

holds for all distinct $F_0, F_1, \ldots, F_M \in \mathcal{F}$.

We are looking for the maximum cardinality T of an M-cover-free family $\mathcal{F} \subseteq 2^U$, where $|U| = n$. This problem is analogous to the determination of minimal length of ZFD codes. Matching of parameters is the following. Put $U = \{1, 2, \ldots, n\}$, and a set $F \in \mathcal{F}$ corresponds to a binary code word $\mathbf{x}_F$ the i-th coordinate of which is 1 iff $i \in F$. Cardinality T of the family plays the role of number of potential users, M-cover-free property corresponds to the ZFD property of order M, and the size n of underlying set U corresponds to the code length.

2.2 Lower bounds

In the following we give bounds on the minimal code length $n(T, M)$.

Possibly the simplest lower bound can be computed using the fact that each sum of at most M code words must be distinct, so cannot exceed the number of n-digit binary numbers.

Sphere packing bound:

$$\sum_{k=0}^{M} \binom{T}{k} \leq 2^n.$$

Using that $\sum\limits_{k=0}^{M} \binom{T}{k} \sim T^M$, we get

$$n(T, M) \geq M \log T.$$

In the sequel we summarize the bounds $(1 \ll M \ll T)$

$$c_1 \frac{M^2}{\log M} \log T \leq n(T, M) \leq c_2 M^2 \log T.$$

The unpublished result of Bassalygo gives the first bound which uses the following lemmata. Let $t(w)$ denote the number of code words with weight w.

Lemma 2.5 (Kautz and Singleton (1964), Dyachkov and Rykov (1982)). *If any code word of a ZFD(T, M, n) code has weight no greater than M, it must have a 1 in some position where no other code word has a 1, thus*

$$\sum_{w=1}^{M} t(w) \leq n.$$

Proof. Suppose that there exists a code word of a ZFD(T, M, n) code which has weight no greater than M, but it has 1's just in positions where some of the other code words have 1, too. This code word is then covered by the sum of M other code words, so the code can not be ZFD(T, M, n). □

Lemma 2.6 (Bassalygo, cf. Dyachkov and Rykov (1982), A (1986)). *If $\mathcal{C}$ is a ZFD(T, M, n) code and it has a code word of weight w then*

$$w \leq n - n(T-1, M-1).$$

Proof. Consider $\mathcal{C}$ as a $T \times n$ binary matrix. Choose a code word with weight w. In the code matrix, delete all columns where this fixed code word has 1's, and also delete the row corresponding to this code word. The resulting matrix of size $(T-1)\times(n-w)$ can be easily verified to be a ZFD$(T-1, M-1, n-w)$ code. □

Theorem 2.2 (Bassalygo bound, cf. Dyachkov and Rykov (1982; 1983), A (1986)).

$$n(M, T) \geq \min\left\{\frac{(M+1)(M+2)}{2}, T\right\}$$

Proof. Let $\mathcal{C}$ be a ZFD(T, M) code of length n and let $w_{\max}$ be the maximum weight.

1. If $w_{\max} \leq M$ then by Lemma 2.5 we get
$$T = \sum_{w=1}^{w_{\max}} t(w) \leq n$$
2. If $w_{\max} \geq M+1$ then by Lemma 2.6 we get
$$n \geq n(T-1, M-1) + M + 1$$

Combining these two inequalities results in

$$n(T, M) \geq \min\left\{n(T-1, M-1) + M + 1, T\right\}.$$

We use induction. The statement holds for $M = T = 1$. Assume that the statement holds for sizes up to $T-1$ then

$$\begin{aligned} n(M, T) &\geq \min\left\{n(T-1, M-1) + M + 1, T\right\} \\ &\geq \min\left\{\min\left\{\frac{M(M+1)}{2}, T-1\right\} + M + 1, T\right\} \\ &= \min\left\{\frac{(M+1)(M+2)}{2}, T\right\}. \end{aligned}$$

□

The main consequence of the theorem is that for any $\sqrt{2T} < M < T$, $n(T,M) = T$, so in this range of M no ZFD(T,M) code is better than the time-sharing.

Lemma 2.7 (Dyachkov and Rykov (1982)). *In a ZFD(T,M,n) code number of code words which have weight $w \geq M+1$ is bounded*

$$t(w) \leq M^2 \frac{\binom{n}{\lceil w/M \rceil}}{\binom{\lfloor w/M \rfloor M}{\lfloor w/M \rfloor}}. \tag{2.1}$$

Proof. Consider the ZFD(T,M,n) code as a family of sets $\mathcal{F} := F_1, F_2, \ldots, F_T$ on the underlying set $\{1,2,\ldots,n\}$. We shall show that the number $t(w)$ of subsets of the family $\mathcal{F}$ which contain w elements, satisfies inequality (2.1).

Choose an arbitrary subset $F \in \mathcal{F}$ which contains w elements. Let us assume that w can be divided by M, and we set $k = \frac{w}{M}$. We call subsets $\{A_i\}_{i=1}^M, A_i \subset F$ a partition of F into M parts, if $A_i \cap A_j = \emptyset, |A_i| = k$ and $F = \bigcup\limits_{i=1}^{M} A_i$. The number of all partitions is equal to $\frac{w!}{M!(k!)^M}$. Partitions $\{A_i\}_{i=1}^M$ and $\{A'_j\}_{j=1}^M$ are different from one another iff there exists at least one pair of numbers (i,j) for which $A_i \neq A'_j$. Partitions $\{A_i\}_{i=1}^M$ and $\{A'_j\}_{j=1}^M$ are called non-intersecting if $A_i \neq A'_j$ for any $1 \leq i,j \leq M$. We would like to determine the number of all non-intersecting partitions $R(w,M,k)$. Let us fix an arbitrary partition $\{A_i\}_{i=1}^M$. The number of partitions $\{A'_j\}$ that involve the set $A_i, 1 \leq i \leq M$ is equal to $\frac{(w-k)!}{(M-1)!(k!)^{M-1}}$. Therefore the number of partitions that intersect with $\{A_i\}_{i=1}^M$, does not exceed $M\frac{(w-k)!}{(M-1)!(k!)^{M-1}}$, so

$$R(w,M,k) \cdot M \frac{(w-k)!}{(M-1)!(k!)^{M-1}} \geq \frac{w!}{M!(k!)^M},$$

and for the number of non-intersecting partitions we get

$$R(w,M,k) \geq \frac{\binom{w}{k}}{M^2}. \tag{2.2}$$

From the M-cover-free property of $\mathcal{F}$ follows that each partition $\{A_i\}_{i=1}^M$ contains at least one term $A_i, |A_i| = k$ that belongs only to F and does not belong to any other $F_i, 1 \leq i \leq T$. Therefore F contains at least $R(w,M,k)$ subsets of volume k which do not belong to the remaining terms of family $F_1, F_2, \ldots, F_T$. Consequently,

$$t(w)R(w,M,k) \leq \binom{n}{k}. \tag{2.3}$$

From inequalities (2.2) and (2.3) we get

$$t(w) \le M^2 \frac{\binom{n}{k}}{\binom{w}{k}}.$$

Now assume that w cannot be divided by M, i.e., $w = kM + r$, where $k = \lfloor w/M \rfloor, 1 \le r \le M-1$. Let us fix an arbitrary subset $A \subset F$ in which the number of elements $|A| = r$, and we consider partitions of F in the following form: $F = A \cup \bigcup_{i=1}^{M} A_i$, where an $\{A_i\}_{i=1}^{M}$ is one of the $R(kM, M, k)$ non-intersecting partitions of set $F \setminus A$. It follows from the M-cover-free property of $\mathcal{F}$ that in any partition of F there exists a term A_i $(1 \le i \le M)$ and element $\omega \in A$, such that the set $A_i' = A_i + \omega$ $(|A_i'| = k+1 = \lceil w/M \rceil)$ belongs only to F and does not belong to any other F_i, $1 \le i \le T$. Then, we have an inequality similar to (2.3)

$$t(w) R(kM, M, k) \le \binom{n}{k+1}, \quad k+1 = \lceil w/M \rceil$$

which means together with (2.2) that expression (2.1) is valid. □

From Lemma 2.5, 2.6 and 2.7 follows the next theorem.

Theorem 2.3 (Dyachkov and Rykov (1982; 1983)). *The length n of any ZFD(T, M, n) code for $2 \le M < T$ satisfies the inequality*

$$T \le n + M^2 \sum_{w=M+1}^{n-n(T-1,M-1)} \frac{\binom{n}{\lceil w/M \rceil}}{\binom{\lfloor w/M \rfloor M}{\lfloor w/M \rfloor}} \tag{2.4}$$

Theorem 2.3 implies a lower bound on the minimal code length.

Theorem 2.4 (Dyachkov and Rykov (1982; 1983)). *If $T \to \infty$ and M is constant then*

$$n(M, T) \ge K(M) \log T (1 + o(1)),$$

where the sequence $K(M)$ is defined recurrently. $K(1) := 1$ and if $M \ge 2$ then $K(M)$ can be bounded by

$$K(M) \ge \frac{M^2}{2 \log \frac{\mathrm{e}(M+1)}{2}} \qquad (M \ge 2).$$

If $M \to \infty$ then

$$K(M) = \frac{M^2}{2 \log M}(1 + o(1)) \qquad (M \to \infty).$$

Proof. Let us take inequality (2.4) of Theorem 2.3 as starting point and apply Theorem B.1.

$$\begin{aligned}
T &\leq n + M^2 \sum_{w=M+1}^{n-n(T-1,M-1)} \frac{2^{nh\left(\frac{w}{Mn}\right)} \sqrt{\frac{n}{2\pi \frac{w}{M}\left(n-\frac{w}{M}\right)}}}{2^{wh\left(\frac{1}{M}\right)} \sqrt{\frac{w}{8\frac{w}{M}\left(w-\frac{w}{M}\right)}}} \\
&= n + M^2 \sum_{w=M+1}^{n-n(T-1,M-1)} 2^{nh\left(\frac{w}{Mn}\right)-wh\left(\frac{1}{M}\right)} \sqrt{\frac{4\left(1-\frac{1}{M}\right)}{\pi\left(1-\frac{w}{Mn}\right)}} \\
&\leq n + M^2 n \, 2^{nh\left(\frac{w}{Mn}\right)-wh\left(\frac{1}{M}\right)} \sqrt{\frac{4\left(1-\frac{1}{M}\right)}{\pi\left(1-\frac{w}{Mn}\right)}}\Bigg|_{w=w_{\max}}
\end{aligned}$$

Taking the logarithm of both sides we get the following asymptotic lower bound for the code length

$$\frac{1}{h\left(\frac{w}{Mn}\right) - \frac{w}{n}h\left(\frac{1}{M}\right)\Big|_{w=w_{\max}}} \log T \lesssim n. \tag{2.5}$$

We are looking for the lower bound of the code length in the following form

$$K(M)\log T \leq n(T,M) = n,$$

so from (2.5) we know

$$\frac{1}{h\left(\frac{w}{Mn}\right) - \frac{w}{n}h\left(\frac{1}{M}\right)\Big|_{w=w_{\max}}} \leq K(M) \tag{2.6}$$

From Lemma 2.6 follows that

$$\begin{aligned}
\frac{w_{\max}}{n} &= \frac{w_{\max}}{n(T,M)} \\
&= \frac{n(T,M) - n(T-1,M-1)}{n(T,M)} \\
&\simeq 1 - \frac{n(T,M-1)}{n(T,M)} \\
&= 1 - \frac{K(M-1)}{K(M)} \qquad (2.7)
\end{aligned}$$

Substitution (2.7) to inequality (2.6) results

$$\frac{1}{h\left(\left(1-\frac{K(M-1)}{K(M)}\right)\frac{1}{M}\right) - \left(1-\frac{K(M-1)}{K(M)}\right)h\left(\frac{1}{M}\right)} \leq K(M) \tag{2.8}$$

Now our task is to determine an explicit formula for the recurrently given $K(M)$. We have $K(1)=1$. We show that

$$K(M) \simeq \frac{1}{2}\frac{M^2}{\log(M+1)} \tag{2.9}$$

is an asymptotic solution of inequality (2.8). The left side of (2.8) can be written in view of (2.9)

$$\begin{aligned}
&\frac{1}{h\left(\left(1-\frac{K(M-1)}{K(M)}\right)\frac{1}{M}\right)-\left(1-\frac{K(M-1)}{K(M)}\right)h\left(\frac{1}{M}\right)} \\
&\simeq \frac{1}{h\left(\left(1-\frac{(M-1)^2}{M^2}\right)\frac{1}{M}\right)-\left(1-\frac{(M-1)^2}{M^2}\right)h\left(\frac{1}{M}\right)} \\
&= \frac{1}{h\left(\frac{2M-1}{M^3}\right)-\frac{2M-1}{M^2}h\left(\frac{1}{M}\right)} \\
&= \frac{1}{h\left(\frac{2}{M^2}\right)-\frac{2}{M}h\left(\frac{1}{M}\right)} \qquad (2.10)
\end{aligned}$$

Let us analyze the factors of denominator asymptotically.

$$\begin{aligned}
h\left(\frac{2}{M^2}\right) &= -\frac{2}{M^2}\log\frac{2}{M^2}-\left(1-\frac{2}{M^2}\right)\log\left(1-\frac{2}{M^2}\right) \\
&\simeq -\frac{2}{M^2}\log\frac{2}{M^2}-\left(1-\frac{2}{M^2}\right)\frac{2}{M^2} \\
&= -\frac{2}{M^2}\log 2+\frac{4}{M^2}\log M-\left(1-\frac{2}{M^2}\right)\frac{2}{M^2}
\end{aligned}$$

and

$$\begin{aligned}
-\frac{2}{M}h\left(\frac{1}{M}\right) &= \frac{2}{M^2}\log\frac{1}{M}+\frac{2}{M}\left(1-\frac{1}{M}\right)\log\left(1-\frac{1}{M}\right) \\
&\simeq -\frac{2}{M^2}\log M-\frac{2}{M^2}\left(1-\frac{1}{M}\right)
\end{aligned}$$

therefore

$$\begin{aligned}
&h\left(\frac{2}{M^2}\right)-\frac{2}{M}h\left(\frac{1}{M}\right) \\
&\simeq -\frac{2}{M^2}\log 2+\frac{2}{M^2}\log M-\frac{2}{M^2}\left(1-\frac{2}{M^2}+1-\frac{1}{M}\right) \\
&= \frac{2}{M^2}\log\frac{M}{2}+\frac{2}{M^2}\left(\frac{1}{M}+\frac{2}{M^2}-2\right)
\end{aligned}$$

Applying this asymptotic approximation in (2.10) we get

$$\frac{1}{h\left(\frac{2}{M^2}\right)-\frac{2}{M}h\left(\frac{1}{M}\right)} \simeq \frac{M^2}{2}\frac{1}{\log\frac{M}{2}+\frac{1}{M}+\frac{2}{M^2}-2} \simeq \frac{1}{2}\frac{M^2}{\log M}$$

□

Similar lower bounds have been proved using the set theoretical approach.

Lemma 2.8 (Füredi (1996)). *If $\mathcal{F}$ is an M-cover-free family over an n-element underlying set U, then*

$$|\mathcal{F}| \le M + \binom{n}{\lceil (n-M)/\binom{M+1}{2}\rceil}. \tag{2.11}$$

Proof. Let us fix an integer w which $0 < w \le \frac{n}{2}$. Define $\mathcal{F}_w \subset \mathcal{F}$ as the family of members having an own w-subset, i.e.,

$$\mathcal{F}_w := \{F \in \mathcal{F} : \exists A \in \tbinom{F}{w}, A \not\subseteq F', \forall F' \in \mathcal{F}, F' \ne F\}.$$

Let

$$\mathcal{F}_0 := \{F \in \mathcal{F} : |F| < w\}.$$

We show that

$$|\mathcal{F}_0 \cup \mathcal{F}_w| \le \binom{n}{w} \tag{2.12}$$

and for $w := \lceil (n-M)/\binom{M+1}{2}\rceil$

$$|\mathcal{F} \setminus (\mathcal{F}_0 \cup \mathcal{F}_w)| \le M \tag{2.13}$$

which implies the lemma:

$$\begin{aligned} |\mathcal{F}| &= |\mathcal{F} \setminus (\mathcal{F}_0 \cup \mathcal{F}_w)| + |\mathcal{F}_0 \cup \mathcal{F}_w| \\ &\le M + \binom{n}{w} \\ &= M + \binom{n}{\lceil (n-M)/\binom{M+1}{2}\rceil}. \end{aligned}$$

Let $\mathcal{A}$ be the family of the own w-subsets,

$$\mathcal{A} := \{A \in \tbinom{U}{w} : \exists F \in \mathcal{F}, A \subseteq F, A \not\subseteq F', \forall F' \in \mathcal{F}, F' \ne F\}$$

and let $\mathcal{B}$ be the family of w-sets containing a member of $\mathcal{F}_0$, i.e.,

$$\mathcal{B} := \{B \in \tbinom{U}{w}, \exists F \in \mathcal{F}_0, F \subset B\}.$$

In order to prove inequality (2.12) we need to show following inequalities.

$$|\mathcal{F}_0| \leq |\mathcal{B}|$$

and

$$|\mathcal{F}_w| \leq |\mathcal{A}|.$$

For the first inequality observe that as $\mathcal{F}$ is an M-cover-free family, it has 1-cover-free (antichain) property too. $\mathcal{F}_0 \subseteq \mathcal{F}$, that is why the same is true for $\mathcal{F}_0$. Suppose a maximal length chain of U $(\mathcal{C}_1 \subset \mathcal{C}_2 \subset \cdots \subset \mathcal{C}_n, \mathcal{C}_i \subseteq U, |\mathcal{C}_i| = i)$. The 1-cover-free property of $\mathcal{F}_0$ implies that at most one element of a maximal length chain can be a member of $\mathcal{F}_0$, and then the w-element set of this chain $(\mathcal{C}_w)$ is a member of $\mathcal{B}$ (because $\mathcal{B}$ contains all the w-sets which contains an element of $\mathcal{F}_0$). The second inequality follows from the definition of own subsets, namely own subsets are different for every member of $\mathcal{F}_w$.

It is also true that $\mathcal{A}$ and $\mathcal{B}$ are disjoint. As an indirect way, suppose that there is a w-element set F^* which is the member of both families. $F^* \in \mathcal{A}$ implies that there is $F \in \mathcal{F}$ such that $F^* \subseteq F$ and for all $F' \in \mathcal{F}, F' \neq F$ we have $F^* \not\subseteq F'$. $F^* \in \mathcal{B}$ implies that there is $F_0 \in \mathcal{F}_0$ such that $F_0 \subset F^*$. Combining these relations $F_0 \subset F^* \subseteq F$ follows, which is a contradiction to the M-cover-free (even to antichain) property.

From the previous considerations the following inequality can be derived:

$$|\mathcal{F}_0 \cup \mathcal{F}_w| = |\mathcal{F}_0| + |\mathcal{F}_w| \leq |\mathcal{B}| + |\mathcal{A}| \leq \binom{n}{w},$$

so (2.12) is proved.

Let $\mathcal{F}' := \mathcal{F} \setminus (\mathcal{F}_0 \cup \mathcal{F}_w)$. The members of $\mathcal{F}'$ are at least w-element sets having no own w-subset, then $F' \in \mathcal{F}', F_1, F_2, \ldots, F_i \in \mathcal{F}$ $(F_j \neq F', 1 \leq j \leq i)$ imply

$$\left| F' \setminus \bigcup_{j=1}^{i} F_j \right| > w(M-i). \tag{2.14}$$

In order to see this, suppose that $F' \setminus \bigcup\limits_{j=1}^{i} F_j = \bigcup\limits_{j=i+1}^{M} A_j$, where $|A_j| = w$ (sets $\{A_j\}_{j=i+1}^{M}$ are not necessarily disjoint). As $F' \in \mathcal{F}'$, for every $A_j \subseteq F'$ there exists $F_j \in \mathcal{F} : F_j \neq F', A_j \subseteq F_j$, so

$$F' \setminus \bigcup_{j=1}^{i} F_j = \bigcup_{j=i+1}^{M} A_j \subseteq \bigcup_{j=i+1}^{M} F_j,$$

and from this $F' \subseteq \bigcup\limits_{i=1}^{M} F_i$ follows which is a contradiction to the M-cover-free property.

So that to prove (2.13), again, use an indirect way of proof. Assume that $|\mathcal{F}'| > M$. For $F'_1, F'_2, \ldots, F'_{M+1} \in \mathcal{F}'$ inequality (2.14) implies that

$$\begin{aligned}
&\left|\bigcup_{i=1}^{M+1} F'_i\right| \\
&\quad = |F'_1| + |F'_2 \setminus F'_1| + |F'_3 \setminus (F'_2 \cup F'_1)| + \cdots + |F'_{M+1} \setminus (F'_M \cup \cdots \cup F'_1)| \\
&\quad \geq M + 1 + w\binom{M+1}{2}.
\end{aligned}$$

The right hand side of this inequality exceeds n for $w := \left\lceil (n-M)/\binom{M+1}{2} \right\rceil$, which is a contradiction, implying (2.13). □

Theorem 2.5 (Füredi (1996)). *For* $1 \ll M \ll T$

$$n(M,T) \geq \frac{1}{4}\,\frac{M^2}{\log M}\,(1+o(1))\,\log T$$

1st Proof. We get the upper bound from inequality (2.11) using $\binom{n}{k} \leq \left(\frac{\mathrm{e}n}{k}\right)^k$:

$$\begin{aligned}
T = |\mathcal{F}| &\leq M + \binom{n}{\left\lceil (n-M)/\binom{M+1}{2} \right\rceil} \\
&\leq M + \left(\frac{\mathrm{e}n}{\left\lceil (n-M)/\binom{M+1}{2} \right\rceil} \right)^{\left\lceil (n-M)/\binom{M+1}{2} \right\rceil} \\
&\leq M + \left(\frac{\mathrm{e}M(M+1)}{2} \right)^{2n/M^2}
\end{aligned}$$

Taking the logarithm of both side we get asymptotically in M:

$$\log T \leq 4\frac{n}{M^2}\log M,$$

which corresponds to the statement of theorem. □

2nd Proof. We get the upper bound from inequalit (2.11) and Theorem B.1.

$$\begin{aligned}
T = |\mathcal{F}| &\leq M + \binom{n}{\left\lceil (n-M)/\binom{M+1}{2} \right\rceil} \\
&\leq M + 2^{nh\left(\left\lceil \frac{n-M}{\binom{M+1}{2}} \right\rceil \frac{1}{n}\right)} \\
&\leq M + 2^{nh\left(\frac{1}{\binom{M+1}{2}}\right)} \\
&\leq M + 2^{nh\left(\frac{2}{M^2}\right)}. \qquad (2.15)
\end{aligned}$$

For binary entropy we get the following upper bound,

$$\begin{aligned} h\left(\frac{2}{M^2}\right) &\le \frac{2}{M^2}\log\frac{M^2}{2} + \left(1-\frac{2}{M^2}\right)\log\left(\frac{M^2}{M^2-2}\right) \\ &\le 4\frac{\log M}{M^2} - \frac{4}{M^2} + \frac{1}{M^2}\log\left(1+\frac{2}{M^2-2}\right)^{M^2-2} \\ &\le 4\frac{\log M}{M^2} - \frac{4}{M^2} + \frac{2\log \mathrm{e}}{M^2}. \qquad (2.16) \end{aligned}$$

From inequalities (2.15) and (2.16) we get upper bound for T asymptotically in M,

$$\log T \le 4\frac{\log M}{M^2}n$$

□

Lemma 2.9 (Erdős et al. (1985)). *If $\mathcal{F}_w \subset \binom{U}{w}$ is a maximal M-cover-free family over an n-element underlying set U, then*

$$|\mathcal{F}_w| \le \frac{\binom{n}{k}}{\binom{w-1}{k-1}},$$

where we set $k := \lceil w/M \rceil$.

For the proof of Lemma 2.9 we need three lemmata which follow.

Let us define the family $\mathcal{N}(F)$ of non-own-subsets of F of size k, i.e.,

$$\mathcal{N}(F) := \{A \in \tbinom{F}{k} : \exists F' \in \mathcal{F}_w, F' \ne F, A \subset F'\}$$

Lemma 2.10 (Erdős et al. (1985)). *If $\mathcal{F}_w$ is an M-cover-free family, $F \in \mathcal{F}_w$ and $A_1, A_2, \ldots, A_M \in \mathcal{N}(F)$ then*

$$\left|\bigcup_{i=1}^{M} A_i\right| < w.$$

Proof. From the definition of $\mathcal{N}(F)$ follows that for each $A_i \in \mathcal{N}(F)$ there exists an $F_i \in \mathcal{F}_w, F_i \ne F$ for which $A_i \subset F_i$, so

$$A_1 \cup A_2 \cup \cdots \cup A_M \subseteq F_1 \cup F_2 \cup \cdots \cup F_M.$$

As each $A_i \subset F$,

$$A_1 \cup A_2 \cup \cdots \cup A_M \subseteq F,$$

too. If equality were true in the previous statement, F would be covered by the union of $F_1, \ldots, F_M$ which is a contradiction. For this reason the following inequality has to be satisfied,

$$\left| \bigcup_{i=1}^{M} A_i \right| < |F| = w.$$

□

Lemma 2.11 (Frankl (1976)). *Let F be a finite set having w elements. Let $\mathcal{N}(F) \subset \binom{F}{k}$ be such that for $A_1, \ldots, A_M \in \mathcal{N}(F)$ we have that $\bigcup_{j=1}^{M} A_j \neq F$. If $k \geq \frac{w}{M}$, then*

$$|\mathcal{N}(F)| \leq \binom{w-1}{k}.$$

Proof. Let $x_1, x_2, \ldots, x_w, x_1$ be a cyclic ordering of the elements of F. We shall estimate the number of sets in $\mathcal{N}(F)$ consisting of k consecutive elements relative to this ordering. If there exists at least one such set, then we may suppose that x_w is the last element of either. (Last element means that its neighbor to the right is not contained in the set.) To all set in $\mathcal{N}(F)$ consisting of consecutive elements relative to this ordering we associate the index of its last element but to the set ending with x_w we associate all integers from interval $[w, Mk]$. If there are i sets consisting of consecutive elements relative to the ordering, then we have associated with them $Mk - w + 1 + (i - 1) = Mk - w + i$ indices from the interval $[1, Mk]$. Let us divide the elements of this interval into residue classes modulo k. Each class contains M elements. If we could pick out M sets from $\mathcal{N}(F)$ consisting of consecutive elements relative to the cyclic ordering such that the integers associated with them completely cover one of the classes, then the union of these sets were F, because for every x_j the smallest element in the class greater than j (in the cyclic sense) would be associated with a set of k consecutive elements which cover x_j. This would be a contradiction to the property that $\bigcup_{j=1}^{M} A_j \neq F$ for $A_1, \ldots, A_M \in \mathcal{N}(F)$. Hence, there exists an element in each of the classes to which we have not associated any of the sets. As we have associated with different sets different indices, we get:

$$Mk - w + i \leq Mk - k$$

what is equivalent,

$$i \leq w - k.$$

There are $(w-1)!$ possible cyclic ordering and each has at most $w-k$ sets consisting of consecutive elements relative to the cyclic ordering, and we count $(w-k)!\,k!$ times each set of $\mathcal{N}(F)$. Then we get the following upper bound to the cardinality of $\mathcal{N}(F)$:

$$|\mathcal{N}(F)| \le \frac{(w-1)!\,(w-k)}{(w-k)!\,k!} = \binom{w-1}{w-k-1} = \binom{w-1}{k}$$

□

Lemma 2.12 (Erdős et al. (1985)). *If $\mathcal{F}_w$ is an M-cover-free family, $F \in \mathcal{F}_w$ and $k = \lceil w/M \rceil$ then*

$$|\mathcal{N}(F)| \le \binom{w-1}{k}$$

Proof. In view of Lemma 2.10 $\mathcal{N}(F) \subset \binom{F}{k}$ satisfies that $\bigcup_{j=1}^{M} A_j \neq F$ for $A_1, \dots, A_M \in \mathcal{N}(F)$, and $|F| = w \le Mk$. Thus by Lemma 2.11

$$|\mathcal{N}(F)| \le \binom{w-1}{k}.$$

□

Proof of Lemma 2.9. Each $F \in \mathcal{F}_w$ has $\binom{w}{k}$ k-subsets, and Lemma 2.12 implies that there are at most $\binom{w-1}{k}$ non-own subsets from this. That is why each $F \in \mathcal{F}_w$ has at least

$$\begin{aligned}\binom{w}{k} - \binom{w-1}{k} &= \frac{w!}{k!\,(w-k)!} - \frac{(w-1)!}{k!\,(w-k-1)!} \\ &= \frac{(w-1)!\,(w-(w-k))}{k!\,(w-k)!} \\ &= \frac{(w-1)!}{(k-1)!\,(w-k)!} \\ &= \binom{w-1}{k-1}\end{aligned}$$

own subsets. There are $\binom{n}{k}$ possible k-elements subsets, consequently,

$$|\mathcal{F}_w| \binom{w-1}{k-1} \le \binom{n}{k}$$

holds, yielding the desired upper bound. □

Lemma 2.13 (Ruszinkó (1994)). *If $\mathcal{F} := \{F_1, F_2, \dots, F_T\}$ is an M-cover-free family, $F_i \in \mathcal{F}$ is an arbitrary element and $A_i \subseteq F_i$ is an arbitrary subset of F_i, then we can construct a new family $\mathcal{F}' := \{F_j \setminus A_i\}_{j=1,\dots,T}^{j \neq i}$ for which*

1. *$\mathcal{F}'$ is $(M-1)$-cover-free,*

2. *$|\mathcal{F}'| = T - 1$.*

Proof.

1. As an indirect way of proof, suppose that

   $$F_{j_0} \setminus A_i \subseteq (F_{j_1} \setminus A_i) \cup (F_{j_2} \setminus A_i) \cup \dots \cup (F_{j_{M-1}} \setminus A_i)$$

   for some $\{j_0, j_1, \dots, j_{M-1}\} \subseteq \{1, \dots, i-1, i+1, \dots T\}$. Then

   $$F_{j_0} \subseteq F_{j_1} \cup F_{j_2} \cup \dots \cup F_{j_{M-1}} \cup F_i,$$

   which is a contradiction ($\mathcal{F}$ is M-cover-free).

2. From the M-cover-free property of $\mathcal{F}$ it follows that $\mathcal{F}$ is 1-cover-free, too. That is why $F_j \not\subseteq F_i$ for any $i \neq j$, so we left out only F_i from $\mathcal{F}$ during the construction of $\mathcal{F}'$. Members of $\mathcal{F}'$ are distinct. As an indirect way, suppose that $F_k \setminus A_i = F_l \setminus A_i$ for some $k \neq l$. Then $F_k \subseteq F_l \cup F_i$ which is a contradiction ($\mathcal{F}$ is 2-cover-free, as $M \geq 2$).

□

Theorem 2.6 (Ruszinkó (1994)). *For $1 \ll M \ll T$*

$$n(M, T) \geq \frac{1}{8} \frac{M^2}{\log M} (1 + o(1)) \log T$$

Proof. During the proof we suppose that M^2 divides n and $\frac{n}{M}$ is even. If it is not true, then the same proof works, but we have to be more careful with the integer parts.

Let $\mathcal{F}$ be an M-cover-free family. We use the set compression algorithm of Lemma 2.13.

1. $\mathcal{F}^0 := \mathcal{F}$

2. If every element of $\mathcal{F}^i$ is of size $\leq \frac{2n}{M}$, then stop. If $\mathcal{F}^i = \{F_1^i, F_2^i, \dots, F_{T-i}^i\}$ contains a set $F_{j_0}^i$ of size $> \frac{2n}{M}$, then $\mathcal{F}^{i+1} := \{F_j^i \setminus F_{j_0}^i\}_{j=1,\dots,T-i}^{j \neq j_0}$.

In each step of this algorithm we throw out more than $\frac{2n}{M}$ elements. Since the members of $\mathcal{F}$ have not more than n elements (the underlying set is of size n), the algorithm will stop in at most $\frac{n}{2n/M} = \frac{M}{2}$ steps. Suppose that during this algorithm we threw out p elements from the underlying set in q steps. Let $T(n, M, w)$ denote the maximum cardinality of an M-cover-free family which subsets have w elements (out of n). We know from Lemma 2.9 that

$$T(n, M, w) \leq \frac{\binom{n}{\lceil w/M \rceil}}{\binom{w-1}{\lceil w/M \rceil - 1}} \simeq M \frac{\binom{n}{w/M}}{\binom{w}{w/M}}$$

Using this bound and Lemma 2.13 it follows that

$$\begin{aligned}
T = |\mathcal{F}| &\leq T\left(n - p, M - q, \leq \frac{2n}{M}\right) + q \\
&\leq T\left(n, \frac{M}{2}, \leq \frac{2n}{M}\right) + \frac{M}{2} \\
&\leq \sum_{w=1}^{2n/M} T\left(n, \frac{M}{2}, w\right) + \frac{M}{2} \\
&\leq \sum_{w=1}^{2n/M} \frac{\frac{M}{2}\binom{n}{\frac{2w}{M}}}{\binom{w}{\frac{2w}{M}}} + \frac{M}{2} \\
&\leq \sum_{w=1}^{2n/M} \frac{M}{2}\binom{n}{\frac{2w}{M}} \\
&\leq n\binom{n}{\frac{4n}{M^2}}
\end{aligned}$$

Taking the logarithm of both side and applying Theorem B.1 we get asymptotically

$$\begin{aligned}
\log T &\leq o(n) + n\, h\left(\frac{4}{M^2}\right) \\
&= o(n) + n\left(\tfrac{4}{M^2}\log\tfrac{M^2}{4} + \left(1 - \tfrac{4}{M^2}\right)\log\left(\tfrac{M^2}{M^2-4}\right)\right) \\
&= o(n) + n\left(8\tfrac{\log M}{M^2} - \tfrac{8}{M^2} + \tfrac{1}{M^2}\log\left(1 + \tfrac{4}{M^2-4}\right)^{M^2-4}\right) \\
&\leq o(n) + n\left(8\tfrac{\log M}{M^2} - \tfrac{8}{M^2} + \tfrac{4\log e}{M^2}\right)
\end{aligned}$$

$$\leq \; o(n) + 8\frac{\log M}{M^2}\,n$$

which implies that

$$n(M,T) \geq \frac{1}{8}\,\frac{M^2}{\log M}\,(1+o(1))\,\log T$$

□

2.3 Upper bounds

A and Zeisel (1988) gave upper bound of the minimal code length. They used random code with alphabet $1, 2, \ldots, L$, and a mapping from this alphabet to binary one-weight vectors similarly to the Kautz–Singleton construction:

$$\begin{array}{rcl} 1 & \mapsto & 0\ldots001 \\ 2 & \mapsto & 0\ldots010 \\ & \vdots & \\ L & \mapsto & 1\ldots000 \end{array}$$

where each pattern has length L.

Theorem 2.7 (A and Zeisel (1988)). *If $T \to \infty$ and M is fixed*

$$n(T,M) \leq K(M)M^2(1+o(1))\log T$$

where

$$K(M) \leq 1.5112.$$

If, in addition, $M \to \infty$, too:

$$\limsup_{M\to\infty} K(M) = \frac{1}{\ln 2} \approx 1.4427.$$

Proof. Assume a random code with alphabet $1, 2, \ldots, L$ ($L \geq M$) and length $\frac{n}{L}$ whose characters are independent and uniformly distributed. This code is mapped to a binary code $\mathcal{C}$ with length n by the previously described transformation.

$$\begin{aligned} \mathbf{P}\{\mathcal{C} \text{ is not ZFD}\} \;&\leq\; \binom{T}{M}(T-M)\left(1-\left(1-\frac{1}{L}\right)^M\right)^{\frac{n}{L}} \\ &\leq\; \exp\left((M+1)\ln T + \frac{n}{L}\ln\left(1-\left(1-\frac{1}{L}\right)^M\right)\right) \qquad (2.17) \end{aligned}$$

If this probability is less than one then there exists a ZFD code of order M, so the argument of the exponential function shall be below zero:

$$(M+1)\ln T + \frac{n}{L}\ln\left(1-\left(1-\frac{1}{L}\right)^M\right) < 0.$$

Expressing n from this inequality we get:

$$n(T,M) \leq \frac{(M+1)L\log T}{-\log\left(1-\left(1-\frac{1}{L}\right)^M\right)} < n_{\text{random coding}}.$$

Asymptotically we get the following upper bound:

$$n(T,M) \leq K(M)M^2(1+o(1))\log T$$

where

$$K(M) = \min_{M\leq L} \ln 2 \frac{1}{-\frac{M+1}{L}\ln\left(1-\left(1-\frac{1}{L}\right)^M\right)}$$

Let us choose $L = \left\lfloor \frac{M+1}{\ln 2} \right\rfloor$ and use the inequality $\left(1-\frac{1}{L}\right)^M \geq \exp\left(-\frac{M+1}{L}\right)$ if $M \leq L$, we get

$$K(M) \leq \frac{\ln 2}{-\alpha\ln(1-\mathrm{e}^{-\alpha})} \leq \frac{\ln 2}{\ln\frac{e}{e-1}} \approx 1.5112$$

where $\alpha = \frac{M+1}{\left\lfloor\frac{M+1}{\ln 2}\right\rfloor}$.

If $M \to \infty$, then $\alpha \to \ln 2$, so

$$\limsup_{M\to\infty} K(M) = \frac{\ln 2}{-\ln 2\ln(1-\mathrm{e}^{-\ln 2})} = \frac{1}{\ln 2} \approx 1.4427,$$

and the proof is complete. □

We can get another upper bound on the code length, if we consider a random code such that the 1's in a code word are binomially distributed (instead of the constant weight case of Theorem 2.7).

Theorem 2.8 (Dyachkov and Rykov (1983)). *If* $1 \ll M \ll T$ *and* $T \to \infty$

$$n(T,M) \leq \mathrm{e}\ln 2\, M(M+1)\log T \approx 1.884\, M^2(1+o(1))\log T$$

Proof. Consider a binary random code $\mathcal{C}$ with length n. In a code word a bit is 1 with probability p and 0 with probability $1-p$, so the number of 1's in a code word has binomial distribution.

$$\begin{aligned}
&\mathbf{P}\{\mathcal{C} \text{ is not ZFD}\} \\
&\quad = \sum_{k=0}^{n} \mathbf{P}\{\text{all 1's covered} \mid \#\text{1's} = k\}\mathbf{P}\{\#\text{1's} = k\} \\
&\quad \le \sum_{k=0}^{n} \binom{T}{M}(T-M)\left(1-(1-p)^M\right)^k \binom{n}{k} p^k (1-p)^{n-k} \\
&\quad = \binom{T}{M}(T-M)\sum_{k=0}^{N} \binom{n}{k} \left(p\left(1-(1-p)^M\right)\right)^k (1-p)^{n-k} \\
&\quad = \binom{T}{M}(T-M)\left(p\left(1-(1-p)^M\right)+1-p\right)^n \\
&\quad = \binom{T}{M}(T-M)\left(1-p(1-p)^M\right)^n
\end{aligned}$$

This expression takes its minimum value in $p = \frac{1}{M+1}$, and here

$$\begin{aligned}
\mathbf{P}\{\mathcal{C} \text{ is not ZFD}\} &\le \binom{T}{M}(T-M)\left(1-\frac{1}{M+1}\left(1-\frac{1}{M+1}\right)^M\right)^n \\
&\le \binom{T}{M}(T-M)\left(1-\frac{\mathrm{e}^{-1}}{M+1}\right)^n \\
&\approx \binom{T}{M}(T-M)\mathrm{e}^{-\frac{n}{M+1}\mathrm{e}^{-1}} \\
&\le T^M \mathrm{e}^{-\frac{n}{M+1}\mathrm{e}^{-1}} \\
&= \mathrm{e}^{M\ln 2 \log T - \frac{n}{M+1}\mathrm{e}^{-1}} \\
&< 1
\end{aligned}$$

Taking the logarithm of both side, we get

$$M \ln 2 \log T < \frac{n}{M+1}\mathrm{e}^{-1}$$

and from this

$$n(T,M) \le \mathrm{e} \ln 2\, M(M+1) \log T < n_{\text{random coding}}$$

□

Definition 2.4 (Packing, cf. Erdős et al. (1985)). *A family of sets $\mathcal{P} \subset \binom{U}{w}$ is called a (k, w, n)-packing if*

$$|P \cap P'| < k$$

holds for every pair $P, P' \in \mathcal{P}$.

Lemma 2.14 (Erdős et al. (1985)). *If $\mathcal{F}_w \subset \binom{U}{w}$ is a maximal M-cover-free family over an n-element underlying set U, then*

$$\frac{\binom{n}{k}}{\binom{w}{k}^2} \leq |\mathcal{F}_w|,$$

where we set $k := \lceil w/M \rceil$.

Proof. In (1985) Rödl have shown that there exists a (k, w, n)-packing for fixed k and w, whenever $n \to \infty$. If we set $w = M(k-1) + 1$ then a $(k, M(k-1) + 1, n)$-packing $\mathcal{P}$ is M-cover-free, because $|P \cap P'| \leq k - 1$ holds for all $P, P' \in \mathcal{P}$, so the union of M sets can cover at most $M(k-1)$ elements of the $w = M(k-1) + 1$ elements of a distinct set.

If $\mathcal{F}_w$ is a maximal (k, w, n)-packing then for every $G \in \binom{U}{w}$ there is an $F \in \mathcal{F}_w$ such that $|G \cap F| \geq k$ holds (otherwise $\mathcal{F}_w \cup G$ would also be a (k, w, n)-packing, that is why $\mathcal{F}_w$ would not be maximal). Hence we have

$$\binom{n}{w} \leq \sum_{F \in \mathcal{F}_w} \left| \left\{ G \in \binom{U}{w} : |G \cap F| \geq k \right\} \right| \leq |\mathcal{F}_w| \binom{w}{k} \binom{n-k}{w-k}. \quad (2.18)$$

Left-hand side inequality of (2.18) follows from the previous property that for every $G \in \binom{U}{w}$ there is at least one $F \in \mathcal{F}_w$ such that $|G \cap F| \geq k$ holds.

We get right-hand side inequality of (2.18) if we consider the following. For an arbitrary $F \in \mathcal{F}_w$ there are at most $\binom{w}{k}\binom{n-k}{w-k}$ sets $G \in \binom{U}{w}$ with property $|G \cap F| \geq k$, because we can choose k elements from the w elements of F and $w - k$ from the other $n - k$ elements of U.

Using

$$\begin{aligned} \binom{n}{w}\binom{w}{k} &= \frac{n!\, w!}{w!\, (n-w)!\, k!\, (w-k)!} \\ &= \frac{n!\, (n-k)!}{k!\, (n-k)!\, (w-k)!\, (n-k-(w-k))!} \\ &= \binom{n}{k}\binom{n-k}{w-k} \end{aligned}$$

this yields the lower bound of $|\mathcal{F}_w|$,

$$\begin{aligned}
\binom{n}{w} &\leq |\mathcal{F}_w|\binom{w}{k}\binom{n-k}{w-k} \\
\binom{n}{w}\binom{w}{k} &\leq |\mathcal{F}_w|\binom{w}{k}^2\binom{n-k}{w-k} \\
\binom{n}{k}\binom{n-k}{w-k} &\leq |\mathcal{F}_w|\binom{w}{k}^2\binom{n-k}{w-k} \\
\frac{\binom{n}{k}}{\binom{w}{k}^2} &\leq |\mathcal{F}_w|
\end{aligned}$$

□

Theorem 2.9 (Erdős et al. (1985)). *If* $1 \ll M \ll T$ *then*

$$n(M,T) \leq 5.122\, M^2\,(1+o(1))\,\log T.$$

Proof. The upper bound is obtained from inequality of Lemma 2.14 with setting $w := \alpha\frac{n}{M}$. We are looking for the optimal value of constant α. Let us observe that maximum cardinality of an M-cover-free family is at least the maximum cardinality of a w-uniform M-cover-free family ($|\mathcal{F}_w| \leq |\mathcal{F}|$). Using Theorem B.1 we can write the following.

$$\begin{aligned}
|\mathcal{F}| &\geq \frac{\binom{n}{\alpha\frac{n}{M^2}}}{\binom{\alpha\frac{n}{M}}{\alpha\frac{n}{M^2}}^2} \\
&\simeq \frac{2^{n\,h\left(\frac{\alpha}{M^2}\right)}}{\left(2^{\alpha\frac{n}{M}\,h\left(\frac{1}{M}\right)}\right)^2} \\
&= 2^{n\left(h\left(\frac{\alpha}{M^2}\right)-2\frac{\alpha}{M}\,h\left(\frac{1}{M}\right)\right)}
\end{aligned}$$

Let us simplify the exponent asymptotically in M,

$$\begin{aligned}
&h\left(\tfrac{\alpha}{M^2}\right) - 2\tfrac{\alpha}{M}\,h\left(\tfrac{1}{M}\right) \\
&\quad= -\tfrac{\alpha}{M^2}\log\tfrac{\alpha}{M^2} - \left(1-\tfrac{\alpha}{M^2}\right)\log\left(1-\tfrac{\alpha}{M^2}\right) \\
&\qquad+ 2\tfrac{\alpha}{M^2}\log\tfrac{1}{M} + 2\tfrac{\alpha}{M}\left(1-\tfrac{1}{M}\right)\log\left(1-\tfrac{1}{M}\right)
\end{aligned}$$

$$\begin{aligned}&= \tfrac{\alpha}{M^2}\log\tfrac{M^2}{\alpha M^2} + \tfrac{1}{M^2}\log\left(1+\tfrac{\alpha}{M^2-\alpha}\right)^{M^2-\alpha} - 2\tfrac{\alpha}{M^2}\log\left(1+\tfrac{1}{M-1}\right)^{M-1}\\ &\simeq \tfrac{\alpha}{M^2}\log\tfrac{1}{\alpha} + \left(\tfrac{\alpha}{M^2} - 2\tfrac{\alpha}{M^2}\right)\log \mathrm{e}\\ &= \tfrac{\alpha}{M^2}\log\tfrac{1}{\alpha} - \tfrac{\alpha}{M^2}\log \mathrm{e}.\end{aligned}$$

Let us calculate minimum value of this expression by differentiating in α:

$$\tfrac{1}{M^2}\log\tfrac{1}{\alpha} + \tfrac{\alpha}{M^2}\,\alpha\,\left(-\tfrac{1}{\alpha^2}\right)\log \mathrm{e} - \tfrac{1}{M^2}\log \mathrm{e} = \tfrac{1}{M^2}\log\tfrac{1}{\alpha} - \tfrac{2}{M^2}\log \mathrm{e} = 0$$

Solution of equation is $\alpha = \mathrm{e}^{-2} \approx 0.135$, from this we get

$$n(M,T) \le \left.\frac{M^2}{\frac{\alpha}{\log\frac{1}{\alpha}} - \alpha\log \mathrm{e}}\log T\right|_{\alpha=\mathrm{e}^{-2}} = 5.122\, M^2 \log T$$

□

2.4 Code constructions

Kautz and Singleton (1964) presented a construction of constant weight codes which can be used as ZFD codes.

We call maximum overlap or cross-correlation of a code the maximum number of positions in which two arbitrary code words both can have 1's.

Lemma 2.15 (Kautz and Singleton (1964)). *Let w_{min} be the minimum weight of code words in $\mathcal{C}$. If maximum overlap (cross-correlation) between code words is c then $\mathcal{C}$ is a ZFD code whose order is at least M_0 given by the following inequality:*

$$M \ge \left\lfloor \frac{w_{min}-1}{c} \right\rfloor = M_0.$$

If every c-tuple appears in two or more code words of $\mathcal{C}$ then the order of the code is exactly M_0 (and no greater).

Proof. $\mathcal{C}$ is a ZFD code of order M_0, because weight of all code word is at least $w_{\min} \ge M_0c+1$, so no code word can be covered by the superposition sum of any M_0 other codewords since it overlaps each of these other code words in no more then c positions.

If every c-tuple appears in two or more code words then for any code word whose weight is at most $(M_0+1)c$ there can be found M_0+1 other code words whose sum covers it. Thus, $\mathcal{C}$ can not be ZFD code of order M_0+1. □

Theorem 2.10 (Kautz and Singleton (1964), cf. Erdős et al. (1985)). *Let $\mathcal{C}_Q$ be a code over GF(Q) (Q is prime power) with parameters (n_Q, k) and code distance d_Q. Replace each Q-ary symbol by Q-length one-weight binary patterns. E.g., the mapping is the following:*

$$\begin{array}{ccc} 0 & \mapsto & 0\dots001 \\ 1 & \mapsto & 0\dots010 \\ & \vdots & \\ Q-1 & \mapsto & 1\dots000 \end{array}$$

where each pattern has length Q. The resulting code has

$$T = Q^k$$

code words and length

$$n = Qn_Q.$$

The maximum order of the concatenated superimposed code $\mathcal{C}$ is

$$M \geq \left\lfloor \frac{n_Q - 1}{n_Q - d_Q} \right\rfloor.$$

Proof. Obviously, $T = |\mathcal{C}| = |\mathcal{C}_Q| = Q^k,\ n = Qn_Q$. The minimum Hamming distance of binary code is twice the Q-ary distance: $d = 2d_Q$, and each code word has weight $w = n_Q$. For binary code the maximum overlap is

$$c = w - \frac{d}{2} = n_Q - d_Q.$$

□

Let us consider some special cases of Kautz–Singleton code construction.

Reed–Solomon code. (cf. A et al. (1992), Erdős et al. (1985), Zinoviev (1983)) Let $\mathcal{C}_Q$ be a Reed–Solomon code with maximal length $n_Q = Q - 1$. Resulting code $\mathcal{C}$ has

$$T = Q^k$$

code words, each has weight $w = n_Q = Q - 1$. Since Reed–Solomon code has MDS property, $d_Q = n_Q - k + 1$, and from this

$$c = n_Q - (n_Q - k + 1) = k - 1.$$

In an MDS code any k symbol may be taken as message symbols, thus each $c = k-1$-tuple is repeated exactly Q times in the binary code. So, Lemma 2.15 shows that the order of the ZFD code is exactly

$$M = \left\lfloor \frac{w-1}{c} \right\rfloor = \left\lfloor \frac{Q-2}{k-1} \right\rfloor .$$

By using $k = \frac{\log T}{\log Q}$ we get

$$\frac{\log T}{\log Q} \leq \frac{Q-2}{M} + 1. \tag{2.19}$$

If T and M are given and we would like to construct a Kautz–Singleton code with minimal code length $n = Qn_Q = Q(Q-1)$ then we have to find the minimal (prime power) Q satisfying this inequality.

Berlekamp–Justesen code. (cf. Berlekamp and Justesen (1974), Rocha (1984), A et al. (1992)) Berlekamp and Justesen have given constructions of MDS codes over $GF(Q)$ of length $n_Q = Q+1$. Using such a code we can reach better code parameters than using of Reed–Solomon code. Resulting code $\mathcal{C}$ has

$$T = |\mathcal{C}_Q| = Q^k.$$

code words, each has weight $w = n_Q = Q+1$, length $n = Qn_Q = Q(Q+1)$ and $c = k-1$. For the order of code $\mathcal{C}$ we get

$$M = \left\lfloor \frac{w-1}{c} \right\rfloor = \left\lfloor \frac{Q}{k-1} \right\rfloor ,$$

and inequality (2.19) is altered to

$$\frac{\log T}{\log Q} \leq \frac{Q}{M} + 1.$$

BCH code. (cf. Györfi and Vajda (1993)) Let $\mathcal{C}_Q$ be a BCH code with maximal length $n_Q = Q^r - 1$ for some $r \geq 2$, then resulting code $\mathcal{C}$ has

$$T = |\mathcal{C}_Q| = Q^{(k-1)r+1}$$

code words, each has weight $w = n_Q = Q^r - 1$ and length

$$n = Qn_Q = Q(Q^r - 1).$$

We can give lower bound for the minimum distance of $\mathcal{C}_Q$ (cf. Blahut (1984)),

$$d_Q \geq Q^r - 1 - (k-1)Q^{r-1}.$$

We can give lower bound for the order of $\mathcal{C}$,

$$\begin{aligned} M \geq M_0 &= \frac{n_Q - 1}{n_Q - d_Q} \\ &\geq \frac{(Q^r-1)-1}{(Q^r-1)-(Q^r-1-(k-1)Q^{r-1})} \\ &= \frac{Q^r-2}{(k-1)Q^{r-1}} \\ &\simeq \frac{Q}{k-1}, \end{aligned}$$

so this is approximately the same as in the case of Reed–Solomon code. Advantage of using BCH code is that we get a huge number T for potential users, even for small r. True enough code length n is also larger than in the Reed-Solomon case.

Reader can find a detailed survey on code constructions in Dyachkov et al. (2000). A promising code construction method may be the one based on algebraic geometry codes (cf. Ericson and Zinoviev (1987), Lint and Springer (1987)).

2.5 Performance evaluation of the Kautz–Singleton code

One of the most popular construction of ZFD codes is the Kautz–Singleton construction (cf. Kautz and Singleton (1964), Zinoviev (1983), Erdős et al. (1985), Győri (2003)) which is based on a Reed–Solomon code.

Let us take a Reed–Solomon code of maximum length over GF(q) with parameters $(N = q-1, K)$, so the number of code words, i.e., the number of users is

$$T = q^K.$$

This code can be mapped to a binary code by concatenating it with the identity matrix, so each element of GF(q) is replaced by a binary pattern of length q and weight 1:

$$\begin{array}{ccc} 0 & \mapsto & 0\ldots001, \\ \alpha^0 & \mapsto & 0\ldots010, \\ \alpha^1 & \mapsto & 0\ldots100, \\ & \vdots & \\ \alpha^{q-2} & \mapsto & 1\ldots000, \end{array}$$

where α is a primitive element in GF(q). In this case the binary code has weight $w = N = q - 1$, length $n = qN = q(q-1)$ and ZFD property of order M_0 if:

$$M_0 = \left\lfloor \frac{N-1}{N-d_{\min}} \right\rfloor = \left\lfloor \frac{N-1}{K-1} \right\rfloor = \left\lfloor \frac{q-2}{\frac{\log T}{\log q} - 1} \right\rfloor \tag{2.20}$$

(cf. Kautz and Singleton (1964)) which implies that

$$\frac{\log T}{\log q} \leq \frac{q-2}{M_0} + 1.$$

So, if T and M_0 are given, then q can be calculated, which determines the code length n, too. (Remember, that q must be a prime power.)

We would like to use such codes if more than M_0 users may communicate simultaneously. In this section the error probability will be investigated in this case for *synchronous access*.

If more than M_0 users are communicating in one time block, then it can happen that the Boolean sum of the code words of some ($> M_0$) users covers the code word of another user. Our task is to calculate the probability of this event, which is called error probability.

Select a user, and call it tagged user. Let $U_1, U_2, \ldots, U_m$ be the identifiers of the interfering users (if m users are active) which are independent random variables. They are uniformly distributed on the set of potentially interfering users (all users except the tagged user). For the sake of simplicity we use the model *sampling with replacement.* Since for practical cases $T \gg M$, it can be shown that the distributions for sampling with and without replacement are close to each other (cf. Györfi, Jordán and Vajda (2000)). Let $S(U_i)$ be the set of positions where user U_i covers the 1's of the tagged user. Define V_m as the size of the set of the covered positions of the tagged user, so

$$V_m = \left| \bigcup_{i=1}^{m} S(U_i) \right|.$$

Let us denote the detection error probability by $P_e(m)$ if exactly m users are active in the channel. Detection error occurs if all 1's of the tagged user are covered by the others. The Kautz–Singleton construction results a constant weight binary code with weight $w = q-1$, therefore

$$P_e(m) = \mathbf{P}\{V_m = w\} = \mathbf{P}\{V_m = q-1\}.$$

$|S(U_i)|$ for all $i = 1, \dots, m$ are independent, identically distributed random variables, and their distribution can be calculated. Introduce the notation

$$\mathbf{P}\{|S(U_i)| = \ell\} = p_\ell, \qquad 0 \le \ell \le K-1$$

where

$$\sum_{\ell=0}^{K-1} p_\ell = 1.$$

We note that the code word of an arbitrary user can cover the code word of the tagged user in at most $K-1$ positions, so $|S(U_i)| \le K-1$. (This is because of the MDS property of the Reed–Solomon code. As the minimum distance is $d_{\min} = N-K+1$, the number of identical coordinates between two code words is at most $N - d_{\min} = K-1$.)

$\{V_m\}$ forms a homogeneous Markov chain on the state space $\{0, 1, \dots, q-1\}$, so its distribution can be calculated in a recursive way (cf. Györfi, Jordán and Vajda (2000)). For $m = 1$ we have the initial distribution of the chain:

$$\mathbf{P}\{V_1 = \ell\} = \mathbf{P}\{|S(U_1)| = \ell\} = p_\ell \qquad (0 \le \ell \le K-1).$$

$\{V_m\}$ is monotonically increasing, because if we add another user to the active set, they all together can cover at least the same number of positions than in the previous step. The growth can be between 0 and $K-1$.

The transition probability matrix of the Markov chain can be calculated in the following way. Growth i can happen if the new user covers the code word of tagged user in $i+k$ positions (of course $i+k \le K-1$) from which i are out of the previously non-covered ones and k have been previously covered. If the number of previously covered positions is j, we need k positions out of this j, and i positions out of the other $w-j$. A new user can cover the code word of the tagged user in $i+k$ positions with probability p_{i+k}. So, the transition probability matrix contains the following values ($m \ge 2$):

$$\mathbf{P}\{V_m = j+i \mid V_{m-1} = j\} = \sum_{k=0}^{\min\{K-1-i,j\}} p_{i+k} \frac{\binom{w-j}{i}\binom{j}{k}}{\binom{w}{i+k}}$$

for $0 \le j, j+i \le w,\ 0 \le i \le K-1$.

Remark. Since the Reed–Solomon code is linear, the difference between the q-ary code word of the tagged user and the q-ary code word of an arbitrary user runs through all q-ary code words except the all 0 one, and the number of covered positions corresponds to the number of zero positions in the difference. If the i^{th} position of a code word is 0, then α^i is a root of its message polynomial $u(x)$. It is easy to prove that among the message polynomials there are the same number having 0's at positions $i_1, i_2, \dots, i_m$ and $i'_1, i'_2, \dots, i'_m$ $(0 \leq m \leq K-1)$, thus the code words having 0's at a given m-tuples are uniformly distributed on the code words having 0's at exactly m coordinates. If a code word has 0's at positions $i_1, i_2, \dots, i_m$, then its message polynomial is in the form of

$$u(x) = (x - \alpha^{i_1})^{k_1}(x - \alpha^{i_2})^{k_2} \cdots (x - \alpha^{i_m})^{k_m} f(x)$$

where $1 \leq k_j$ for all $1 \leq j \leq m$, $f(x)$ is an irreducible polynomial over $\mathrm{GF}(q)$, and $k_1 + k_2 + \cdots + k_m + \deg\{f(x)\} \leq K-1$. If we assign to this code word the code word having message polynomial

$$u(x) = (x - \alpha^{i'_1})^{k_1}(x - \alpha^{i'_2})^{k_2} \cdots (x - \alpha^{i'_m})^{k_m} f(x)$$

for the same k_j's and $f(x)$, then this is a bijection between code words having 0's at positions $i_1, i_2, \dots, i_m$ and $i'_1, i'_2, \dots, i'_m$.

We can calculate the detection error probability for different number of active users m recursively as the probability of the last position of the Markov chain:

$$P_e(m) = \mathbf{P}\{V_m = w\}, \tag{2.21}$$

and for this we only need the distribution of $|S(U_i)|$.

Lemma 2.16 (Győri (2004)). *The distribution of* $|S(U_i)|$ *is the following:*

$$p_\ell = \mathbf{P}\{|S(U_i)| = \ell\} = \frac{\binom{q-1}{\ell}(q-1)\sum\limits_{k=0}^{K-\ell-1}(-1)^k\binom{q-\ell-2}{k}q^{K-\ell-k-1}}{q^K - 1}$$

for all $0 \leq \ell \leq K-1$.

Proof. The number of covered positions of the tagged user caused by another user is the number of positions in the binary code words where both have 1's. As the Kautz–Singleton construction maps a Reed–Solomon code to binary code by concatenation with the identity matrix, this number is equal to the number of identical coordinates in the q-ary Reed–Solomon code words. This is called the Hamming correlation between the two code words. Our goal is to calculate the distribution of the number of identical coordinates while code word of the other user runs through all possible code words except the tagged user's one. Since the Reed–Solomon code is linear, this distribution is identical to the distribution of the number of zero coordinates of the code words except the all 0 one.

Weight distribution function of MDS codes A_w gives the number of code words having weight w (cf. Blahut (1984)). $A_0 = 1$ and

$$A_w = \binom{N}{w}(q-1)\sum_{k=0}^{w-d_{\min}}(-1)^k\binom{w-1}{k}q^{w-d_{\min}-k}$$

for all $d_{\min} \leq w \leq N$, otherwise it is 0. The number of zero coordinates trivially equals to $N-w$. We get probabilities p_ℓ if $A_{N-\ell}$ is divided by the number of all possible code words except the all 0 one

$$p_\ell = \frac{A_{N-\ell}}{q^K-1}.$$

In Kautz–Singleton construction we have the minimum distance $d_{\min} = N - K + 1$ and code length $N = q-1$, so p_ℓ can be calculated in the following way:

$$p_\ell = \frac{\binom{q-1}{\ell}(q-1)\sum_{k=0}^{K-\ell-1}(-1)^k\binom{q-\ell-2}{k}q^{K-\ell-k-1}}{q^K-1}$$

□

Györfi, Jordán and Vajda (2000) conjectured that the distribution of V_m is approximately Gaussian, so

$$P_e(m) \approx \Phi\left(-\frac{w-\mathbf{E}\{V_m\}}{\boldsymbol{\sigma}\{V_m\}}\right).$$

For calculating the detection error probability the mean value and the variance of V_m is needed. Györfi, Jordán and Vajda (2000) derived such a result, but they considered only the case when $K \leq 3$. This result can be extended to our case when K can be greater than 3.

Lemma 2.17 (Győri (2004)). *The mean value and the variance of Markov chain $\{V_m\}$ are*

$$\mathbf{E}\{V_m\} = w\left(1-\left(1-\frac{\bar{c}}{w}\right)^m\right)$$

and

$$\boldsymbol{\sigma}^2\left\{V_m\right\} = \mathbf{E}\{V_m\} - \mathbf{E}\{V_m\}^2 + w(w-1)\left(1 - 2\left(1-\tfrac{\bar{c}}{w}\right)^m + \left(\sum_{\ell=0}^{K-1} p_\ell \frac{\binom{w-2}{\ell}}{\binom{w}{\ell}}\right)^m\right), \tag{2.22}$$

where

$$\bar{c} = \mathbf{E}\{|S(U_1)|\} = \sum_{\ell=1}^{K-1} \ell p_\ell. \tag{2.23}$$

Proof. As

$$V_m = \left|\bigcup_{i=1}^{m} S(U_i)\right| = \sum_{j=1}^{w} I_{\left\{j \in \bigcup\limits_{i=1}^{m} S(U_i)\right\}},$$

and random variables $S(U_i)$ are independent, identically distributed, we get

$$\begin{aligned} \mathbf{E}\{V_m\} &= \sum_{j=1}^{w} \mathbf{P}\left\{j \in \bigcup_{i=1}^{m} S(U_i)\right\} \\ &= \sum_{j=1}^{w} \left(1 - \left(1 - \mathbf{P}\{j \in S(U_1)\}\right)^m\right). \end{aligned}$$

One can write

$$\begin{aligned} \mathbf{P}\{j \in S(U_1)\} &= \sum_{\ell=1}^{K-1} \mathbf{P}\{j \in S(U_1) \mid |S(U_1)| = \ell\} \mathbf{P}\{|S(U_1)| = \ell\} \\ &= \sum_{\ell=1}^{K-1} \frac{\binom{w-1}{\ell-1}}{\binom{w}{\ell}} p_\ell \\ &= \sum_{\ell=1}^{K-1} \frac{\ell p_\ell}{w} = \frac{\bar{c}}{w}, \end{aligned}$$

where

$$\bar{c} = \mathbf{E}\{|S(U_1)|\} = \sum_{\ell=1}^{K-1} \ell p_\ell,$$

so

$$\mathbf{E}\{V_m\} = w\left(1 - \left(1 - \frac{\bar{c}}{w}\right)^m\right).$$

For the second moment the following is true:

$$\begin{aligned}
\mathbf{E}\{V_m^2\} &= \mathbf{E}\left\{\left(\sum_{j=1}^{w} I_{\left\{j\in\bigcup\limits_{i=1}^{m} S(U_i)\right\}}\right)^2\right\}\\
&= \mathbf{E}\{V_m\} + \sum_{j\neq k}\mathbf{P}\left\{j\in\bigcup_{i=1}^{m} S(U_i), k\in\bigcup_{i=1}^{m} S(U_i)\right\}\\
&= \mathbf{E}\{V_m\} + \sum_{j\neq k}\left(1-\mathbf{P}\left\{j\notin\bigcup_{i=1}^{m} S(U_i)\right\}-\mathbf{P}\left\{k\notin\bigcup_{i=1}^{m} S(U_i)\right\}\right.\\
&\quad\left.+\mathbf{P}\left\{\left\{j\notin\bigcup_{i=1}^{m} S(U_i)\right\}\cap\left\{k\notin\bigcup_{i=1}^{m} S(U_i)\right\}\right\}\right)\\
&= \mathbf{E}\{V_m\} + w(w-1)\cdot\left(1-2\left(1-\tfrac{\bar{c}}{w}\right)^m+\left(\sum_{\ell=0}^{K-1} p_\ell\frac{\binom{w-2}{\ell}}{\binom{w}{\ell}}\right)^m\right),
\end{aligned}$$

from which it follows that

$$\boldsymbol{\sigma}^2\left\{V_m\right\} = \mathbf{E}\{V_m\}-\mathbf{E}\{V_m\}^2+w(w-1)\left(1-2\left(1-\tfrac{\bar{c}}{w}\right)^m+\left(\sum_{\ell=0}^{K-1} p_\ell\frac{\binom{w-2}{\ell}}{\binom{w}{\ell}}\right)^m\right).$$

□

Next, we give some upper bounds on detection error probability which can be easily calculated numerically. For the first one we apply Hoeffding's inequality:

Lemma 2.18 (Győri (2004)). *If* $\frac{w}{m}\geq\mathbf{E}\{|S(U_1)|\}$, *then the detection error probability can be upper bounded as*

$$P_e(m)\leq\exp\left(-\frac{2m\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)^2}{(K-1)^2}\right).$$

Proof.

$$\begin{aligned}
P_e(m) &= \mathbf{P}\{V_m=w\}\\
&= \mathbf{P}\left\{\left|\bigcup_{i=1}^{m} S(U_i)\right|=w\right\}\\
&\leq \mathbf{P}\left\{\sum_{i=1}^{m}|S(U_i)|\geq w\right\}\\
&= \mathbf{P}\left\{\frac{1}{m}\sum_{i=1}^{m}\left(|S(U_i)|-\mathbf{E}\{|S(U_i)|\}\right)\geq\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right\}.
\end{aligned}$$

Let us apply Hoeffding's inequality (Lemma B.3) to this probability with $a = 0,\ b = K - 1$. If $\varepsilon := \frac{w}{m} - \mathbf{E}\{|S(U_1)|\} > 0$ the following bound stands:

$$P_e(m) \leq \exp\left(-\frac{2m\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)^2}{(K-1)^2}\right),$$

otherwise we use trivial bound

$$P_e(m) \leq 1,$$

where $\mathbf{E}\{|S(U_1)|\}$ can be calculated by (2.23). □

The next bound follows from Bernstein's inequality:

Lemma 2.19 (Győri (2004)). *If* $\frac{w}{m} \geq \mathbf{E}\{|S(U_1)|\}$, *then the detection error probability can be upper bounded as*

$$P_e(m) \leq \exp\left(-\frac{m\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)^2}{2\boldsymbol{\sigma}^2\{|S(U_1)|\}+2\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)(K-1)/3}\right).$$

Proof. Similarly to the first part of the proof of Lemma 2.18, and then by applying Bernstein's inequality (Lemma B.2) an alternative upper bound can be calculated. If $\varepsilon := \frac{w}{m} - \mathbf{E}\{|S(U_1)|\} > 0$, the error probability can be bounded

$$\begin{aligned} P_e(m) &\leq \mathbf{P}\left\{\frac{1}{m}\sum_{i=1}^{m}\left(|S(U_i)| - \mathbf{E}\{|S(U_i)|\}\right) \geq \frac{w}{m} - \mathbf{E}\{|S(U_1)|\}\right\} \\ &\leq \exp\left(-\frac{m\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)^2}{2\boldsymbol{\sigma}^2\{|S(U_1)|\}+2\left(\frac{w}{m}-\mathbf{E}\{|S(U_1)|\}\right)(K-1)/3}\right) \end{aligned}$$

otherwise we use trivial bound

$$P_e(m) \leq 1,$$

where $\boldsymbol{\sigma}^2\{|S(U_1)|\}$ can be calculated as

$$\boldsymbol{\sigma}^2\{|S(U_1)|\} = \sum_{\ell=1}^{K-1} \ell^2 p_\ell - \bar{c}^2.$$

□

Table 2.1: Detection error probabilities for at most $M_{\max}$ active users

T	M_0	$M_{\max}$	q	K	n	P_e Markov
10^5	1	1	7	6	42	0
10^5	2	4	11	5	110	$7.952 \cdot 10^{-6}$
10^5	2	6	13	5	156	$8.295 \cdot 10^{-6}$
10^5	7	20	23	4	506	$7.686 \cdot 10^{-6}$
10^5	25	85	53	3	2756	$8.247 \cdot 10^{-6}$
10^4	1	1	7	5	42	0
10^4	3	4	11	4	110	$3.410 \cdot 10^{-6}$
10^4	10	21	23	3	506	$7.249 \cdot 10^{-6}$

Let us suppose that the maximum number of simultaneously active users $M_{\max}$ is given. Kautz–Singleton construction guarantees error free detection if at most M_0 users are active (cf. eq. (2.20)). We have detection error if the number of active users m is between M_0 and $M_{\max}$. Because of the detection error probability is a monotonically increasing function of the number of active users, it can be upper bounded by applying the Markov modell (cf. eq. (2.21)) for the worst case situation (when $M_{\max}$ users are active):

$$P_e \leq P_e(M_{\max}).$$

There are some code parameters for $P_e \leq 10^{-5}$ in Table 2.1. M_0 denotes the maximum number of active users for error-free detection. If we allow for at most $M_{\max}$ users to communicate simultaneously (instead of at most M_0) the detection error P_e can be seen in the last column of the table. We can conclude that it is possible to allow the maximum number of active users to be much bigger than the theoretical error-free limit while keeping the detection error probability small.

Figures 2.2 and 2.3 illustrate how many users can communicate simultaneously as the function of the detection error probability P_e for given $T = 10^5$ and n.

Let us suppose that a user is active with probability p independently from the others. The number of active users is the sum of T i.i.d. indicator random variables, so it is binomially distributed with parameters (T, p). Kautz–Singleton construction guarantees error free detection if the number of active users is not greater than M_0. The detection error probability can be calculated in the following way:

$$P_e^*(T, M_0, p) = \sum_{m=M_0+1}^{T} \binom{T}{m} p^m (1-p)^{T-m} P_e(m).$$

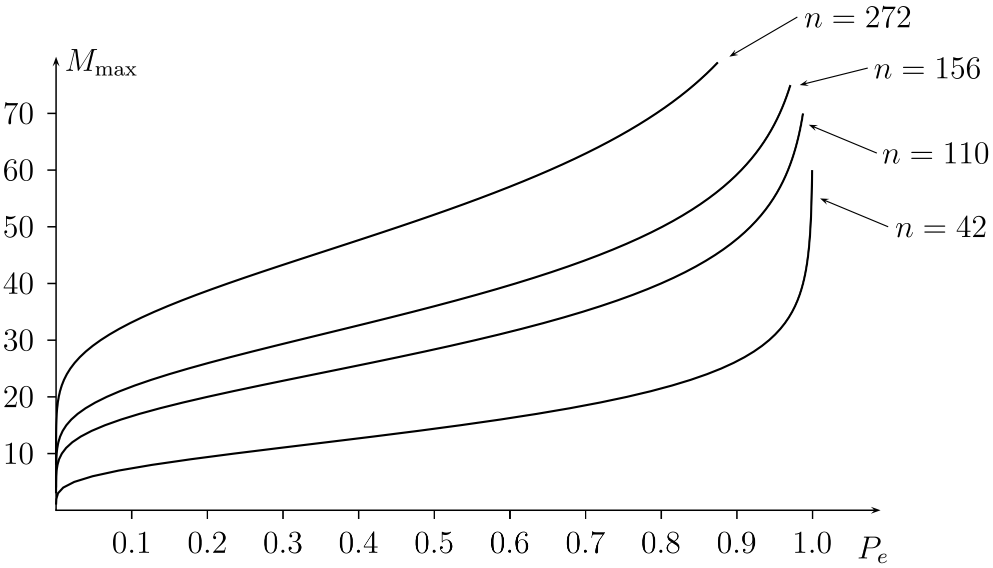

Figure 2.2: Maximum number of simultaneously active users for given detection error probability P_e, code length n and $T = 10^5$

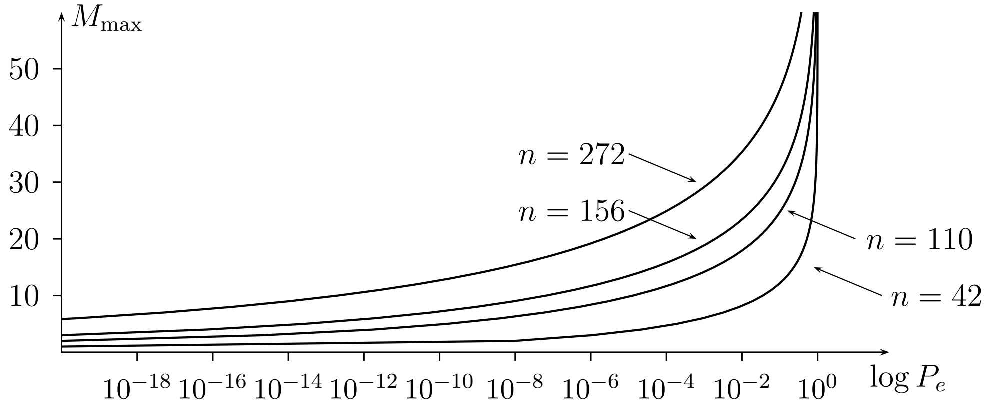

Figure 2.3: Maximum number of simultaneously active users for given detection error probability P_e in logarithmic scale, code length n and $T = 10^5$

Consider that the number of potential users is $T = 10^5$, Table 2.2 contains detection error probabilities for different activities p. There are two rows for each activity p. The first row corresponds to the conventional design resulting the smallest code length n when detection error probability is calculated such that $P_e(m)$ is upper bounded simply by 1. The second row corresponds to the design resulting the smallest code length n when detection error probability is calculated such that $P_e(m)$ is derived exactly from the Markov chain model.

Table 2.2: Detection error probabilities for $T = 10^5$

p	M_0	q	K	n	P_e^* Markov	P_e^* Φ	P_e^* Hoef	P_e^* Bern
10^{-7}	2	11	5	110	$7.38 \cdot 10^{-14}$	$3.91 \cdot 10^{-15}$	$1.83 \cdot 10^{-8}$	$1.89 \cdot 10^{-8}$
10^{-7}	1	7	6	42	$1.69 \cdot 10^{-8}$	$1.25 \cdot 10^{-9}$	$2.39 \cdot 10^{-5}$	$1.71 \cdot 10^{-5}$
10^{-6}	3	13	5	156	$4.67 \cdot 10^{-13}$	$6.88 \cdot 10^{-14}$	$4.56 \cdot 10^{-7}$	$3.59 \cdot 10^{-7}$
10^{-6}	1	7	6	42	$1.92 \cdot 10^{-6}$	$2.25 \cdot 10^{-7}$	$2.28 \cdot 10^{-3}$	$1.63 \cdot 10^{-3}$
10^{-5}	7	23	4	506	$1.01 \cdot 10^{-17}$	$6.12 \cdot 10^{-17}$	$4.61 \cdot 10^{-8}$	$9.79 \cdot 10^{-8}$
10^{-5}	1	9	6	72	$1.79 \cdot 10^{-5}$	$6.83 \cdot 10^{-6}$	$8.12 \cdot 10^{-2}$	$6.18 \cdot 10^{-2}$
10^{-4}	26	53	3	2756	$7.60 \cdot 10^{-28}$	$1.68 \cdot 10^{-22}$	$2.28 \cdot 10^{-10}$	$7.87 \cdot 10^{-9}$
10^{-4}	5	19	4	342	$9.65 \cdot 10^{-6}$	$2.46 \cdot 10^{-5}$	$2.76 \cdot 10^{-1}$	$2.04 \cdot 10^{-1}$
10^{-3}	26	53	3	2756	$6.20 \cdot 10^{-6}$	$7.63 \cdot 10^{-5}$	$2.61 \cdot 10^{-1}$	$2.61 \cdot 10^{-1}$

Table 2.3: Detection error probabilities for $T = 10^4$

p	M_0	q	K	n	P_e^* Markov	P_e^* Φ	P_e^* Hoef	P_e^* Bern
10^{-6}	3	11	4	110	$1.44 \cdot 10^{-15}$	$6.85 \cdot 10^{-16}$	$4.36 \cdot 10^{-11}$	$5.07 \cdot 10^{-11}$
10^{-6}	1	7	5	42	$1.38 \cdot 10^{-8}$	$1.23 \cdot 10^{-9}$	$1.58 \cdot 10^{-5}$	$1.38 \cdot 10^{-5}$
10^{-5}	3	11	4	110	$1.58 \cdot 10^{-11}$	$8.13 \cdot 10^{-12}$	$4.17 \cdot 10^{-7}$	$4.78 \cdot 10^{-7}$
10^{-5}	1	7	5	42	$1.62 \cdot 10^{-6}$	$2.22 \cdot 10^{-7}$	$1.53 \cdot 10^{-3}$	$1.32 \cdot 10^{-3}$
10^{-4}	10	23	3	506	$4.66 \cdot 10^{-20}$	$1.69 \cdot 10^{-17}$	$3.20 \cdot 10^{-11}$	$2.40 \cdot 10^{-10}$
10^{-4}	1	9	5	72	$1.64 \cdot 10^{-5}$	$6.80 \cdot 10^{-6}$	$4.61 \cdot 10^{-2}$	$4.66 \cdot 10^{-2}$
10^{-3}	5	19	4	342	$9.62 \cdot 10^{-6}$	$2.46 \cdot 10^{-5}$	$2.76 \cdot 10^{-1}$	$2.04 \cdot 10^{-1}$

In both cases we would like to guarantee that detection error probability is below 10^{-5}. Table 2.3 contains detection error probabilities for $T = 10^4$.

Györfi, Jordán and Vajda (2000) found that for the collision channel the Gaussian approximation of decoding error probability is always greater than the exact value of P_e. In our case this is not true, however, it is a good approximation. In the column "P_e Φ" of Tables 2.2 and 2.3 it can be seen that for some parameters Gaussian approximation can be smaller than the exact value of the detection error probability.

2.6 Case study for UD(2)

2.7 Random activity

Chapter 3

OR channel: asynchronous access

3.1 Fast frequency hopping

In this section *fast frequency hopping* (FFH) communications system is considered where the bandwidth is partitioned into L frequency subbands, and time is divided into intervals called slots. There is a longer unit called frame or block which consists of n slots.

A frequency *hopping sequence* (a two dimensional time–frequency binary code word) of length n is assigned to each user that specifies the sequence of frequency subbands in which the user is permitted to transmit a sine waveform during a time slot. If in a particular time slot at least one user sends a sine waveform in a frequency subband, then the receiver can detect it. Therefore the channel output is formally a binary $L \times n$ matrix which has a 1 at position (i, j), if there is at least one active user in subband i in the j^{th} time slot. This channel can be interpreted as a set of L parallel multiple access OR channels without feedback, noise and delay. Therefore the OR channel is a special case of FFH, when $L = 1$. But it is true vice versa. FFH is the same as the communication on a multiple access OR channel with constant weight code words. In the case of the OR channel the code words are simple binary vectors (they are mapped from the L-ary code words by concatenating with the identity matrix), that is why they are L times longer than the length of hopping sequences in our case (cf. Fig. 3.1).

If the users are always active and there is a common synchronization between the users, then the problem is trivial, with a time–frequency sharing the utilization 1 can be achieved such that each user has an own time–frequency slot, so $T = Ln$, where n is the block length.

For synchronous communication Einarsson (1980) introduced the following code. Let L be a prime power, and α be a primitive element of GF(L). If

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{K-1} & \alpha^{2(K-1)} & \cdots & \alpha^{(N-1)(K-1)} \end{bmatrix} \tag{3.1}$$

denotes the generator matrix of a Reed–Solomon code with parameters (N, K) (see (A.6)), then for $K = 2$ Einarsson (1980) defined the code words as

$$\mathbf{c} = (m, a)\mathbf{G},$$

where $a \in$ GF(L) is the address (identifier) of the user, and $m \in$ GF(L) is his message. As mentioned previously, the code word $\mathbf{c}$ can be represented by a binary matrix. There are L addresses in the system, so the total number of users is $T = L$, and the maximum number of active users M can be T.

If there is *no synchronization*, then Einarsson and Vajda (1987) introduced a code for $K = 3$

$$\mathbf{c} = (m, 1, a)\mathbf{G}.$$

In this construction $T = L$, too, but $M = \frac{L}{2}$.

For taking advantage of CDMA, we would like to allow much more users T in the system than the number of subbands L, but it should be guaranteed that only a small fraction $M \ll T$ of them can be active simultaneously (in a frame), then it could be decided which users are active (*identification*) and where begin their hopping sequences (*synchronization*). This is called the problem of *signature coding.* We are looking for a code (hopping sequences) of minimum length $n(T, M, L)$ achieving the previous requirements.

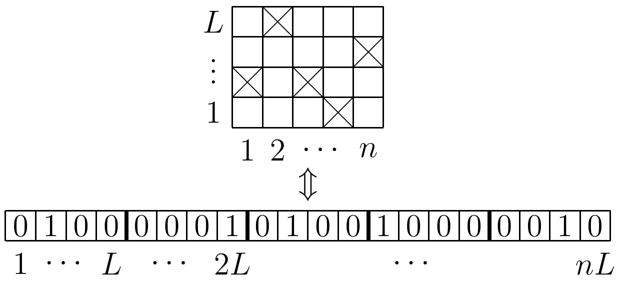

Figure 3.1: Mapping a time–frequency code word (FFH) to a binary code word (OR channel)

As the minimum code length for asynchronous access is at least the minimum code length for synchronous access, for giving a lower bound on the minimum code length it is enough to study here the synchronous case.

From Lemma 2.9 it follows that

$$T \leq M \frac{\binom{Ln}{n/M}}{\binom{n}{n/M}} \tag{3.2}$$

which gives a lower bound on the minimum code length. We denote by $\gtrsim$ and $\lesssim$ lower and upper bounds, respectively, which hold asymptotically in case of some given conditions.

Theorem 3.1. *If M, L are fixed and $T \to \infty$, then*

$$n_{asyn}(T, M, L) \geq n_{syn}(T, M, L) \gtrsim \frac{1}{Lh\left(\frac{1}{LM}\right) - h\left(\frac{1}{M}\right)} \log T,$$

where $h(\cdot)$ is the binary entropy function:

$$h(x) = -x \log x - (1-x)\log(1-x).$$

Proof. Let us apply Theorem B.1 on (3.2), then we get

$$T \leq 2^{Lnh\left(\frac{1}{LM}\right) - nh\left(\frac{1}{M}\right)}$$

from which the statement follows. □

Next a random coding argument will be applied to give an upper bound on the minimum code length n. By Theorem 2.7 the minimum code length n can be upper bounded in the frame synchronous case.

Theorem 3.2. *For frame synchronous access, if M, L are fixed and $T \to \infty$, then*

$$n_{syn}(T, M, L) \lesssim \frac{M+1}{-\log\left(1 - \left(1 - \frac{1}{L}\right)^M\right)} \log T.$$

In Theorem 3.3 we prove that, asymptotically, this upper bound is true for the asynchronous case, too.

Theorem 3.3. *For frame asynchronous access, if M, L are fixed and $T \to \infty$, then*

$$n_{asyn}(T, M) \lesssim \frac{M+1}{-\log\left(1 - \left(1 - \frac{1}{L}\right)^M\right)} \log T.$$

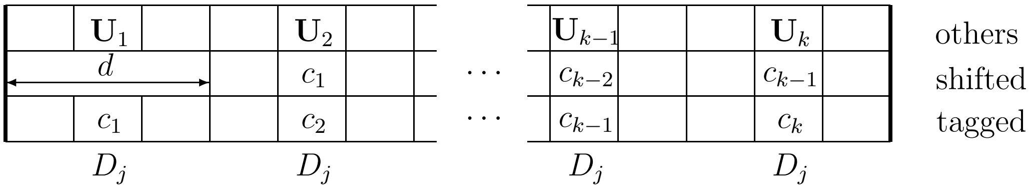

Figure 3.2: Components of the hopping sequences of the tagged user and other active users

For the proof of this theorem, we need some lemmata and considerations. Each user has a unique hopping sequence (code word, binary matrix) whose components are independently chosen of each other (and of the other users), and they are uniformly distributed on the frequency subbands $1, \ldots, L$.

We say that the channel output (the superposition of some code words) *covers* the hopping sequence of a user if the output matrix has 1's at all of the positions where the user's code word has 1's.

The detection is done by the following algorithm. A sliding window is used which length equals to the code length n. If, starting at a position, the binary matrix of the channel output covers the code word of a user, then it is declared as active (*identification*) beginning at this position (*synchronization*). Obviously, two different types of errors can happen: false identification, and false synchronization.

REMARK. During the design of the code it is supposed that the decoding algorithm does *not* have a memory (stateless). We have synchronization error only when a code word is covered by the *beginning* of its shifted version and some other code words. During the application of this code we use a decoding algorithm with memory (stateful). If a user is declared as active beginning at a given position, then he will be active in the next n time slots, so the algorithm need not to check its coverage in the next n time slots. Consequently, it does not cause synchronization problem if a code word is covered by the *end* of its shifted version and some other code words.

In the sequel it is supposed that *exactly* M users are active simultaneously (in each frame), which gives us an upper bound on the false identification and synchronization error probabilities compared to the original case, when *at most* M users are active.

Identification error occurs if (at most) M code words cover an other code word.

Lemma 3.1. *For frame asynchronous access*

$$\mathbf{P}\{\text{false identification}\} \leq \mathrm{e}^{(M+1)\ln T+M\ln n+n\ln\left(1-\left(1-\frac{1}{L}\right)^{M}\right)} \tag{3.3}$$

Proof. Let us fix M arbitrarily shifted hopping sequences and choose an $(M+1)^{\text{th}}$ (tagged) hopping sequence distinctly from the others. The probability that in a given time slot some of the other M users utilize the same subband as the tagged user (i.e., the code word of the tagged user is covered in a given time slot) is $1-\left(1-\frac{1}{L}\right)^M$. The probability that the code word of the tagged user is covered by the other users (in all the n time slots) is $\left(1-\left(1-\frac{1}{L}\right)^M\right)^n$, then the probability that there exists a user such that its code word is covered by another M users is at most

$$\begin{aligned}\mathbf{P}\{\text{false identification}\} &\leq \binom{T}{M}(T-M)n^M\left(1-\left(1-\tfrac{1}{L}\right)^M\right)^n\\ &\leq T^{M+1}n^M\left(1-\left(1-\tfrac{1}{L}\right)^M\right)^n\\ &= \mathrm{e}^{(M+1)\ln T+M\ln n+n\ln\left(1-\left(1-\frac{1}{L}\right)^M\right)},\end{aligned}$$

where the factor n^M is needed because of the shift of the other hopping sequences. □

The tagged user may also among the M active users (with some shift). We have synchronization error if the hopping sequence of the tagged user is covered by its shifted version and the hopping sequences of the other $M-1$ users.

Depending on the number of time slots d with which the hopping sequence of the tagged user is shifted, disjoint *classes of time slots* $D_1,\dots,D_d$ can be distinguished, where

$$D_j=\left\{j+\ell d:\ \ell=0,1,\dots,k-1\ \text{ and }\ k=\left\lfloor\tfrac{n-j}{d}\right\rfloor+1\right\}$$

$(j=1,\dots,d)$. Each time slot belongs to exactly one class. All classes have $k=\left\lfloor\frac{n}{d}\right\rfloor$ or $\left\lceil\frac{n}{d}\right\rceil$ elements, and $|D_1|+\dots+|D_d|=n$.

The probability $f(D_j)$ that in an arbitrary class of time slots D_j the code word of the tagged user is covered, can be derived in a recursive way, starting at the last slot. It is supposed that D_j contains $k=|D_j|$ time slots. (We note that the probability $f(D_j)$ depends only on the size of D_j and not on the actual elements of it.) For the sake of simplicity we use $1,2,\dots,k$ as slot indices instead of $j,j+d,\dots,j+(k-1)d$. c_ℓ denotes the location of the 1 for the code word of the tagged user at slot ℓ $(\ell=1,\dots,k)$, and let $\mathbf{U}_\ell=(U_\ell(1),\dots,U_\ell(L))$ be a binary vector of length L which has 1's in that positions where the corresponding frequency band has at least one active of the other users (at time slot ℓ). As the shifted code word of the tagged user is still not active at the first slot of the class D_j, there should be considered

M users instead of $M-1$ in the calculation of $\mathbf{U}_1$. (Remember, that exactly M active users were supposed in each time slot.) Thus, for all $\ell = 1, \ldots, k$

$$\mathbf{P}\{c_\ell = \phi\} = \frac{1}{L},$$

and

$$\begin{aligned} \mathbf{P}\{U_\ell(\phi) = 0\} &= \begin{cases} (1 - \frac{1}{L})^{M-1}, & \text{if } \ell = 2, \ldots, k, \\ (1 - \frac{1}{L})^{M}, & \text{if } \ell = 1 \end{cases} \\ \mathbf{P}\{U_\ell(\phi) = 1\} &= 1 - \mathbf{P}\{U_\ell(\phi) = 0\}, \end{aligned}$$

where $\phi = 1, \ldots, L$, and the components of vector $\mathbf{U}_\ell$ are independent of each other (cf. Fig. 3.2).

Lemma 3.2. *If V, W and Z are independent random variables, and $f(\cdot), g(\cdot)$ are arbitrary functions, then*

$$\mathbf{E}\{f(V,W)g(V,Z) \mid V\} = \mathbf{E}\{f(V,W) \mid V\}\mathbf{E}\{g(V,Z) \mid V\}.$$

Proof.

$$\begin{aligned} \mathbf{E}\{f(V,W)g(V,Z) \mid V\} &= \mathbf{E}\{\mathbf{E}\{f(V,W)g(V,Z) \mid V,W\} \mid V\} \\ &= \mathbf{E}\{f(V,W)\mathbf{E}\{g(V,Z) \mid V,W\} \mid V\} \\ &= \mathbf{E}\{f(V,W)\mathbf{E}\{g(V,Z) \mid V\} \mid V\} \\ &= \mathbf{E}\{g(V,Z) \mid V\}\mathbf{E}\{f(V,W) \mid V\}. \end{aligned}$$

□

Lemma 3.3. *For frame asynchronous access*

$$\mathbf{P}\{\text{code word of the tagged user is covered}\} = \left(1 - \left(1 - \tfrac{1}{L}\right)^M\right)^n.$$

Proof. Let us introduce the sequence of events

$$\begin{aligned} A_\ell &:= \{\text{time slot } \ell \text{ is covered}\} \\ &= \begin{cases} \{c_{\ell-1} = c_\ell\} \cup \{\{c_{\ell-1} \neq c_\ell\} \cap \{U_\ell(c_\ell) = 1\}\}, & \text{if } \ell = 2, \ldots, k, \\ \{U_1(c_1) = 1\}, \quad \text{if } \ell = 1. \end{cases} \end{aligned}$$

Thus

$$f(D_j) := \mathbf{P}\{\text{all positions in } D_j \text{ are covered}\} = \mathbf{P}\left\{\bigcap_{\ell=1}^{k} A_\ell\right\}.$$

We denote by a_i^ϕ ($i = 1, \dots, k$, $\phi = 1, \dots, L$) the conditional probabilities that the code word of the tagged user is covered up to the i^{th} position (in class D_j) given that the code word of the tagged user has ϕ at the i^{th} position ($c_i = \phi$).

$$a_i^\phi := \mathbf{P}\Big\{ \bigcap_{\ell=1}^{i} A_\ell \mid c_i = \phi \Big\}.$$

Therefore

$$\begin{aligned} f(D_j) &= \mathbf{P}\Big\{ \bigcap_{\ell=1}^{k} A_\ell \Big\} \\ &= \sum_{\phi=1}^{L} \mathbf{P}\Big\{ \bigcap_{\ell=1}^{k} A_\ell \mid c_k = \phi \Big\} \mathbf{P}\{c_k = \phi\} \\ &= \frac{1}{L} \sum_{\phi=1}^{L} a_k^\phi. \end{aligned}$$

Let us apply Lemma 3.2 with $V = \{c_{i-1}, c_i\}$, $W = \{c_1, \dots, c_{i-2}, \mathbf{U}_1, \dots, \mathbf{U}_{i-1}\}$, $Z = \{\mathbf{U}_i\}$, and $f(V, W) = I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}}$, $g(V, Z) = I_{\{A_i\}}$. (Note, that $\mathbf{P}\{B\} = \mathbf{E}\big\{I_{\{B\}}\big\}$ for an arbitrary event B.)

$$\begin{aligned} \mathbf{P}\Big\{ \bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i, c_{i-1} \Big\} &= \mathbf{E}\left\{ I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}} I_{\{A_i\}} \mid c_i, c_{i-1} \right\} \\ &= \mathbf{E}\{ f(V, W) g(V, Z) \mid V \} \\ &= \mathbf{E}\{ f(V, W) \mid V \} \mathbf{E}\{ g(V, Z) \mid V \} \\ &= \mathbf{E}\left\{ I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}} \mid c_i, c_{i-1} \right\} \mathbf{E}\big\{ I_{\{A_i\}} \mid c_i, c_{i-1} \big\} \\ &= \mathbf{P}\Big\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_i, c_{i-1} \Big\} \mathbf{P}\{ A_i \mid c_i, c_{i-1} \} \end{aligned}$$

By using this result we have for the conditional probabilities ($i \geq 2$)

$$\begin{aligned} a_i^\phi &= \mathbf{P}\Big\{ \bigcap_{\ell=1}^{i} A_\ell \mid c_i = \phi \Big\} \\ &= \mathbf{P}\Big\{ \bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i = \phi \Big\} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i = \phi, c_{i-1} = \psi \right\} \mathbf{P}\{c_{i-1} = \psi\} \\
&= \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_i = \phi, c_{i-1} = \psi \right\} \\
&\quad \cdot \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\} \mathbf{P}\{c_{i-1} = \psi\} \\
&= \frac{1}{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_{i-1} = \phi \right\} \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \phi\} \\
&\quad + \frac{1}{L} \sum_{\substack{\psi=1 \\ \psi \neq \phi}}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_{i-1} = \psi \right\} \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\} \\
&= \frac{1}{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_{i-1} = \phi \right\} \cdot 1 \\
&\quad + \frac{1}{L} \sum_{\substack{\psi=1 \\ \psi \neq \phi}}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_\ell \mid c_{i-1} = \psi \right\} \left(1 - \left(1 - \tfrac{1}{L}\right)^{M-1}\right) \\
&= \frac{1}{L} a_{i-1}^{\phi} + \frac{1}{L} \left(1 - (1 - \tfrac{1}{L})^{M-1}\right) \sum_{\substack{\psi=1 \\ \psi \neq \phi}}^{L} a_{i-1}^{\psi}.
\end{aligned}
$$

The first slot of the class D_j is different from the others, because the shifted code word of the tagged user is not active here.

$$
\begin{aligned}
a_1^{\phi} &:= \mathbf{P}\{A_1 \mid c_1 = \phi\} \\
&= \mathbf{P}\{\{U_1(c_1) = 1\} \mid c_1 = \phi\} \\
&= \mathbf{P}\{U_1(\phi) = 1\} \\
&= 1 - \left(1 - \tfrac{1}{L}\right)^{M}.
\end{aligned}
$$

Introduce the $L \times L$ matrix

$$
\mathbf{A} = \begin{pmatrix}
1 & 1 - \left(1 - \frac{1}{L}\right)^{M-1} & \cdots & 1 - \left(1 - \frac{1}{L}\right)^{M-1} \\
1 - \left(1 - \frac{1}{L}\right)^{M-1} & 1 & \cdots & 1 - \left(1 - \frac{1}{L}\right)^{M-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 - \left(1 - \frac{1}{L}\right)^{M-1} & 1 - \left(1 - \frac{1}{L}\right)^{M-1} & \cdots & 1
\end{pmatrix},
$$

then

$$\begin{aligned}
f(D_j) &= \left(\tfrac{1}{L},\ \tfrac{1}{L},\ \ldots,\ \tfrac{1}{L}\right)\left(a_k^1,\ a_k^2,\ \ldots,\ a_k^L\right)^T \\
&= \left(\tfrac{1}{L},\ \tfrac{1}{L},\ \ldots,\ \tfrac{1}{L}\right)\tfrac{1}{L}\mathbf{A}\left(a_{k-1}^1,\ a_{k-1}^2,\ \ldots,\ a_{k-1}^L\right)^T \\
&\vdots \\
&= \left(\tfrac{1}{L},\ \tfrac{1}{L},\ \ldots,\ \tfrac{1}{L}\right)\tfrac{1}{L^{k-1}}\mathbf{A}^{k-1}\left(a_1^1,\ a_1^2,\ \ldots,\ a_1^L\right)^T \\
&= \tfrac{1}{L^{k-1}}\left(\tfrac{1}{L},\ \tfrac{1}{L},\ \ldots,\ \tfrac{1}{L}\right)\mathbf{A}^{k-1}\begin{pmatrix}1-\left(1-\frac{1}{L}\right)^M \\ 1-\left(1-\frac{1}{L}\right)^M \\ \vdots \\ 1-\left(1-\frac{1}{L}\right)^M\end{pmatrix}.
\end{aligned}$$

For calculating the power of matrix **A** firstly its diagonal form is needed. It has L eigenvalues

$$\begin{aligned}
\lambda_1 &= L\left(1-\left(1-\tfrac{1}{L}\right)^M\right), \\
\lambda_2 = \cdots = \lambda_L &= \left(1-\tfrac{1}{L}\right)^{M-1},
\end{aligned}$$

and the corresponding eigenvectors are

$$\begin{aligned}
v_1 &= \left(1,\ 1,\ \cdots\ 1\right)^T, \\
v_2 &= \left(1,\ -1,\ 0,\ \cdots\ 0\right)^T, \\
v_3 &= \left(1,\ 0,\ -1,\ \cdots\ 0\right)^T, \\
&\vdots \\
v_L &= \left(1,\ 0,\ 0,\ \cdots\ -1\right)^T.
\end{aligned}$$

Thus, the decomposition of matrix **A** is

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1},$$

where

$$\mathbf{V} = \begin{pmatrix}1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -1\end{pmatrix}$$

$$\begin{aligned}
\mathbf{\Lambda} &= \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_2 \end{pmatrix} \\
\mathbf{V}^{-1} &= \begin{pmatrix} \frac{1}{L} & \frac{1}{L} & \cdots & \frac{1}{L} \\ \frac{1}{L} & \frac{1}{L}-1 & \cdots & \frac{1}{L} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{L} & \frac{1}{L} & \cdots & \frac{1}{L}-1 \end{pmatrix},
\end{aligned}$$

the $(k-1)^{\text{th}}$ power of $\mathbf{A}$ is

$$\mathbf{A}^{k-1} = \mathbf{V}\mathbf{\Lambda}^{k-1}\mathbf{V}^{-1},$$

where

$$\mathbf{\Lambda}^{k-1} = \begin{pmatrix} \lambda_1^{k-1} & 0 & \cdots & 0 \\ 0 & \lambda_2^{k-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_2^{k-1} \end{pmatrix},$$

and the probability $f(D_j)$ is

$$\begin{aligned}
f(D_j) &= \frac{1}{L^{k-1}}\left(1-\left(1-\tfrac{1}{L}\right)^M\right)\lambda_1^{k-1} \\
&= \frac{1}{L^{k-1}}\left(1-\left(1-\tfrac{1}{L}\right)^M\right)L^{k-1}\left(1-\left(1-\tfrac{1}{L}\right)^M\right)^{k-1} \\
&= \left(1-\left(1-\tfrac{1}{L}\right)^M\right)^k,
\end{aligned}$$

where, remember, $k = |D_j|$.

As the components of the code words are chosen independently of each other, and classes D_j's are disjoint, we have

$$\begin{aligned}
&\mathbf{P}\{\text{code word of the tagged user is covered}\} \\
&= \mathbf{P}\left\{\bigcap_{j=1}^{d}\{\text{all positions in } D_j \text{ are covered}\}\right\} \\
&= \prod_{j=1}^{d}\mathbf{P}\{\text{all positions in } D_j \text{ are covered}\}
\end{aligned}$$

$$
\begin{aligned}
&= \prod_{j=1}^{d} f(D_j)\\
&= \prod_{j=1}^{d} \left(1-\left(1-\tfrac{1}{L}\right)^M\right)^{|D_j|}\\
&= \left(1-\left(1-\tfrac{1}{L}\right)^M\right)^{\sum_{j=1}^{d}|D_j|}\\
&= \left(1-\left(1-\tfrac{1}{L}\right)^M\right)^{n},
\end{aligned}
$$

which is the same as the detection error probability for the synchronous case. □

Lemma 3.4. *For frame asynchronous access*

$$\mathbf{P}\{\text{false synchronization}\} \le \mathrm{e}^{M\ln T+M\ln n+n\ln\left(1-\left(1-\frac{1}{L}\right)^M\right)}. \tag{3.4}$$

Proof. Let us select $M-1$ arbitrarily shifted hopping sequences, and another (tagged) hopping sequence which is also active, but with some shift. By Lemma 3.3 the probability that the code word of the tagged user is covered by the others can be upper bounded by $\left(1-\left(1-\frac{1}{L}\right)^M\right)^n$, so

$$
\begin{aligned}
\mathbf{P}\{\text{false synchronization}\} &\le \binom{T}{M-1}(T-M+1)n^M\left(1-\left(1-\tfrac{1}{L}\right)^M\right)^n\\
&\le T^M n^M\left(1-\left(1-\tfrac{1}{L}\right)^M\right)^n\\
&= \mathrm{e}^{M\ln T+M\ln n+n\ln\left(1-\left(1-\frac{1}{L}\right)^M\right)},
\end{aligned}
$$

where the factor n^M is needed because of the shifts of the hopping sequences. □

Proof of Theorem 3.3. If a randomly chosen code $\mathcal{C}$ which has T hopping sequences of length n satisfy the requirements of identification and synchronization, then $\mathcal{C}$ can be applied for T users in communication by the fast frequency hopping scheme. Obviously,

$$\mathbf{P}\{\mathcal{C}\text{ is bad}\} \le \mathbf{P}\{\text{false identification}\} + \mathbf{P}\{\text{false synchronization}\}$$

and we need

$$\mathbf{P}\{\mathcal{C}\text{ is bad}\} < 1,$$

since then there is a good code. This gives an upper bound on minimum length of hopping sequences n. Thus, we need the following probabilities to tend to 0

$$\begin{aligned} \mathbf{P}\{\text{false identification}\} &\to 0, & (3.5)\\ \mathbf{P}\{\text{false synchronization}\} &\to 0. & (3.6)\end{aligned}$$

If we choose the code length n to

$$n = (1+\delta)\frac{M+1}{-\log\left(1-\left(1-\frac{1}{L}\right)^M\right)}\log T$$

for an arbitrary constant $\delta > 0$, the exponents in (3.3) and (3.4) become

$$-(M+1)\log T\left(\delta\left(1-\tfrac{\gamma}{M+1}\right)\ln 2-\left(1-\tfrac{1}{M+1}\right)\frac{\ln\left((1+\delta)\frac{M+1}{-\log\left(1-\left(1-\frac{1}{L}\right)^M\right)}\log T\right)}{\log T}\right),$$

where constant $\gamma = 1$ and 2, respectively. Both exponents tend to $-\infty$ when $T \to \infty$, that is why we have (3.5) and (3.6).

As the reasoning above is true for all arbitrarily small $\delta > 0$, the following asymptotic upper bound on the minimum code length n has been shown. If $T \to \infty$, then

$$n_{\text{asyn}}(T, M, L) \lesssim \frac{M+1}{-\log\left(1-\left(1-\frac{1}{L}\right)^M\right)}\log T.$$

□

3.2 Non-binary cyclically permutable codes

Similarly to the Kautz-Singleton code for OR channel, for fast frequency hopping, any L-ary code can be considered. However, for asynchronous access the codewords should be cyclically different. Gilbert (1963) has defined a *cyclically permutable code* to be a block code of block length n such that each codeword has n distinct cyclic shifts and such that no codeword can be obtained by cyclic shifting, one or more times, of another codeword.

More formally, let $\mathcal{C}$ be a block code then the *cyclic minimum distance* d_{cyc} of $\mathcal{C}$ is defined as

$$d_{\text{cyc}} = \min\left\{\min_{\mathbf{c}\in\mathcal{C},0<\tau<n} d(\mathbf{c}, S^\tau\mathbf{c}), \min_{\mathbf{c}\neq\mathbf{c}'\in\mathcal{C},0\leq\tau<n} d(\mathbf{c}, S^\tau\mathbf{c}')\right\},$$

therefore $\mathcal{C}$ is cyclically permutable if

$$d_{\text{cyc}} > 0.$$

Unfortunately, a linear code $\mathcal{C}$ is not cyclically permutable, since it contains the all 0 codeword, thus its cyclic distance is 0.

Constructing a cyclically permutable code, an obvious idea is to start with a linear cyclic code $\mathcal{C}$. The codewords $\mathbf{c}$ and $\mathbf{c}'$ are said to be in the same equivalence class if $S^\tau \mathbf{c} = \mathbf{c}'$ for some $0 < \tau < N$. Let $\mathcal{C}'$ be a subcode of $\mathcal{C}$ such that the codewords of $\mathcal{C}'$ lie in distinct cyclic equivalence classes and each of these classes contains n codewords (Gilbert (1963), Maracle, Wolverton (1974), Neumann (1964)). If $d_{\min}$ is the minimum distance of $\mathcal{C}$, and d_{cyc} is the cyclic distance of $\mathcal{C}'$ then

$$d_{\text{cyc}} \geq d_{\min}.$$

There are two problems with this construction:
(i) it is fairly laborous, it is not easily implementable,
(ii) it is hard to know a lower bound on $|\mathcal{C}'|$.

We now show a general way for constructing cyclically permutable codes, which is a nonlinear subcode of a linear cyclic code.

Let $\mathcal{C}$ be a linear cyclic code with minimum distance $d_{\min}$. Assume an orthogonal decomposition

$$\mathcal{C} = \mathcal{C}' + \mathcal{C}'^{\perp}$$

such that both $\mathcal{C}'$ and $\mathcal{C}'^{\perp}$ are cyclic, and $\mathcal{C}'^{\perp}$ contains a codeword $\mathbf{c}^*$ which has n distinct cyclic shifts. Put

$$\mathcal{C}^* = \mathcal{C}' + \mathbf{c}^*.$$

Theorem 3.4. *The code $\mathcal{C}^*$ is cyclically permutable with cyclic distance at least $d_{\min}$ and*

$$|\mathcal{C}^*| = |\mathcal{C}'|.$$

Proof. Consider two codewords from $\mathcal{C}^*$:

$$\mathbf{c} = \mathbf{c}_1 + \mathbf{c}^*$$

and

$$\mathbf{c}' = \mathbf{c}_2 + \mathbf{c}^*$$

$(\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}')$, where either $0 < \tau < n$ or $\mathbf{c}_1 \neq \mathbf{c}_2$. Then

$$d(\mathbf{c}, S^\tau \mathbf{c}') = w(\mathbf{c} - S^\tau \mathbf{c}') = w(\mathbf{c}_1 - S^\tau \mathbf{c}_2 + \mathbf{c}^* - S^\tau \mathbf{c}^*).$$

If $0 < \tau < n$ then $\mathbf{c}^* - S^\tau \mathbf{c}^* \neq \mathbf{0}$ and is from $\mathcal{C}'^\perp$, and because of $\mathbf{c}_1 - S^\tau \mathbf{c}_2 \in \mathcal{C}'$

$$\mathbf{c}_1 - S^\tau \mathbf{c}_2 + \mathbf{c}^* - S^\tau \mathbf{c}^* \neq \mathbf{0},$$

and therefore

$$w(\mathbf{c}_1 - S^\tau \mathbf{c}_2 + \mathbf{c}^* - S^\tau \mathbf{c}^*) \geq d_{\min}.$$

If $\tau = 0$ and $\mathbf{c}_1 \neq \mathbf{c}_2$ then

$$\mathbf{c}_1 - S^\tau \mathbf{c}_2 + \mathbf{c}^* - S^\tau \mathbf{c}^* = \mathbf{c}_1 - \mathbf{c}_2 \neq \mathbf{0},$$

and again

$$w(\mathbf{c}_1 - S^\tau \mathbf{c}_2 + \mathbf{c}^* - S^\tau \mathbf{c}^*) \geq d_{\min}.$$

□

For two code words $\mathbf{c}_1$ and $\mathbf{c}_2$, denote by

$$c(\mathbf{c}_1, \mathbf{c}_2)$$

the correlation of $\mathbf{c}_1$ and $\mathbf{c}_2$, i.e. it is the number of positions, where both $\mathbf{c}_1$ and $\mathbf{c}_2$ have 1, then the *cyclic maximum correlation* c_{cyc} of $\mathcal{C}$ is defined as

$$c_{\text{cyc}} = \max \left\{ \max_{\mathbf{c} \in \mathcal{C}, 0 < \tau < n} c(\mathbf{c}, S^\tau \mathbf{c}), \max_{\mathbf{c} \neq \mathbf{c}' \in \mathcal{C}, 0 \leq \tau < n} c(\mathbf{c}, S^\tau \mathbf{c}') \right\}.$$

Because of

$$c(\mathbf{c}_1, \mathbf{c}_2) = n - d(\mathbf{c}_1, \mathbf{c}_2)$$

we have that

$$c_{\text{cyc}} = n - d_{\text{cyc}}.$$

Similarly to the synchronous access, if

$$M c_{\text{cyc}} < n$$

then the code can serve M active users for asynchronous access, too.

3.3 A subcode of a Reed-Solomon code for fast frequency hopping

As an application of Theorem 3.4, assume that $L = q$ is a prime power, and let $\mathcal{C}$ be a Reed-Solomon code over $GF(q)$ with parameters $(n = q - 1, K)$. Introduce the subcode

$$\mathcal{C}^* = \{\mathbf{c} = (x_0, 1, x_2, \ldots, x_{K-1})\mathbf{G}\}.$$

Using the notations of Theorem 3.4, we can get $\mathcal{C}^*$ if

$$\mathcal{C}' = \{\mathbf{c} = (x_0, 0, x_2, \ldots, x_{K-1})\mathbf{G}\}$$

and

$$\mathcal{C}'^{\perp} = \{\mathbf{c} = (0, x_1, 0, \ldots, 0)\mathbf{G}\}$$

and

$$\mathbf{c}^* = (0, 1, 0, \ldots, 0)\mathbf{G} = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1}),$$

where $\mathbf{G}$ has been defined by (3.1). Then for the user population size we have

$$T = q^{K-1},$$

moreover, Theorem 3.4 implies that

$$c_{\text{cyc}} = n - d_{\text{cyc}} \leq n - (n - K + 1) = K - 1.$$

In fact, using the argument of ?????? one can prove that

$$c_{\text{cyc}} = K - 1.$$

This code with $K = 3$ is due to Vajda and Einarsson (1987), its modifications are investigated in A, Györfi, Massey (1992), Györfi, Vajda (1993) and Vajda (1995).

3.4 A subcode of a BCH code for fast frequency hopping

In a similar way we can construct a subcode of BCH code for fast frequency hopping with asynchronous access. Apply the notations of Section A.8. For a prime p put $q = p$, and let $\mathcal{C}$ be a BCH code over $GF(p)$ of length

$$n = p^r - 1$$

defined by the parity check polynomial

$$h(x) = l.c.m.\{M_0(x), M_1(x), M_2(x), \ldots, M_{K-1}\},$$

where $3 \leq K < p - 1$ and if α is a primitive element of $\mathrm{GF}(p^r)$, then $M_i(x)$ denotes the minimal polynomial of α^i over $GF(p)$. Because of $3 \leq K < p-1$

$$h(x) = M_0(x) \prod_{j=1}^{K-1} M_j(x).$$

Consider two other BCH codes with the same length. The code $\mathcal{C}'$ has the parity check polynomial

$$h_0(x) = M_0(x) \prod_{j=2}^{K-1} M_j(x),$$

while the code $\mathcal{C}'^{\perp}$ has the parity check polynomial

$$h_1(x) = M_1(x).$$

Zierler (1959) proved that

$$\mathcal{C} = \mathcal{C}' + \mathcal{C}'^{\perp}.$$

α is a primitive element of $GF(p^r)$ and $h_1(x) = M_1(x)$ is the minimal polynomial of α, therefore $M_1(x)$ is a primitive polynomial, so $\mathcal{C}'^{\perp}$ contains an m-sequence $\mathbf{c}^*$, which implies that the cyclic shifts of $\mathbf{c}^*$ are all different (cf. Theorem 7.44 in Lidl, Niederreiter (1986)). Define $\mathcal{C}^*$ by

$$\mathcal{C}^* = \mathcal{C}' + \mathbf{c}^*.$$

It is easy to see that the population size is

$$T = |\mathcal{C}^*| = p^{(K-2)r+1}.$$

Theorem 3.4 and (A.13) imply that

$$d_{cyc} \geq d_{min} \geq p^r - 1 - (K-1)p^{r-1},$$

therefore

$$c_{cyc} = n - d_{cyc} \leq (K-1)p^{r-1}.$$

This code is introduced in Györfi, Vajda (1993).

3.5 Asynchronous OR channel

Although the OR channel is a special case of FFH when the number of frequency subbands is one ($L = 1$), it is impossible the simply adapt the results of Section 3.1 to the OR channel. The problem is that the random code construction applied there does not work for $L = 1$. An other adaptation attempt would be the mapping of the code words of FFH to binary code words by concatenating them with the identity matrix. Unfortunately, in the case of asynchronous OR channel the time shift can be any multiple of a time slot and not just any multiple of L times the time slot (as in the case of

FFH). Therefore we can not construct independent classes of time slots. So, another random code construction should be used.

If frame asynchronous access is assumed, the coding method have to ensure not just the identification but the synchronization, too. In Theorem 2.8 it was given an upper bound on minimum code length $n_{\text{syn}}(T,M)$ in the case of synchronous access.

$$n_{\text{syn}}(T,M) \lesssim \mathrm{e}\ln 2\,(M+1)^2 \log T.$$

In this section we give upper bound on $n_{\text{asyn}}(T,M)$, and show that the bounds for synchronous and asynchronous access are asymptotically equal.

Theorem 3.5. *For frame asynchronous access, if M is fixed and $T \to \infty$*

$$n_{asyn}(T,M) \lesssim \mathrm{e}\ln 2\,(M+1)^2 \log T$$

The detection is done by the following algorithm. A sliding window is used which length equals to the code length n. If, starting at a position, the binary vector of the channel output covers the code word of a user, then it is declared as active (*identification*) beginning at this position (*synchronization*). Obviously, two different types of errors can happen: false identification, and false synchronization.

Remark. During the design of the code it is supposed that the decoding algorithm does *not* have a memory (stateless). We have synchronization error only when a code word is covered by the *beginning* of its shifted version and some other code words. During the application of this code we use a decoding algorithm with memory (stateful). If a user is declared as active beginning at a given position, then he will be active in the next n time slots, so the algorithm need not to check its coverage in the next n time slots. Consequently, it does not cause synchronization problem if a code word is covered by the *end* of its shifted version and some other code words.

In the sequel it is supposed that *exactly M* users are active simultaneously (in each time slot), which gives us an upper bound on the covering probabilities compared to the original case, when *at most M* users are active.

Identification error occurs if the Boolean sum of the code words of the active users covers the code word of an other user.

Lemma 3.5. *For frame asynchronous access, if $p = \frac{1}{M+1}$*

$$\mathbf{P}\{\text{false identification}\} \le \mathrm{e}^{(M+1)\ln T + M\ln n - \frac{n}{M+1}\mathrm{e}^{-1}}.$$

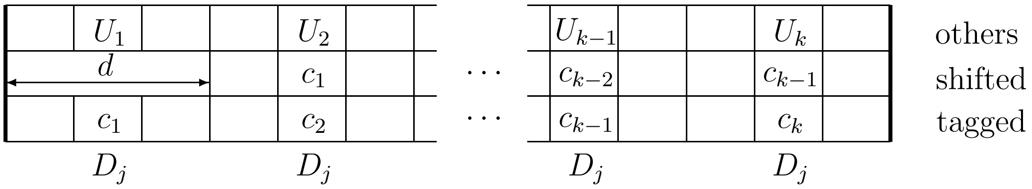

Figure 3.3: Bits of the tagged user and other active users

Proof. As the bits of the code words of the users are chosen independently of each other, the identification error probability can be similarly calculated as in Theorem 2.8. Let us select M arbitrarily shifted code words, and another (tagged) code word. The probability that in a given position the tagged code word has an uncovered 1 is $p(1-p)^M$. Therefore

$$\mathbf{P}\{\text{false identification}\} \le \binom{T}{M}(T-M)n^M\left(1-p(1-p)^M\right)^n,$$

where the factor n^M is needed because of the shift of the code words. Let $p := \frac{1}{M+1}$, then

$$\begin{aligned}\mathbf{P}\{\text{false identification}\} &\le \binom{T}{M}(T-M)n^M\left(1-\tfrac{1}{M+1}\left(1-\tfrac{1}{M+1}\right)^M\right)^n\\ &\le T^{M+1}n^M\left(1-\tfrac{\mathrm{e}^{-1}}{M+1}\right)^n\\ &\le T^{M+1}n^M\mathrm{e}^{-\frac{n}{M+1}\mathrm{e}^{-1}}\\ &= \mathrm{e}^{(M+1)\ln 2\log T+M\ln n-\frac{n}{M+1}\mathrm{e}^{-1}},\end{aligned} \tag{3.7}$$

where we applied that $\left(1-\frac{1}{M+1}\right)^M \ge \mathrm{e}^{-1}$, and $1+x \le \mathrm{e}^x$ for all $x \in \mathbb{R}$. □

Synchronization error occurs if a code word is covered by the shifted version of itself and some other active users. Depending on the number of bits d with which the code word of the tagged user is shifted, disjoint *classes of positions* $D_1, \dots, D_d$ can be distinguished, where

$$D_j = \{j + \ell d :\ \ell = 0, 1, \dots, k-1 \ \text{ and } \ k = \left\lfloor \tfrac{n-j}{d} \right\rfloor + 1\}$$

$(j = 1, \dots, d)$. Each position belongs to exactly one class. All classes have $k = \left\lfloor \frac{n}{d} \right\rfloor$ or $\left\lceil \frac{n}{d} \right\rceil$ elements, and $|D_1| + \cdots + |D_d| = n$.

The probability $f(D_j)$ that in an arbitrary class of positions D_j the tagged user has no uncovered 1's, can be derived in a recursive way, starting at the last position. It is supposed that D_j contains $k = |D_j|$ positions. (We note that the probability $f(D_j)$ depends only on the size of D_j and not on the actual elements of it.) For the sake of simplicity we use $1, 2, \ldots, k$ as position indices instead of $j, j+d, \ldots, j+(k-1)d$. c_ℓ denotes the component of the code word of the tagged user at position ℓ ($\ell = 1, \ldots, k$), and let U_ℓ be 0 if and only if all the other users have 0 at this position (else it is 1). As the shifted code word of the tagged user is still not active at the first position of the class D_j, there should be considered M users instead of $M-1$ in the calculation of U_1. (Remember, that exactly M active users were supposed in each position.) That is why for all $\ell = 1, \ldots, k$

$$\mathbf{P}\{c_\ell = 0\} = 1 - p, \qquad \mathbf{P}\{c_\ell = 1\} = p,$$

and

$$\begin{aligned} \mathbf{P}\{U_\ell = 0\} &= \begin{cases} (1-p)^{M-1}, & \text{if } \ell = 2, \ldots, k, \\ (1-p)^M, & \text{if } \ell = 1 \end{cases} \\ \mathbf{P}\{U_\ell = 1\} &= 1 - \mathbf{P}\{U_\ell = 0\} \end{aligned}$$

(cf. Fig. 3.3).

Lemma 3.6. *For frame asynchronous access, if* $p = \frac{1}{M+1}$

$$\mathbf{P}\{\text{code word of the tagged user is covered}\} \le \left(1 - p(1-p)^M\right)^n .$$

Proof. Let us introduce the sequence of events

$$\begin{aligned} A_\ell &:= \{\text{position } \ell \text{ is covered}\} \\ &= \begin{cases} \{c_{\ell-1} = 1\} \cup \{\{c_{\ell-1} = 0\} \cap \{c_\ell = 1, U_\ell = 0\}^c\}, & \text{if } \ell = 2, \ldots, k, \\ \{c_1 = 1, U_1 = 0\}^c, \ \text{if } \ell = 1, \end{cases} \end{aligned}$$

where $\{\ \}^c$ denotes the complement of an event. Thus

$$f(D_j) := \mathbf{P}\{\text{all 1's in class } D_j \text{ are covered}\} = \mathbf{P}\left\{\bigcap_{\ell=1}^{k} A_\ell\right\}.$$

We denote by a_i^ϕ ($i = 1, \ldots, k,\ \phi = 0, 1$) the conditional probabilities that there is no uncovered 1 up to the i^{th} position given that the tagged user has a 0 ($\phi = 0$) or 1 ($\phi = 1$) at the i^{th} position ($c_i = 0$ or 1), respectively.

$$a_i^\phi := \mathbf{P}\left\{\bigcap_{\ell=1}^{i} A_\ell \mid c_i = \phi\right\}.$$

Hence

$$\begin{aligned}
f(D_j) &= \mathbf{P}\left\{\bigcap_{\ell=1}^{k} A_\ell\right\} \\
&= \mathbf{P}\left\{\bigcap_{\ell=1}^{k} A_\ell \mid c_k = 1\right\}\mathbf{P}\{c_k = 1\} + \mathbf{P}\left\{\bigcap_{\ell=1}^{k} A_\ell \mid c_k = 0\right\}\mathbf{P}\{c_k = 0\} \\
&= pa_k^1 + (1-p)a_k^0.
\end{aligned}$$

Let us apply Lemma 3.2 with $V = \{c_{i-1}, c_i\}$, $W = \{c_1, \ldots, c_{i-2}, U_1, \ldots, U_{i-1}\}$, $Z = \{U_i\}$, and $f(V,W) = I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}}$, $g(V,Z) = I_{\{A_i\}}$. (Note, that $\mathbf{P}\{B\} = \mathbf{E}\big\{I_{\{B\}}\big\}$ for an arbitrary event B.)

$$\begin{aligned}
\mathbf{P}\left\{\bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i, c_{i-1}\right\} &= \mathbf{E}\left\{I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}} I_{\{A_i\}} \mid c_i, c_{i-1}\right\} \\
&= \mathbf{E}\{f(V,W)g(V,Z) \mid V\} \\
&= \mathbf{E}\{f(V,W) \mid V\}\mathbf{E}\{g(V,Z) \mid V\} \\
&= \mathbf{E}\left\{I_{\left\{\bigcap_{\ell=1}^{i-1} A_\ell\right\}} \mid c_i, c_{i-1}\right\}\mathbf{E}\big\{I_{\{A_i\}} \mid c_i, c_{i-1}\big\} \\
&= \mathbf{P}\left\{\bigcap_{\ell=1}^{i-1} A_\ell \mid c_i, c_{i-1}\right\}\mathbf{P}\{A_i \mid c_i, c_{i-1}\}
\end{aligned}$$

By using this result we have for the conditional probabilities ($i \geq 2$)

$$\begin{aligned}
a_i^\phi &= \mathbf{P}\left\{\bigcap_{\ell=1}^{i} A_\ell \mid c_i = \phi\right\} \\
&= \mathbf{P}\left\{\bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i = \phi\right\} \\
&= \sum_{\psi=0}^{1}\mathbf{P}\left\{\bigcap_{\ell=1}^{i-1} A_\ell \cap A_i \mid c_i = \phi, c_{i-1} = \psi\right\}\mathbf{P}\{c_{i-1} = \psi\} \\
&= \sum_{\psi=0}^{1}\mathbf{P}\left\{\bigcap_{\ell=1}^{i-1} A_\ell \mid c_i = \phi, c_{i-1} = \psi\right\}\mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\}\mathbf{P}\{c_{i-1} = \psi\} \\
&= \sum_{\psi=0}^{1} a_{i-1}^{\psi}\mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\}\mathbf{P}\{c_{i-1} = \psi\},
\end{aligned}$$

thus

$$\begin{aligned} a_i^1 &= pa_{i-1}^1 + (1-p)\left(1-(1-p)^{M-1}\right)a_{i-1}^0, \\ a_i^0 &= pa_{i-1}^1 + (1-p)a_{i-1}^0. \end{aligned}$$

The first position of the class D_j is different from the others, because the shifted code word of the tagged user is not active here.

$$\begin{aligned} a_1^\phi &:= \mathbf{P}\{A_1 \mid c_1 = \phi\} \\ &= \mathbf{P}\{\{c_1 = 1, U_1 = 0\}^c \mid c_1 = \phi\} \\ &= 1 - \mathbf{P}\{c_1 = 1, U_1 = 0 \mid c_1 = \phi\}, \end{aligned}$$

so

$$\begin{aligned} a_1^1 &= 1-(1-p)^M, \\ a_1^0 &= 1. \end{aligned}$$

Introduce the notation

$$\mathbf{A} = \begin{pmatrix} p & (1-p)(1-(1-p)^{M-1}) \\ p & 1-p \end{pmatrix},$$

then

$$\begin{aligned} f(D_j) &= \begin{pmatrix} p, & 1-p \end{pmatrix} \begin{pmatrix} a_k^1 \\ a_k^0 \end{pmatrix} \\ &= \begin{pmatrix} p, & 1-p \end{pmatrix} \mathbf{A} \begin{pmatrix} a_{k-1}^1 \\ a_{k-1}^0 \end{pmatrix} \\ &\vdots \\ &= \begin{pmatrix} p, & 1-p \end{pmatrix} \mathbf{A}^{k-1} \begin{pmatrix} a_1^1 \\ a_1^0 \end{pmatrix} \\ &= \begin{pmatrix} p, & 1-p \end{pmatrix} \mathbf{A}^{k-1} \begin{pmatrix} 1-(1-p)^M \\ 1 \end{pmatrix}. \end{aligned}$$

For calculating the power of matrix $\mathbf{A}$ firstly its diagonal form is needed. It has two eigenvalues

$$\begin{aligned} \lambda_1 &= \frac{1}{2} + \frac{1}{2}\sqrt{1-4p(1-p)^M}, \\ \lambda_2 &= \frac{1}{2} - \frac{1}{2}\sqrt{1-4p(1-p)^M}, \end{aligned}$$

and the corresponding eigenvectors are

$$v_1 = \left(\tfrac{\lambda_1-1+p}{p},\ 1\right)^T, \qquad v_2 = \left(\tfrac{\lambda_2-1+p}{p},\ 1\right)^T.$$

Thus, the decomposition of matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{pmatrix} \frac{\lambda_1-1+p}{p} & \frac{\lambda_2-1+p}{p} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \frac{p}{\lambda_1-\lambda_2} & -\frac{\lambda_2-1+p}{\lambda_1-\lambda_2} \\ -\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_1-1+p}{\lambda_1-\lambda_2} \end{pmatrix},$$

the $(k-1)^{\text{th}}$ power of $\mathbf{A}$ is

$$\mathbf{A}^{k-1} = \begin{pmatrix} \frac{\lambda_1-1+p}{p} & \frac{\lambda_2-1+p}{p} \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1^{k-1} & 0 \\ 0 & \lambda_2^{k-1} \end{pmatrix} \begin{pmatrix} \frac{p}{\lambda_1-\lambda_2} & -\frac{\lambda_2-1+p}{\lambda_1-\lambda_2} \\ -\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_1-1+p}{\lambda_1-\lambda_2} \end{pmatrix},$$

and the probability $f(D_j)$ is

$$\begin{aligned} f(D_j) &= \left(1+\sqrt{1-4q}\right)^{k-2} 2^{-(k-2)} \left(\frac{\frac{1}{2}-2q+q^2}{\sqrt{1-4q}} + \frac{1}{2} - q\right) \\ &\quad - \left(1-\sqrt{1-4q}\right)^{k-2} 2^{-(k-2)} \left(\frac{\frac{1}{2}-2q+q^2}{\sqrt{1-4q}} - \frac{1}{2} + q\right), \end{aligned}$$

where $q = p(1-p)^M$. Notice, that $0 \le q \le \frac{4}{27} \approx 0.148$ for all $M \ge 2$ and $p = \frac{1}{M+1}$. By considering that for such a q

$$\left(\frac{\frac{1}{2}-2q+q^2}{\sqrt{1-4q}} - \frac{1}{2} + q\right) \ge 0,$$

and

$$\left(\frac{\frac{1}{2}-2q+q^2}{\sqrt{1-4q}} + \frac{1}{2} - q\right) \le (1-q)^2,$$

$f(D_j)$ can be upper bounded

$$\begin{aligned} f(D_j) &\le \left(1+\sqrt{1-4q}\right)^{k-2} 2^{-(k-2)} (1-q)^2 \\ &= \left(\frac{1}{2}+\sqrt{\frac{1}{4}-q}\right)^{k-2} (1-q)^2 \\ &\le (1-q)^{k-2} (1-q)^2 \\ &= (1-q)^k \\ &= \left(1-p(1-p)^M\right)^k. \end{aligned} \tag{3.8}$$

If $M = 1$ and $p = \frac{1}{M+1}$, then

$$\begin{aligned} f(D_j) &= \left(\tfrac{1}{2},\ \tfrac{1}{2}\right) \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}^{k-1} \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} \\ &= \left(\tfrac{1}{2},\ \tfrac{1}{2}\right) \begin{pmatrix} \frac{1}{2^{k-1}} & 0 \\ \frac{k-1}{2^{k-1}} & \frac{1}{2^{k-1}} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 1 \end{pmatrix} \\ &= \frac{k+2}{2^{k+1}} \\ &\le \left(\frac{3}{4}\right)^k, \end{aligned}$$

where $1 - p(1-p)^M = \frac{3}{4}$ for $M = 1$, so inequality (3.8) is true for $M = 1$, too.

As the components of the code words are chosen independently of each other, and classes D_j's are disjoint, we have

$$\begin{aligned} &\mathbf{P}\{\text{code word of the tagged user is covered}\} \\ &\quad = \mathbf{P}\left\{ \bigcap_{j=1}^{d} \{\text{all 1's in class } D_j \text{ are covered}\} \right\} \\ &\quad = \prod_{j=1}^{d} \mathbf{P}\{\text{all 1's in class } D_j \text{ are covered}\} \\ &\quad = \prod_{j=1}^{d} f(D_j) \\ &\quad \le \prod_{j=1}^{d} \left(1 - p(1-p)^M\right)^{|D_j|} \\ &\quad = \left(1 - p(1-p)^M\right)^{\sum_{j=1}^{d} |D_j|} \\ &\quad = \left(1 - p(1-p)^M\right)^n, \end{aligned}$$

which is the same as the covering probability for the synchronous case. □

Lemma 3.7. *For frame asynchronous access, if* $p = \frac{1}{M+1}$

$$\mathbf{P}\{\text{false synchronization}\} \le \mathrm{e}^{M \ln T + M \ln n - \frac{n}{M+1}\mathrm{e}^{-1}}.$$

Proof. Let us select $M-1$ arbitrarily shifted code words, and another (tagged) code word which is also active, but with some shift. By Lemma 3.6 the probability that in a class of positions D_j all the 1's of the tagged code word are covered can be upper bounded by $\left(1-p(1-p)^M\right)^n$. As the classes are independent of each other, from (3.8) it follows that

$$\mathbf{P}\{\text{false synchronization}\} \leq \binom{T}{M-1}(T-M+1)n^M\left(1-p(1-p)^M\right)^n,$$

where the factor n^{M-1} is needed because of the shift of the code words. Let $p := \frac{1}{M+1}$, then

$$\begin{aligned}\mathbf{P}\{\text{false identification}\} &\leq \binom{T}{M-1}(T-M+1)n^M\left(1-\tfrac{1}{M+1}\left(1-\tfrac{1}{M+1}\right)^M\right)^n \\ &\leq T^M n^M\left(1-\tfrac{\mathrm{e}^{-1}}{M+1}\right)^n \\ &\leq T^M n^M \mathrm{e}^{-\frac{n}{M+1}\mathrm{e}^{-1}} \\ &= \mathrm{e}^{M\ln T+M\ln n-\frac{n}{M+1}\mathrm{e}^{-1}}. \end{aligned} \tag{3.9}$$

□

Proof of Theorem 3.5. If a randomly chosen code $\mathcal{C}$ which has T code words of length n satisfy the requirements of identification and synchronization, then $\mathcal{C}$ can be applied for T users in communication via a multiple-access OR channel. Obviously,

$$\mathbf{P}\{\mathcal{C}\text{ is bad}\} \leq \mathbf{P}\{\text{false identification}\} + \mathbf{P}\{\text{false synchronization}\}$$

and we need

$$\mathbf{P}\{\mathcal{C}\text{ is bad}\} < 1,$$

since then there is a good code. This gives an upper bound on minimum code length n. Thus, we need the following probabilities to tend to 0

$$\mathbf{P}\{\text{false identification}\} \rightarrow 0, \tag{3.10}$$

$$\mathbf{P}\{\text{false synchronization}\} \rightarrow 0. \tag{3.11}$$

If we choose $p = \frac{1}{M+1}$, and the code length n to

$$n = (1+\delta)\mathrm{e}\ln 2\,(M+1)^2\log T$$

for an arbitrary constant $\delta > 0$, the exponents in (3.10) and (3.11) become

$$-(M+1)\log T\left(\delta\left(1-\tfrac{\gamma}{M+1}\right)\ln 2-\left(1-\tfrac{1}{M+1}\right)\frac{\ln\left((1+\delta)e\ln 2(M+1)^2\log T\right)}{\log T}\right),$$

where constant $\gamma = 1$ and 2, respectively. Both exponents tend to $-\infty$ when $T \to \infty$, that is why we have (3.10) and (3.11).

As the reasoning above is true for all arbitrarily small $\delta > 0$, the following asymptotic upper bound on the minimum code length n has been shown

$$n_{\text{asyn}}(T, M) \lesssim \mathrm{e} \ln 2 \, (M+1)^2 \log T.$$

□

3.6 Binary cyclically permutable codes

In order to construct binary cyclically permutable codes, firstly we study the cyclic concatenations of two cyclically permutable codes. Assume two cyclically permutable codes $\mathcal{C}_{out}$ and $\mathcal{C}_{inn}$, and their cyclic concatenation results in a cyclically permutable code. The idea of cyclic concatenation dates back to Burton and Weldon (1965). They showed that the cyclic concatenation of two cyclic codes is cyclic.

The operator cyclic ordering can be formulated as follows: let $\mathbf{A}$ be an $m \times n$ array

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & \ddots & \vdots \\ a_{m-1,0} & \cdots & a_{m-1,n-1} \end{bmatrix}$$

and assume that

$$gcd(m, n) = 1.$$

The cyclic ordering of the array $\mathbf{A}$ is a map of $\mathbf{A}$ into an N-tuple

$$\mathbf{b} = (b_0, b_1, \ldots, b_{N-1})$$

such that

$$N = nm$$

and

$$b_i = a_{i \bmod m, i \bmod n}.$$

We show that because of $gcd(m, n) = 1$ this mapping is one-to-one. If there were $0 \leq i < i' \leq N-1$ such that

$$i \bmod m = i' \bmod m$$

and

$$i \bmod n = i' \bmod n,$$

then

$$m|i' - i$$

and

$$n|i' - i,$$

therefore because of $gcd(m, n) = 1$

$$mn|i' - i,$$

which is impossible since $0 < i' - i < N = nm$.

Let R denote the operator that shifts the columns of an $m \times n$ array cyclically one position rightwards, and let D denote the operator that shifts the rows cyclically one position downards.

Lemma 3.8. *(A, Györfi, Massey (1992)) If* $\mathbf{b}$ *is the cyclic ordering of* $\mathbf{A}$ *then*

$$RD(\mathbf{A}) = DR(\mathbf{A})$$

and $S(\mathbf{b})$ *is the cyclic ordering of* $RD(\mathbf{A})$*.*

Proof.

$$\begin{aligned} S(\mathbf{b})_i &= b_{i-1 \bmod mn} \\ &= a_{(i-1 \bmod mn) \bmod m, (i-1 \bmod mn) \bmod n} \\ &= a_{i-1 \bmod m, i-1 \bmod n}, \end{aligned}$$

which is the (i, i)-th element of $RD(\mathbf{A})$.

For the cyclic concatenation let $\mathcal{C}_{out}$ be the outer code with length n, alphabet F_{out} and cyclic distance d_{out}, and let $\mathcal{C}_{inn}$ be the inner code with length m, alphabet F_{inn} and cyclic distance d_{inn}. Assume that

$$|F_{out}| \leq |\mathcal{C}_{inn}|,$$

so the concatenation of the two codes is possible. Choose an arbitrary subcode $\mathcal{C}$ of $\mathcal{C}_{inn}$ with

$$|F_{out}| = |\mathcal{C}|.$$

Let

$$f : F_{out} \to \mathcal{C}$$

be an arbitrary one-to-one mapping.

The cyclic concatenation of $\mathcal{C}_{out}$ and $\mathcal{C}_{inn}$ consists of three steps:
i) Choose $\mathbf{c} \in \mathcal{C}_{out}$, ($\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$).
ii) Write $f(c_0), f(c_1), \ldots, f(c_{n-1})$ into an array $\mathbf{A}$ as columns:

$$\mathbf{A} = [f(c_0)^T, f(c_1)^T, \ldots, f(c_{n-1})^T].$$

iii) Generate the cyclic ordering $\mathbf{b}$ of $\mathbf{A}$. □

We say that $\mathbf{b}$ corresponds to $\mathbf{c}$. The set of these $\mathbf{b}$ vectors is denoted by $\mathcal{C}^*$, and called the cyclic concatenation of $\mathcal{C}_{out}$ and $\mathcal{C}_{inn}$.

If $\mathcal{C}_{inn}$ is binary then $\mathcal{C}^*$ is binary, too.

Theorem 3.6. *The code $\mathcal{C}^*$ is cyclically permutable with cyclic distance at least $d_{out}d_{inn}$, with length nm and*

$$|\mathcal{C}^*| = |\mathcal{C}_{out}|.$$

Proof. Let $\mathbf{b}, \mathbf{b}' \in \mathcal{C}^*$ be arbitrary with the correspondences

$$\mathbf{c} \to \mathbf{A} \to \mathbf{b}$$

and

$$\mathbf{c}' \to \mathbf{A}' \to \mathbf{b}'$$

$(\mathbf{c}, \mathbf{c}' \in \mathcal{C}_{out})$. Introduce the notations

$$\tau_1 = \tau \bmod n$$

and

$$\tau_2 = \tau \bmod m.$$

Then by Lemma 3.8

$$\begin{aligned} & d(\mathbf{b}, S^\tau \mathbf{b}') \\ = \;& d(\mathbf{A}, (DR)^\tau \mathbf{A}') \\ = \;& d(\mathbf{A}, D^\tau R^\tau \mathbf{A}') \\ = \;& d(\mathbf{A}, D^{\tau_2} R^{\tau_1} \mathbf{A}') \\ = \;& d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_0)^T, \ldots, S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_{n-1})^T]). \end{aligned}$$

Case a): Either $0 < \tau_1$ or $\mathbf{c} \neq \mathbf{c}'$.
Select the positions i for which $c_i \neq (S^{\tau_1}\mathbf{c}')_i$. The number of these positions is at least d_{out}. Then we lower bound

$$d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_0)^T, \ldots, S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_{n-1})^T])$$

by the Hamming distance of columns corresponding to these positions. Then because of $c_i \neq (S^{\tau_1}\mathbf{c}')_i$

$$d(f(c_i)^T, S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_i)^T) \geq d_{inn},$$

therefore

$$\begin{aligned} & d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_0)^T, \ldots, S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_{n-1})^T]) \\ \geq \;& d_{out}d_{inn}. \end{aligned}$$

Case b): $\tau_1 = 0$ and $\mathbf{c} = \mathbf{c}'$.
Then because of $\tau > 0$ therefore we have $\tau_2 > 0$ and

$$\begin{aligned} & d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_0)^T, \ldots, S^{\tau_2} f((S^{\tau_1}\mathbf{c}')_{n-1})^T]) \\ = \; & d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2} f(c_0)^T, \ldots, S^{\tau_2} f(c_{n-1})^T]) \\ \geq \; & n d_{inn} \\ \geq \; & d_{out} d_{inn}. \end{aligned}$$

□

For another construction, let $\mathcal{C}$ be a linear cyclic code with minimum distance d_{min}, with length n and with alphabet F_{out}. Assume an orthogonal decomposition

$$\mathcal{C} = \mathcal{C}' + \mathcal{C}'^{\perp}$$

such that both $\mathcal{C}'$ and $\mathcal{C}'^{\perp}$ are cyclic, and $\mathcal{C}'^{\perp}$ contains a codeword $\mathbf{c}^*$ which has N distinct cyclic shifts and contains the all 1 codeword $\mathbf{1}$. Put

$$\mathcal{C}_{out} = \mathcal{C}' + \mathbf{c}^*.$$

Then bacause of Theorem 3.4 $\mathcal{C}_{out}$ is cyclically permutable with cyclic distance

$$d_{\text{cyc}} \geq d_{\text{min}}$$

and

$$|\mathcal{C}_{out}| = |\mathcal{C}'|.$$

Assume, moreover, a cyclically permutable code $\tilde{\mathcal{C}}$ with cyclic distance d_{inn}, length m, alphabet F_{inn}. Let $\mathcal{C}_{inn}$ be the set of all cyclic shifts of all codewords in $\tilde{\mathcal{C}}$. Assume that

$$|F_{out}| \leq |\mathcal{C}_{inn}|$$

and let $f(a)$ be a one-to-one mapping of $|F_{out}|$ to a subcode of $\mathcal{C}_{inn}$ with the property

$$Sf(a) = f(a+1). \tag{3.12}$$

Let $\mathcal{C}^*$ be the cyclic concatenation of $\mathcal{C}_{out}$ and $\mathcal{C}_{inn}$. Again, if $\mathcal{C}_{inn}$ is binary then $\mathcal{C}^*$ is binary, too.

Theorem 3.7. *The code $\mathcal{C}^*$ is cyclically permutable with cyclic distance at least $d_{out}d_{inn}$, with length nm and*

$$|\mathcal{C}^*| = |\mathcal{C}_{out}|.$$

Proof. The proof is similar to the proof of Theorem 3.6. Let $\mathbf{b}, \mathbf{b}' \in \mathcal{C}^*$ be arbitrary with the correspondences

$$\mathbf{c} \to \mathbf{A} \to \mathbf{b}$$

and

$$\mathbf{c}' \to \mathbf{A}' \to \mathbf{b}'$$

$(\mathbf{c}, \mathbf{c}' \in \mathcal{C}_{out})$. Introduce the notations

$$\tau_1 = \tau \bmod n$$

and

$$\tau_2 = \tau \bmod m.$$

Then by Lemma 3.8

$$\begin{aligned} & d(\mathbf{b}, S^{\tau}\mathbf{b}') \\ = \;& d(\mathbf{A}, D^{\tau_2}R^{\tau_1}\mathbf{A}') \\ = \;& d([f(c_0)^T, \ldots, f(c_{n-1})^T], [S^{\tau_2}f((S^{\tau_1}\mathbf{c}')_0)^T, \ldots, S^{\tau_2}f((S^{\tau_1}\mathbf{c}')_{n-1})^T]) \\ = \;& d([f(c_0)^T, \ldots, f(c_{n-1})^T], [f((S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1})_0)^T, \ldots, f((S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1})_{n-1})^T]). \end{aligned}$$

If

$$[f(c_0)^T, \ldots, f(c_{n-1})^T]$$

and

$$[f((S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1})_0)^T, \ldots, f((S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1})_{n-1})^T]$$

differ in l columns then

$$d([f(c_0)^T, \ldots, f(c_{n-1})^T], [f((S^{\tau_1}\mathbf{c}'+\tau_2\mathbf{1})_0)^T, \ldots, f((S^{\tau_1}\mathbf{c}'+\tau_2\mathbf{1})_{n-1})^T]) \geq l d_{inn}.$$

We show that

$$l \geq d_{out}.$$

Obviously

$$l = d(\mathbf{c}, S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1}).$$

Put

$$\mathbf{c} = \mathbf{c}_1 + \mathbf{c}^*$$

and

$$\mathbf{c}' = \mathbf{c}_2 + \mathbf{c}^*$$

$(\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}')$, and

$$\begin{aligned} & d(\mathbf{c}, S^{\tau_1}\mathbf{c}' + \tau_2\mathbf{1}) \\ = \;& w(\mathbf{c} - S^{\tau_1}\mathbf{c}' - \tau_2\mathbf{1}) \\ = \;& w(\mathbf{c}_1 - S^{\tau_1}\mathbf{c}_2 + \mathbf{c}^* - S^{\tau_1}\mathbf{c}^* - \tau_2\mathbf{1}). \end{aligned}$$

Then

$$\mathbf{c}_1 - S^{\tau_1}\mathbf{c}_2 \in \mathcal{C}'$$

and

$$\mathbf{c}^* - S^{\tau_1}\mathbf{c}^* - \tau_2\mathbf{1} \in \mathcal{C}'^{\perp},$$

so it suffices to show that at least one of these two codewords is not $\mathbf{0}$.
Case a): $\tau > 0$.
The codewords $\mathbf{c}^* - S^{\tau_1}\mathbf{c}^*$ and $-\tau_2\mathbf{1}$ are linearly independent so their sum $\mathbf{c}^* - S^{\tau_1}\mathbf{c}^* - \tau_2\mathbf{1}$ can be $\mathbf{0}$ if and only if both $\mathbf{c}^* - S^{\tau_1}\mathbf{c}^*$ and $-\tau_2\mathbf{1}$ are $\mathbf{0}$, which is possible if and only if $\tau_1 = 0$ and $\tau_2 = 0$. However, this cannot happen because of $\tau > 0$.
Case b): $\tau = 0$ and $\mathbf{c}_1 \neq \mathbf{c}_2$.
Then

$$\mathbf{c}_1 - S^{\tau_1}\mathbf{c}_2 = \mathbf{c}_1 - \mathbf{c}_2 \neq \mathbf{0}.$$

□

3.7 Construction derived from a Reed-Solomon code for OR channel

Let's use the notations of Section 3.3. As an application of Theorem 3.7, assume that $L = p$ is a prime, and let $\mathcal{C}$ be a Reed-Solomon code over $GF(p)$ with parameters $(n = p-1, K)$. Using the notations of Theorem 3.7, we can get $\mathcal{C}_{out}$ if

$$\mathcal{C}' = \{\mathbf{c} = (0, 0, x_2, \ldots, x_{K-1})\mathbf{G}\}$$

and

$$\mathcal{C}'^{\perp} = \{\mathbf{c} = (x_0, x_1, 0, \ldots, 0)\mathbf{G}\}$$

and

$$\mathbf{c}^* = (0, 1, 0, \ldots, 0)\mathbf{G} = (1, \alpha, \alpha^2, \ldots, \alpha^{n-1}),$$

where $\mathbf{G}$ has been defined by (3.1). The vector $\mathbf{1}$ is the first row of $\mathbf{G}$, therefore the code

$$\mathcal{C}_{out} = \{\mathbf{c} = (0, 1, x_2, \ldots, x_{K-1})\mathbf{G}\}$$

satisfy the the conditions of Theorem 3.7, and for the user population size we have

$$T = q^{K-2}.$$

Put

$$\mathcal{C}_{inn} = GF(p),$$

and let $f(a)$ be a binary vector of length $m = p$ having 1 only at position a. p is prime, therefore the map $f(a)$ has the property (3.12). Thus, we get a binary code of length

$$N = nm = (p-1)p,$$

and Theorem 3.7 implies that

$$d_{\mathrm{cyc}} \geq d_{out}d_{inn} = (n-K+1)2 = (p-K)2,$$

therefore

$$c_{\mathrm{cyc}} = N - d_{\mathrm{cyc}} \leq (p-1)p - (p-K)2.$$

This code is due to A, Györfi, Massey (1992).

3.8 Construction derived from a BCH code for OR channel

Apply the notations of Section 3.4. For a prime p, let $\mathcal{C}$ be a BCH code over $GF(p)$ of length

$$n = p^r - 1$$

defined by the parity check polynomial

$$h(x) = l.c.m.\{M_0(x), M_1(x), M_2(x), \ldots, M_{K-1}\},$$

where $3 \leq K < p-1$ and if α is a primitive element of GF(p^r), then $M_i(x)$ denotes the minimal polynomial of α^i over $GF(p)$. Because of $3 \leq K < p-1$ and $M_0(x) = x-1$.

$$h(x) = (x-1)\prod_{j=1}^{K-1} M_j(x).$$

Consider two other BCH codes with the same length. The code $\mathcal{C}'$ has the parity check polynomial

$$h_0(x) = \prod_{j=2}^{K-1} M_j(x),$$

while the code $\mathcal{C}'^{\perp}$ has the parity check polynomial

$$h_1(x) = M_0(x)M_1(x) = (x-1)M_1(x).$$

Then in Section 3.4 we proved that

$$\mathcal{C} = \mathcal{C}' + \mathcal{C}'^{\perp},$$

$\mathcal{C}'^{\perp}$ contains the vector $\mathbf{1}$ and a vector $\mathbf{c}^*$ such that the cyclic shifts of $\mathbf{c}^*$ are all different. Define $\mathcal{C}_{out}$ by

$$\mathcal{C}_{out} = \mathcal{C}' + \mathbf{c}^*.$$

It is easy to see that the population size is

$$T = |\mathcal{C}^*| = p^{(K-2)r}.$$

Theorem 3.4 and (A.13) imply that

$$d_{out} \geq d_{min} \geq p^r - 1 - (K-1)p^{r-1}.$$

Let $\mathcal{C}_{inn}$ and $f(a)$ be as in Section 3.7, then we get a binary code of length

$$N = nm = (p^r - 1)p,$$

and Theorem 3.7 implies that

$$d_{\mathrm{cyc}} \geq d_{out}d_{inn} = (p^r - 1 - (K-1)p^{r-1})2,$$

therefore

$$c_{cyc} = N - d_{cyc} \leq (p^r - 1)p - (p^r - 1 - (K-1)p^{r-1})2.$$

This code is introduced in Győrfi, Vajda (1993).

Chapter 4

Collision channel

4.1 Channel model

The concept of collision channel has been introduced by Massey and Mathys (1985). A T user multiple access collision channel is a deterministic channel without feedback which has T inputs $(x_i, i \in [T])$ and one output (y). The traffic to send over this common channel is in the form of packets that are assumed to take values from the input alphabet I. Each user can send an arbitrary packet from the input alphabet I into the channel or if a user wants to be silent, then he formally sends the $\emptyset$ symbol. The output of the channel can be $\emptyset$ if all users were silent, an element of I if exactly one user sent this element and the others were silent, and the so called erasure (collision) symbol $*$ otherwise. The time axis is assumed to be partitioned into intervals called slots (slotted channel) whose duration corresponds to the transmission time for one packet. There is a longer unit called frame or block which consists of n slots. In Section 4.3 frame synchronization is assumed, so frames of users begin at the same slots (no time shift), while in Sections 4.2 and 4.4 we study the frame asynchronous case.

There is no feedback available to inform the senders of the channel outputs in previous slots. If the user population is finite (T), then the coding can be done by a finite set of protocol sequences assigned in a one-to-one manner to the users. Each user, e.g., the i^{th} user has a *protocol sequence* q_i which is a binary sequence of length n that controls his sending of packets in the following way. When user i becomes active—after some time of inactivity—he can send a packet in the j^{th} slot of this activity frame $(1 \leq j \leq n)$ if q_i has a 1 in the j^{th} position, and otherwise he must be silent in this slot. He continues to use his protocol sequence periodically in this manner, until he has no more packets to send, when he again becomes inactive. If q_i has

Hamming weight $w(q_i)$, then user i will send $w(q_i)$ packets in each frame of length n slots where he is active. The protocol sequences can be considered as an outer code.

Let A be the set of messages of user i and suppose that $|A| = S$. User i encodes each message $a_j \in A$ into a code word $c_j^{(i)} \in \mathcal{C}^{(i)}$ of length $w(q_i)$ ($j \in [S]$, $i \in [T]$). The components of $c_j^{(i)}$ are sent according to the protocol sequence q_i. $\mathcal{C}^{(i)}$ is called the code of user i. If the protocol sequences have the same weight, then $\mathcal{C}^{(i)}$ can be the same for all users. Because of collisions, some packets are erased during the transmissions, and these erasure errors are corrected using $\mathcal{C}^{(i)}$. $\mathcal{C}^{(i)}$ is the *inner code.*

If all the T users were active all the time, then—for synchronous access—the time sharing would be the best solution for them (for large T) which is not interesting in this case. Let us suppose that at most M users would like to communicate simultaneously ($2 \leq M \ll T$). Our task is to choose codes $\mathcal{C}^{(i)}$ and protocol sequences q_i such that from the output of the channel it can be determined which users were active (identification), where their code words begin (synchronization) and what they sent (decoding). We are looking for the minimum frame size $n = n(T, M, S)$ which still ensures these requirements.

Massey and Mathys (1985)) constructed an optimal code with $n = M^M$ for $T = M$ and asynchronous access. Tsybakov and Likhanov (1983) extended it for $M < T$ with $n = T^M$. A, Györfi, Massey (1992) and Györfi, Vajda (1993) presented cyclically permutable codes for collision channel.

On the minimum frame size n we derive lower and upper bounds asymptotically with the following conditions: $T \to \infty$, $S \to \infty$ and $\frac{\log T}{\log S} \to 0$. We denote by $\gtrsim$ and $\lesssim$ lower and upper bounds, respectively, which holds asymptotically in case of some given conditions.

In this chapter we show that both for synchronous and asynchronous access, the best possible throughput is e^{-1}, and it can be achieved using Reed–Solomon code as an inner code. Concerning the protocol sequences (outer code), the rates of the existing constructions (A, Györfi, Massey (1992) and Györfi, Vajda (1993)) are far from e^{-1}.

4.2 Bounds for binary packets

In this section multiple access collision channel is considered based on Bassalygo and Pinsker (1983). For the sake of simplicity assume that the input alphabet has only 2 elements, i.e., we have binary packets, $I = \{a, b\}$.

Theorem 4.1 (Bassalygo and Pinsker (1983)). *For fixed M, if $T \to \infty$ and $S \to \infty$ then*

$$n(T, M, S) \gtrsim \max \left\{ M \left(1 - \frac{1}{M}\right)^{1-M} \log S,\ \frac{1}{2} M \log S + \frac{1}{2} M \log T \right\}$$

Before proving the lower bound we need a lemma.

Definition 4.1 (Erasure sum of vectors). *Erasure sum of binary vectors of length n is a binary vector of length n which has in the i^{th} position 0 iff all vectors have 0 in this position, 1 iff exactly one vector has 1 in this position and the others have 0, and erasure symbol $*$ otherwise. We denote the erasure sum of $q_1, \ldots, q_k$ by $q_1 \boxplus \cdots \boxplus q_k$.*

Let us compose a matrix G from the protocol sequences $q_1, \ldots, q_T$ as rows, and let w_k be the number of 1's standing in the k^{th} column of G.

Lemma 4.1 (Bassalygo and Pinsker (1983)). *The sum of the number of 1's in all possible erasure sum vectors $q_{i_1} \boxplus \cdots \boxplus q_{i_M}$ is*

$$\sum_{k=1}^{n} w_k \binom{T - w_k}{M - 1},$$

where $\{i_1, \ldots, i_M\} \subseteq [T]$.

Proof. The k^{th} position of the erasure sum vector $q_{i_1} \boxplus \cdots \boxplus q_{i_M}$ is 1 iff exactly one of its component protocol sequences has 1 in the k^{th} position and the other $M - 1$ have 0 there. There are w_k protocol sequences having 1 in the k^{th} position and $T - w_k$ having 0 there, so such an erasure sum vector can be constructed as $w_k \binom{T-w_k}{M-1}$ different sums. The statement of the lemma is given if we add these quantities for all positions. □

Proof of Theorem 4.1. Firstly, we show that

$$n(T, M, S) \gtrsim \frac{1}{2} M \log S + \frac{1}{2} M \log T.$$

From the channel output it should be determined which M users out of T were active ($\binom{T}{M}$ possible sort) and what the active users sent (S^M possible sort). Because of the channel output symbols can be 4 different kind ($\emptyset$ symbol, two information bits a and b, and erasure symbol $*$), the following must stand

$$4^n \geq \binom{T}{M} S^M,$$

and from this by taking the logarithm of both sides

$$\begin{aligned} n &\geq \frac{1}{2}M\log S + \frac{1}{2}\log\binom{T}{M} \\ &= \frac{1}{2}M\log S + \frac{1}{2}\log\frac{T(T-1)\cdots(T-M+1)}{M(M-1)\cdots 1} \\ &\geq \frac{1}{2}M\log S + \frac{1}{2}M\log\frac{T}{M} \\ &= \frac{1}{2}M\log S + \frac{1}{2}M\log T - \frac{1}{2}M\log M \\ &\simeq \frac{1}{2}M\log S + \frac{1}{2}M\log T. \end{aligned}$$

In the last step we used the conditions of the theorem.

Secondly, we derive that

$$n(T,M,S) \gtrsim M\left(1-\frac{1}{M}\right)^{1-M}\log S.$$

There are $\binom{T}{M}$ different erasure sum consisting of M vectors. From Lemma 4.1 follows that there are rows $i_1,\dots,i_M$ of G such that the erasure sum $q_{i_1}\boxplus\cdots\boxplus q_{i_M}$ of them contains at most

$$\frac{\sum_{k=1}^{n} w_k\binom{T-w_k}{M-1}}{\binom{T}{M}}.$$

1's. Consider users which correspond to rows $i_1,\dots,i_M$. Each of them can send S different messages, independently of the others. In the output vector of the channel just those positions can hold information about messages sent where the erasure sum of the protocol sequences has 1. So that all the messages can be decoded the following necessary condition must be fulfilled

$$\frac{\sum_{k=1}^{n} w_k\binom{T-w_k}{M-1}}{\binom{T}{M}} \geq \log S^M = M\log S. \tag{4.1}$$

In the following we give upper bound to the left side of inequality (4.1) which results the needed lower bound on $n(T,M,S)$. For doing this we are looking for the maximum value of $w\binom{T-w}{M-1}$.

$w\binom{T-w}{M-1}$ is increasing if

$$\begin{aligned}\frac{(w+1)\binom{T-(w+1)}{M-1}}{w\binom{T-w}{M-1}} &= \frac{(w+1)(T-w-1)!(T-w-M+1)!(M-1)!}{w(T-w)!(T-w-M)!(M-1)!}\\ &= \frac{(w+1)(T-w-M+1)}{w(T-w)}\\ &\geq 1,\end{aligned}$$

which is equivalent to

$$\begin{aligned}(w+1)(T-w-M+1) &\geq w(T-w)\\ wT-w^2-wM+w+T-w-M+1 &\geq wT-w^2\\ \frac{T+1}{M}-1 &\geq w.\end{aligned}$$

So $w\binom{T-w}{M-1}$ has its maximum at $w=\left\lfloor\frac{T+1}{M}-1\right\rfloor$ or $w=\left\lfloor\frac{T+1}{M}\right\rfloor$, but asymptotically does not matter which one we choose.

In inequality (4.1) replace each term of the sum by its maximum value

$$\frac{n\left\lfloor\frac{T+1}{M}\right\rfloor\binom{T-\left\lfloor\frac{T+1}{M}\right\rfloor}{M-1}}{\binom{T}{M}} \geq \frac{\sum_{k=1}^{n} w_k\binom{T-w_k}{M-1}}{\binom{T}{M}} \geq M\log S,$$

so we get lower bound on n

$$\begin{aligned}n &\geq M\log S\cdot\frac{\binom{T}{M}}{\left\lfloor\frac{T+1}{M}\right\rfloor\binom{T-\left\lfloor\frac{T+1}{M}\right\rfloor}{M-1}}\\ &\geq M\log S\cdot\frac{\binom{T}{M}}{\frac{T+1}{M}\binom{T-\left\lfloor\frac{T+1}{M}\right\rfloor}{M-1}}\\ &= M\log S\cdot\frac{T(T-1)\cdots(T-M+1)}{(T+1)\left(T-\left\lfloor\frac{T+1}{M}\right\rfloor\right)\left(T-\left\lfloor\frac{T+1}{M}\right\rfloor-1\right)\cdots\left(T-\left\lfloor\frac{T+1}{M}\right\rfloor-M+2\right)}\\ &\geq M\log S\cdot\frac{T-M+1}{T+1}\left(\frac{T}{T-\left\lfloor\frac{T+1}{M}\right\rfloor}\right)^{M-1}\\ &\geq M\log S\cdot\left(1-\frac{M}{T+1}\right)\left(\frac{T}{T-\frac{T}{M}+1}\right)^{M-1}\\ &= M\log S\cdot\left(1-\frac{M}{T+1}\right)\left(1-\frac{1}{M}+\frac{1}{T}\right)^{1-M}\end{aligned}$$

$$\simeq \quad M \log S \cdot \left(1 - \frac{1}{M}\right)^{1-M},$$

where in the last step we used that $\frac{M}{T+1} \to 0$.

Theorem 4.2 (Bassalygo and Pinsker (1983)). *For fixed M, there exists T_0 and S_0 such that if $T \geq T_0$ and $S \geq S_0$, then*

$$n(T, M, S) \leq M \left(1 - \frac{1}{M}\right)^{1-M} (\log S + M \log T)\,(1 + \alpha) \left(1 + \sqrt{3\alpha}\right),$$

where

$$\alpha = \frac{M \ln T}{\log S + (M-1) \log T} \leq 10 + o(1).$$

For getting the upper bound on the minimum block length $n(T, M, S)$, the random coding method will be used.

Definition 4.2 (information segment). *The positions of code words of a code C, where every code words differ, form an information segment with respect to C.*

Let us denote by $G_i(i_1, \ldots, i_{M-1})$ the positions of row vectors of G where both q_i and erasure sum $q_i \boxplus q_{i_1} \boxplus \cdots \boxplus q_{i_{M-1}}$ have 1's ($i, i_1, \ldots, i_{M-1} \subseteq [T]$). It is easy to see that if users $i, i_1, \ldots, i_{M-1}$ are active in the channel, then the output vector of the channel has information about the message of the i^{th} user just on positions $G_i(i_1, \ldots, i_{M-1})$, so these positions have to form an information segment with respect to $C^{(i)}$. Hence, component codes $C^{(i)}$ have to be chosen such that $G_i(i_1, \ldots, i_{M-1})$ do form information segment for all possible $i, i_1, \ldots, i_{M-1} \in [T]$. On construction of such codes the following lemma gives sufficient condition.

Lemma 4.2 (Bassalygo and Pinsker (1983)). *Let $A_1, \ldots, A_N \subseteq [w]$. If $|A_j| \geq \log S + \log N$ ($\forall j \in [N]$), then there exists a binary code of length w and size S such that the positions indicated by A_j form information segment with respect to the code for every $j \in [w]$.*

Proof. The following greedy type method gives an appropriate code. Let us consider the set of binary vectors of length w. It has 2^w elements and we choose the code words from it. The first code word can be an arbitrary element of the set. Throw out those vectors from the set which do not differ from the previously selected code word in positions indicated by any A_j. From the remaining vectors of the set also choose an arbitrary element. Continue with these steps, so throw out those vectors from the set which do not differ from any of the previously selected code words in positions indicated by any A_j, and choose an arbitrary element from the remaining vectors of the set. Algorithm can run until the set does not get empty. In each step the newly chosen code word needs to throw out at most $\sum_{j=1}^{N} 2^{w-|A_j|}$ vectors from the set. Initially, the set has 2^w binary vectors, so this algorithm can run at least to the $(S-1)^{\text{th}}$ step (throwing out) while producing S code words, because

$$(S-1)\cdot\sum_{j=1}^{N} 2^{w-|A_j|} \le (S-1)\cdot N\cdot 2^{w-\log S-\log N} = \frac{S-1}{S}\cdot 2^w \le 2^w.$$

That is why the algorithm can produce at least S code words. □

In the proof of Theorem 4.2 we apply some lemmas and we will use the following two conditions on matrix G. Let us define the ZFD property as $|G_i(i_1,\dots,i_M)| \ge 1$ for all rows q_i and $q_{i_1},\dots,q_{i_M}$ ($q_i \notin \{q_{i_1},\dots,q_{i_M}\}$). If the matrix G has the ZFD property, then the set of active users can be determined from the output vector of the channel (identification). Moreover, $G_i(i_1,\dots,i_{M-1})$ have to form information segment with respect to $\mathcal{C}^{(i)}$ for all $i, i_1,\dots,i_{M-1} \subseteq [T]$, because then the message sent by user i can be decoded. If we apply Lemma 4.2 with the number of code words S and number of possible information segments with respect to $\mathcal{C}^{(i)}$

$$N = \binom{T-1}{M-1} \le T^{M-1}$$

we get

$$|G_i(i_1,\dots,i_{M-1})| \ge \log S + (M-1)\log T$$

(decodable property).

Let us take a $T\times n$ 0-1 matrix G whose elements are chosen randomly independently of each other and the probability of an element being 1 is $\frac{1}{M}$. We denote by probabilities $\mathbf{P}\{\text{ZFD}\}$ and $\mathbf{P}\{\text{decodable}\}$ that the code represented by the random matrix is ZFD and decodable, respectively.

Lemma 4.3. *If* $n \ge \frac{k_0}{p}$ *then*

$$\mathbf{P}\{\text{not decodable}\} \le \exp\left(-\frac{(np-k_0)^2}{3np} + M\ln T\right),$$

where

$$p = \frac{\left(1-\frac{1}{M}\right)^{M-1}}{M}$$

and

$$k_0 = \log S + (M-1)\log T.$$

Proof. Let us select M rows of the random matrix G and call one of them tagged row. Let p be the probability of the event that a fixed position of the tagged row is 1 while the other $M-1$ rows have 0 there (i.e., the tagged row has an uncovered 1 in a fixed position), thus

$$p = \frac{\left(1-\frac{1}{M}\right)^{M-1}}{M}.$$

Then the probability that there are exactly k positions where the tagged row has uncovered 1's and the other $n-k$ positions are covered by the other $M-1$ rows is at most $\binom{n}{k}p^k(1-p)^{n-k}$. The probability that the tagged row has at most k_0 uncovered positions is at most

$$\sum_{k<k_0}\binom{n}{k}p^k(1-p)^{n-k}.$$

The probability that there is a row q_i which has less than k_0 uncovered positions (other positions are covered by rows $q_{i_1},\ldots,q_{i_{M-1}}$) is at most

$$\begin{aligned}
&\mathbf{P}\{\text{not decodable}\}\\
&\quad= \mathbf{P}\{\exists i, i_1, i_{M-1}\in[T] : |G_i(i_1,\ldots,i_{M-1})| < \log S + (M-1)\log T\}\\
&\quad= \mathbf{P}\{\exists i, i_1, i_{M-1}\in[T] : |G_i(i_1,\ldots,i_{M-1})| < k_0\}\\
&\quad\le T\binom{T-1}{M-1}\sum_{k<k_0}\binom{n}{k}p^k(1-p)^{n-k}.
\end{aligned}$$

Let us apply now Bernstein's inequality for upper bounding the tail of binomially distributed random variable (which is the sum of indicator variables)

$$\begin{aligned}
\mathbf{P}\{\text{not decodable}\} &\le T\binom{T-1}{M-1}\sum_{k<k_0}\binom{n}{k}p^k(1-p)^{n-k}\\
&\le \exp\left(-\frac{n\left(p-\frac{k_0}{n}\right)^2}{2p(1-p)+\frac{2}{3}\left(p-\frac{k_0}{n}\right)}+M\ln T\right)\\
&\le \exp\left(-\frac{n\left(p-\frac{k_0}{n}\right)^2}{2p+p}+M\ln T\right)\\
&= \exp\left(-\frac{(np-k_0)^2}{3np}+M\ln T\right).
\end{aligned}$$

□

Lemma 4.4.

$$\mathbf{P}\{\text{not ZFD}\} \leq \exp\left((M+1)\ln T - n\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)$$

Proof. The proof is similar to Lemma 4.3. Let us fix M different rows in matrix G and choose an $(M+1)^{\text{th}}$ (tagged) row distinctly from the others. The probability that one of the positions of the tagged row is covered by the erasure sum of the other M rows is $1-\frac{1}{M}\left(1-\frac{1}{M}\right)^M$. The probability that all positions of the tagged row are covered by the other rows is $\left(1-\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)^n$, then the probability that there exists a row q_i such that all positions of it are covered by another M rows $(q_{i_1},\dots,q_{i_M})$ is at most

$$\begin{aligned}
\mathbf{P}\{\text{not ZFD}\} &\leq T\binom{T}{M}\left(1-\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)^n \\
&\leq T^{M+1}\left(1-\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)^n \\
&= \exp\left((M+1)\ln T + n\ln\left(1-\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)\right) \\
&\leq \exp\left((M+1)\ln T - n\frac{1}{M}\left(1-\frac{1}{M}\right)^M\right)
\end{aligned}$$

where in the last step we applied $\ln(1-x) \leq -x$, $\forall x \in [0,1]$. □

Proof of Theorem 4.2. If a random 0-1 matrix G of size $T \times n$ is ZFD and decodable, then its rows can be used as protocol sequences for T users in communication via a multiple access collision channel. This gives an upper bound on minimum frame size n. Thus, we need the following

$$\mathbf{P}\{\text{not ZFD}\} < 1 \quad \text{and} \quad \mathbf{P}\{\text{not decodable}\} < 1.$$

For the ZFD property

$$\mathbf{P}\{\text{not ZFD}\} \leq \exp\left((M+1)\ln T - \frac{n}{M}\left(1-\frac{1}{M}\right)^M\right) < 1,$$

so we need

$$\begin{aligned}
(M+1)\ln T &< \frac{n}{M}\left(1-\frac{1}{M}\right)^M \\
M(M+1)\left(1-\frac{1}{M}\right)^{-M}\ln T &< n. \qquad (4.2)
\end{aligned}$$

For the decodable property

$$\mathbf{P}\{\text{not decodable}\} \leq \exp\left(-\frac{(np-k_0)^2}{3np} + M\ln T\right) < 1.$$

By taking the logarithm of both sides we get

$$\frac{(np-k_0)^2}{3np} - M\ln T > 0.$$

The solution of this inequality with respect to positive length n is

$$n > \frac{k_0}{p}\left(1+\frac{3}{2}\frac{M\ln T}{k_0}\right)\left(1+\sqrt{1-\frac{1}{\left(1+\frac{3}{2}\frac{M\ln T}{k_0}\right)^2}}\right).$$

Let us introduce

$$\alpha = \frac{M\ln T}{k_0} = \frac{M\ln T}{\log S + (M-1)\log T},$$

then we get a simpler inequality on n

$$\begin{aligned} n &> \frac{k_0}{p}\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{1-\frac{1}{(1+\frac{3}{2}\alpha)^2}}\right) \\ &= \frac{\log S + (M-1)\log T}{p}\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{1-\frac{1}{(1+\frac{3}{2}\alpha)^2}}\right). \qquad (4.3) \end{aligned}$$

From inequalities (4.2) and (4.3) the latter gives the stronger restriction on n. If it is fulfilled then there exists an appropriate random protocol sequence set for T users, so the minimum frame size is upper bounded

$$\begin{aligned} n(T,M,S) &\leq \frac{\log S + (M-1)\log T}{p}\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{1-\frac{1}{(1+\frac{3}{2}\alpha)^2}}\right) \\ &\leq \frac{\log S + (M-1)\log T}{p}\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{3\alpha}\right) \qquad (4.4) \end{aligned}$$

Factor α can be bounded if $M \geq 2$ in the following way

$$\begin{aligned} \alpha &= \frac{M\ln T}{\log S + (M-1)\log T} \\ &= \frac{M}{\frac{\ln S}{\ln T} + M - 1}\ln 2 \\ &\leq \frac{1}{1-\frac{1}{M}}\ln 2 \\ &\leq 2\ln 2, \end{aligned}$$

so the constant factors of (4.4) can be bounded

$$\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{3\alpha}\right) \le (1+3\ln 2)\left(1+\sqrt{6\ln 2}\right) < 10.$$

Let us introduce the sum-rate

$$R_{\text{sum}} = \frac{M\log_{|I|} S}{n}$$

of communication, as usually, and denote by $R_{\text{sum}}(T,M,n)$ the maximum sum-rate for parameters T, M, n.

Theorem 4.3 (Bassalygo and Pinsker (1983)). *If M is fixed, $T \to \infty, S \to \infty$ and $\frac{M\log T}{\log S} \to 0$, then*

$$R_{sum}(T,M,n) \simeq \left(1-\frac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$R_{sum}(T,M,n) \simeq e^{-1}.$$

Proof. From Theorems 4.1 and 4.2 follows

$$\begin{aligned} M\left(1-\frac{1}{M}\right)^{1-M}\log S &\lesssim n(T,M,S) \\ &\lesssim M\left(1-\frac{1}{M}\right)^{1-M}(\log S + M\log T)\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{3\alpha}\right), \quad (4.5) \end{aligned}$$

where

$$\alpha = \frac{M\ln T}{\log S + (M-1)\log T} = \frac{\ln 2}{\frac{\log S}{M\log T} + 1 - \frac{1}{M}}.$$

As $\frac{M\log T}{\log S} \to 0$, $\alpha \to 0$, that is why

$$\left(1+\frac{3}{2}\alpha\right)\left(1+\sqrt{3\alpha}\right) \to 1.$$

Thus, from (4.5) we get

$$\left(1-\frac{1}{M}\right)^{M-1}\frac{1}{1+\frac{M\log T}{\log S}} \lesssim R_{\text{sum}}(T,M,n) \lesssim \left(1-\frac{1}{M}\right)^{M-1},$$

and so

$$\left(1-\frac{1}{M}\right)^{M-1} \lesssim R_{\text{sum}}(T,M,n) \lesssim \left(1-\frac{1}{M}\right)^{M-1}, \tag{4.6}$$

therefore

$$R_{\text{sum}}(T,M,n) \simeq \left(1-\frac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then $\left(1-\frac{1}{M}\right)^{M-1} \to \mathrm{e}^{-1}$, so

$$R_{\text{sum}}(T,M,n) \simeq \mathrm{e}^{-1}.$$

□

4.3 Bounds for non-binary packets

We consider now the case when the input alphabet I contains more than two elements. Here k information packets are encoded, so $S = |I|^k$, therefore the sum-rate is defined as

$$R_{\text{sum}} = \frac{kM}{n}.$$

Theorem 4.4 (Györfi and Győri (2004)). *For non-binary packets, if M is fixed, $T \to \infty, |I| \to \infty$, and $\frac{\log T}{\log |I|} \to 0$, then*

$$n(T,M,k) \gtrsim kM\left(1-\tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \lesssim \left(1-\tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T,M,k) \gtrsim kM\mathrm{e},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \lesssim \mathrm{e}^{-1}.$$

Proof. For the minimum code length, entropy based lower bound is given. For a deterministic channel, the entropy of the channel input block can not be greater than the entropy of the output block of the channel. If the codes can solve the tasks of identification and decoding, then the entropy of the output block is equal to the entropy of the input block. If M users out of T send packets into the channel and each message takes values from a set of size $S = |I|^k$, then the input random variable can take $\binom{T}{M}(|I|^k)^M$ different values. (Note, that minimum block length needed for *at most* M users is greater or equal to the minimum block length needed for *exactly* M users, therefore it is enough to consider here the latter scenario.) Assume that the input variable is uniformly distributed, so the entropy of the input random variable is $\log\binom{T}{M}|I|^{kM}$. For the output random variable we can apply the sum of the componentwise entropy as upper bound. All together we have the following inequality

$$\log\left(\binom{T}{M}|I|^{kM}\right) = H(O_1,\dots,O_n) \le \sum_{i=1}^{n} H(O_i) \le n \max_i H(O_i),$$

where O_i corresponds to the i^{th} position of the output. Let w_i be the number of protocol sequences which have 1's at the i^{th} position. The entropy $H(O_i)$ is the highest possible if O_i is uniformly distributed on all $a \in I$. In this case the distribution of O_i can be calculated in the following way:

$$\begin{aligned}
\mathbf{P}\{O_i = \emptyset\} &= \frac{\binom{T-w_i}{M}}{\binom{T}{M}} := p_0 \\
\mathbf{P}\{O_i = a\} &= \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}|I|} := \frac{p_1}{|I|}, \quad \forall a \in I \\
\mathbf{P}\{O_i = *\} &= 1 - \frac{\binom{T-w_i}{M}}{\binom{T}{M}} - \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}} = 1 - p_0 - p_1.
\end{aligned}$$

The entropy of O_i can be upper bounded as

$$\begin{aligned}
H(O_i) &\le -p_0\log p_0 - |I|\cdot\frac{p_1}{|I|}\log\frac{p_1}{|I|} - (1-p_0-p_1)\log(1-p_0-p_1) \\
&= -p_0\log p_0 - p_1\log p_1 - (1-p_0-p_1)\log(1-p_0-p_1) + p_1\log|I| \\
&\le \log 3 + p_1\log|I| \\
&= \log 3 + \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}}\log|I| \\
&= \log 3 + \frac{w_i M}{T-M+1}\cdot\frac{(T-w_i)\cdots(T-w_i-M+2)}{T\cdots(T-M+2)}\log|I| \\
&\le \log 3 + \frac{w_i M}{T}\frac{1}{1-\frac{M-1}{T}}\left(1-\frac{w_i}{T}\right)^{M-1}\log|I| \\
&\le \log 3 + \frac{1}{1-\frac{M-1}{T}}\left(1-\frac{1}{M}\right)^{M-1}\log|I|
\end{aligned}$$

$$\simeq \quad \log 3 + \left(1 - \frac{1}{M}\right)^{M-1} \log |I|,$$

where we used that $\frac{w_i M}{T}\left(1 - \frac{w_i}{T}\right)^{M-1}$ takes its maximum at $w_i = \frac{T}{M}$. The calculation above implies that

$$\log\left(\binom{T}{M}|I|^{kM}\right) \le n\left(\left(1 - \tfrac{1}{M}\right)^{M-1}\log|I| + \log 3\right).$$

For the minimum code length we get (by using $\left(\frac{T}{M}\right)^M \le \binom{T}{M}$)

$$\begin{aligned} n &\gtrsim \frac{M\log\frac{T}{M} + kM\log|I|}{\left(1-\frac{1}{M}\right)^{M-1}\log|I| + \log 3} \\ &= \frac{\frac{M\log\frac{T}{M}}{\log|I|} + kM}{\left(1-\frac{1}{M}\right)^{M-1} + \frac{\log 3}{\log|I|}}. \end{aligned}$$

If $|I| \to \infty$ and $\frac{\log T}{\log |I|} \to 0$, then

$$n(T, M, k) \gtrsim kM\left(1 - \tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T, M) = \frac{kM}{n} \lesssim \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T, M, k) \gtrsim kM\mathrm{e},$$

and for the sum-rate

$$R_{\text{sum}}(T, M) = \frac{kM}{n} \lesssim \mathrm{e}^{-1}.$$

□

In order to get an upper bound on the minimum block length $n(T, M, k)$, randomly chosen protocol sequences of constant weight w are used, and as an inner code $\mathcal{C}_i = \mathcal{C}$ a Reed–Solomon code of parameters (w, k) is applied over GF$(|I|)$ $(w \le |I|)$. (Remember, that each user has a binary vector of length n called protocol sequence which has a 1 in those positions where the user can send a packet.) Each active user can send w packets in each frame, that is why the code length should be w. If there is a collision in a time slot,

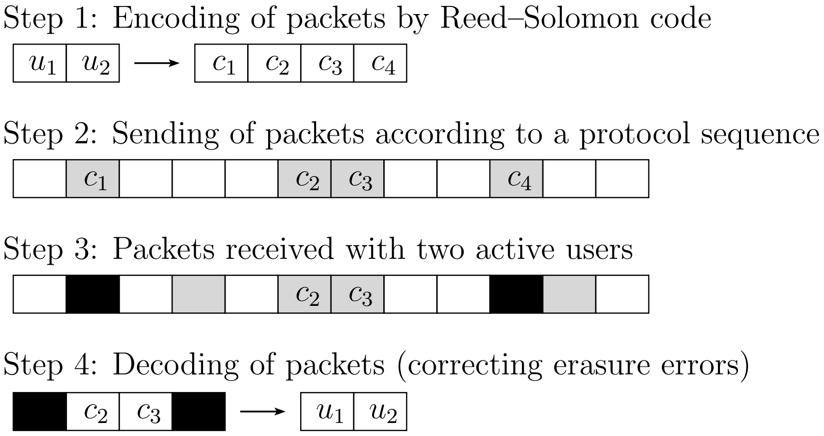
Step 1: Encoding of packets by Reed–Solomon code
u_1 u_2 ⟶ c_1 c_2 c_3 c_4
Step 2: Sending of packets according to a protocol sequence
c_1 c_2 c_3 c_4
Step 3: Packets received with two active users
c_2 c_3
Step 4: Decoding of packets (correcting erasure errors)
c_2 c_3 ⟶ u_1 u_2

Figure 4.1: Packet communication scheme on a collision channel

the output of the channel is the erasure symbol $*$, so the erroneous positions are known. A Reed–Solomon code of parameters (w, k) can correct up to $w - k$ erasure error.

In Figure 4.1 the communications scheme is illustrated in the viewpoint of a tagged user. Let us suppose that the inner code is a Reed–Solomon code of parameters $(w, k) = (4, 2)$, and each user has a protocol sequence of length $n = 12$. In the first step the user encodes its message packets (u_1, u_2) into the code packets (c_1, c_2, c_3, c_4) by the Reed–Solomon code. If the user has the protocol sequence 010001100100, then in the second step it sends the encoded packets into the channel according to this protocol sequence. In the figure the time slots where the tagged user can send a packet, i.e., the protocol sequence has 1's, are light gray shadowed, while empty slots are white boxes. Packets of the other active users may erase some of the packets of the tagged user which are represented by black boxes. In the last step the message packets can be decoded if there are at least $k = 2$ successfully received packets.

Theorem 4.5 (Györfi and Győri (2004)). *For synchronous access and non-binary packets, if M is fixed, $T \to \infty$, $k \to \infty$, $\frac{\log T}{k} \to 0$ and $|I| > \mathrm{e}k$, then*

$$n(T, M, k) \lesssim kM \left(1 - \tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T, M) \gtrsim \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T, M, k) \lesssim kM\mathrm{e},$$

and for the sum-rate

$$R_{sum}(T, M) \gtrsim \mathrm{e}^{-1}.$$

The coding method has to ensure the identification and decoding. The latter depends on the codes of the users, while the first one on the protocol sequences of the users.

The detection is done by a two phase algorithm. In the first step in a given block the successfully transmitted packets and the collision symbol $*$ on the output of the channel are transformed to a bit 1, and the $\emptyset$ symbol to bit 0. The resulting binary vector is actually the Boolean sum of the protocol sequences of the active users. If this binary vector covers the protocol sequence of a user, then it is declared as active (*identification*). In the second step it is already known which users are active in this block, and the task is to decode their messages from the successfully transmitted packets (*decoding*). Obviously, two different types of errors can happen: false identification, and false decoding.

Let us choose T protocol sequences randomly. Each one has constant weight w. Protocol sequences are divided into w segments of length $\frac{n}{w}$ (integer) and in each segment there is exactly one 1 whose position is uniformly distributed and independent of the others.

Firstly, we consider the task of identification.

Lemma 4.5 (Györfi and Győri (2004)). *For synchronous access and non-binary packets*

$$\mathbf{P}\{\text{false identification}\} \le \exp\left((M+1)\ln T - w\left(1-\frac{w}{n}\right)^{M}\right). \tag{4.7}$$

Proof. This is the same problem what was solved in Theorem 2.7. Let the weight of the code words be $w = \frac{n}{M}$, so in Theorem 2.7 the segments' length $L = M$, and the statement follows from (2.17) if we apply that $\ln(1-x) \le -x$, $\forall x \in \mathbb{R}$. □

Decoding error occurs if there are less than k successfully transmitted packets (uncovered 1's in the protocol sequence) of an active user.

Lemma 4.6 (Györfi and Győri (2004)). *For synchronous access and non-binary packets, if $w \ge \frac{k}{p}$ then*

$$\mathbf{P}\{\text{false decoding}\} \le \exp\left(-\frac{(wp-k)^2}{3wp} + M\ln T\right), \tag{4.8}$$

where

$$p = \left(1-\frac{w}{n}\right)^{M-1}.$$

Proof. Let us select at most M protocol sequences (users), and call one of them tagged user. Let p be the probability of the event that in a fixed segment the tagged user has an uncovered 1 (i.e., its 1 is not covered by the other $M-1$ users), thus

$$p = \left(1 - \frac{w}{n}\right)^{M-1}.$$

Then the probability that there are exactly i positions where the tagged user has uncovered 1's and the other $w-i$ positions are covered by the other $M-1$ users is at most $\binom{w}{i}p^i(1-p)^{w-i}$. The probability that the tagged user has less than k uncovered positions is at most

$$\sum_{i<k}\binom{w}{i}p^i(1-p)^{w-i}.$$

The probability that there is a protocol sequence which has less than k uncovered positions (other positions are covered by the other $M-1$ protocol sequences) is at most

$$\mathbf{P}\{\text{false decoding}\} \le T\binom{T-1}{M-1}\sum_{i<k}\binom{w}{i}p^i(1-p)^{w-i}.$$

Let us apply now Lemma B.2 for upper bounding the tail of binomially distributed random variable (which is the sum of indicator variables)

$$\begin{aligned}
\mathbf{P}\{\text{false decoding}\} &\le T\binom{T-1}{M-1}\sum_{i<k}\binom{w}{i}p^i(1-p)^{w-i} \\
&= T\binom{T-1}{M-1}\mathbf{P}\left\{\sum_{i=1}^{w}X_i < k\right\} \\
&= T\binom{T-1}{M-1}\mathbf{P}\left\{\tfrac{1}{w}\sum_{i=1}^{w}(X_i - \mathbf{E}X_i) < -\left(p - \tfrac{k}{w}\right)\right\} \\
&\le \exp\left(-\frac{w\left(p-\frac{k}{w}\right)^2}{2p(1-p)+\frac{2}{3}\left(p-\frac{k}{w}\right)} + M\ln T\right) \\
&\le \exp\left(-\frac{w\left(p-\frac{k}{w}\right)^2}{2p+p} + M\ln T\right) \\
&= \exp\left(-\frac{(wp-k)^2}{3wp} + M\ln T\right),
\end{aligned}$$

where $X_1, X_2, \ldots, X_w$ are independent indicator random variables with parameter p, and in Lemma B.2 $a=0$, $b=1$, $\sigma^2 = p(1-p)$ and $\varepsilon = p - \frac{k}{w}$. □

Proof of Theorem 4.5. If T randomly chosen protocol sequences of length n and weight w satisfy the requirement of identification and the decoding of the sent messages is always possible, then the protocol sequences and codes of users can be applied for T users in communication on a multiple access collision channel.

Obviously,

$$\mathbf{P}\{\text{bad code}\} \quad \leq \quad \mathbf{P}\{\text{false identification}\} + \mathbf{P}\{\text{false decoding}\},$$

and we need

$$\mathbf{P}\{\text{bad code}\} < 1,$$

since then there is a good code. This gives an upper bound on minimum frame size n. Thus, we it is enough if the following probabilities tend to 0

$$\begin{aligned} \mathbf{P}\{\text{false identification}\} &\to 0, \\ \mathbf{P}\{\text{false decoding}\} &\to 0. \end{aligned}$$

For the decodability property we get by taking the logarithm of (4.8)

$$-\frac{(wp-k)^2}{3wp} + M \ln T < 0. \tag{4.9}$$

The solution of this inequality with respect to positive weight w is

$$w > \frac{k}{p}(1+\alpha)\left(1+\sqrt{1-\frac{1}{(1+\alpha)^2}}\right),$$

where

$$\alpha = \frac{3}{2}\frac{M \ln T}{k}.$$

As $\alpha \to 0$, we have the following asymptotic inequality for w:

$$w \gtrsim \frac{k}{p}.$$

Let the length of the segments be M, so $n = Mw$, and now p depends only on M

$$p = \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If we choose the weight of the protocol sequences w to

$$w = (1+\delta)\frac{k}{p} \tag{4.10}$$

for an arbitrary constant $\delta > 0$, the exponent in (4.8) become

$$-k\left(\frac{\delta^2}{3(1+\delta)} - \frac{M\ln T}{k}\right)$$

which tends to $-\infty$ when $k \to \infty$ and $\frac{\ln T}{k} \to 0$, that is why for such a weight w

$$\mathbf{P}\{\text{false decoding}\} \to 0.$$

By the choice of (4.10) the exponent in (4.7) become

$$-k\left((1+\delta)\left(1-\tfrac{1}{M}\right) - \frac{(M+1)\ln T}{k}\right)$$

which also tends to $-\infty$ when $k \to \infty$ and $\frac{\ln T}{k} \to 0$, that is why

$$\mathbf{P}\{\text{false identification}\} \to 0,$$

so there exists a good code $\mathcal{C}$. As the reasoning above is true for all arbitrarily small $\delta > 0$, the next asymptotic upper bound on the minimum weight w is true:

$$w \lesssim \frac{k}{p} = k\left(1-\tfrac{1}{M}\right)^{1-M}.$$

Finally, we have shown the following asymptotic upper bound on the minimum frame size n:

$$n(T,M,k) = Mw \lesssim kM\left(1-\tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim \left(1-\tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T,M,k) \lesssim kM\mathrm{e},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim \mathrm{e}^{-1}.$$

□

From Theorem 4.4 and 4.5 we have the following:

Corollary 4.1 (Györfi and Győri (2004)). *For synchronous access and non-binary packets, if M is fixed, $T \to \infty, |I| \to \infty, k \to \infty, |I| > \mathrm{e}k, \frac{\log T}{\log |I|} \to 0$ and $\frac{\log T}{k} \to 0$, then*

$$n(T,M,k) \simeq kM \left(1 - \tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T,M) \simeq \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T,M,k) \simeq kM\mathrm{e},$$

and for the sum-rate

$$R_{sum}(T,M) \simeq \mathrm{e}^{-1}.$$

4.4 Bounds for asynchronous access

In Section 4.3 frame synchronization was assumed, so frames of the users begin at the same slots (no time shift). In this section we study the frame *asynchronous* case.

As the minimum block length for asynchronous access is lower bounded by the minimum block length for synchronous access, Theorem 4.4 gives us a lower bound on the minimum block length $n(T,M,k)$ in the case of asynchronous access, too.

In order to get an upper bound on the minimum block length $n(T,M,k)$—similarly to the synchronous case—randomly chosen protocol sequences of constant weight w are used, and as an inner code a Reed–Solomon code $\mathcal{C}_i = \mathcal{C}$ of parameters (w,k) is applied over GF$(|I|)$ $(w \le |I|)$.

For asynchronous access the upper bound on the minimum length of the protocol sequences is the same as for synchronous case.

Theorem 4.6 (Györfi and Győri (2005)). *For asynchronous access and non-binary packets, if M is fixed, $T \to \infty$, $k \to \infty$, $\frac{\log T}{k} \to 0$ and $|I| > \mathrm{e}k$, then*

$$n(T,M,k) \lesssim kM \left(1 - \tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T,M) \gtrsim \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T, M, k) \lesssim kM\mathrm{e},$$

and for the sum-rate

$$R_{sum}(T, M) \gtrsim \mathrm{e}^{-1}.$$

In the case of asynchronous access the coding method has to ensure the synchronization in addition to the identification and decoding. The decoding depends on the codes of the users, and the others on the protocol sequences of the users.

The detection is done by a two phase algorithm. In the first step a sliding window is used whose length equals to the block length. The successfully transmitted packets and the collision symbol on the output of the channel are transformed to a bit 1, and the $\emptyset$ symbol to bit 0. The resulted binary vector is actually the Boolean sum of the protocol sequences of the active users. If, starting at a position, this binary vector covers the protocol sequence of a user, then it is declared as active (*identification*) beginning at this position (*synchronization*). In the second step it is already known which users are active in this block, and the task is to decode their messages from the successfully transmitted packets (*decoding*). Obviously, three different types of errors can happen: false identification, false synchronization, and false decoding.

Remark. During the design of the protocol sequences it is supposed that the decoding algorithm does *not* have a memory (stateless). We have synchronization error only when a protocol sequence is covered by the *beginning* of its shifted version and some other protocol sequences. During the application of these protocol sequences we use a decoding algorithm with memory (stateful). If a user is declared as active beginning at a given position, then he will be active in the next n time slots, so the algorithm need not to check its coverage in the next n time slots. Consequently, it does not cause synchronization problem if a protocol sequence is covered by the *end* of its shifted version and some other protocol sequences.

Let us choose T protocol sequences randomly. Each one has constant weight w. Protocol sequences are divided into w segments of length $\frac{n}{w}$ (integer) and in each segment there is exactly one 1 whose position is uniformly distributed and independent of the others.

Firstly, we consider the identification task.

Lemma 4.7.

$$\mathbf{P}\{\text{false identification}\} \leq \exp\left((2M+1)\ln T + 2M\ln n - w\left(1 - \frac{w}{n}\right)^{M}\right) \tag{4.11}$$

Proof. Let us fix some arbitrarily shifted interfering protocol sequences (users) such that there are at most M active ones in every time slot. (This can result in at most $2M$ users.) Choose another (tagged) protocol sequence distinctly from the others. Similarly to the proof of Lemma 4.7, the probability that there exists a protocol sequence such that all positions of it are covered by the sum of another arbitrarily shifted protocol sequences is at most

$$\begin{aligned}
&\mathbf{P}\{\text{false identification}\}\\
&\quad\le T\tbinom{T-1}{2M}n^{2M}\left(1-\left(1-\tfrac{w}{n}\right)^{M}\right)^{w}\\
&\quad\le T^{2M+1}n^{2M}\left(1-\left(1-\tfrac{w}{n}\right)^{M}\right)^{w}\\
&\quad= \exp\left((2M+1)\ln T+2M\ln n+w\ln\left(1-\left(1-\tfrac{w}{n}\right)^{M}\right)\right)\\
&\quad\le \exp\left((2M+1)\ln T+2M\ln n-w\left(1-\tfrac{w}{n}\right)^{M}\right)
\end{aligned}$$

where in the last step we applied that $\ln(1-x)\le -x$, $\forall x\in\mathbb{R}$. □

Now, we consider the synchronization task.

Lemma 4.8.

$$\mathbf{P}\{\text{false synchronization}\}\le \exp\left(2M\ln T+2M\ln n-\frac{w}{6}\left(1-\frac{w}{n}\right)^{M}\right) \tag{4.12}$$

Proof. Let us choose a tagged protocol sequence and suppose that it is also among the active ones but it is shifted with $0<s_1\frac{n}{w}+s_2<n$ time slots ($0\le s_2\le \frac{n}{w}-1$), and fix some arbitrarily shifted protocol sequences (users) such that there are at most M active ones in every time slot. We denote by A_i ($1\le i\le w$) the event that the 1 of the tagged user in the i^{th} segment is covered (either by the shifted version of itself or by the other active users). The probability of each event A_i is

$$\mathbf{P}\{A_i\}=1-\left(1-\frac{w}{n}\right)^{M},$$

but events A_i's are dependent of each other, because A_i depends on the position of the 1 in segments $i, i-s_1-1, i-s_1$ (if these segments are exists). Variables c_i ($1\le i\le w$) correspond to the encoded packets, so their values tell the position of the 1 in a segment. In Figure 4.2 the dependence of covering events A_i is illustrated. A_i can be in conflict with at most 6 other events, namely with $A_{i-1}, A_{i+1}, A_{i-s_1-1}, A_{i-s_1}, A_{i+s_1}, A_{i+s_1+1}$. That is why 6

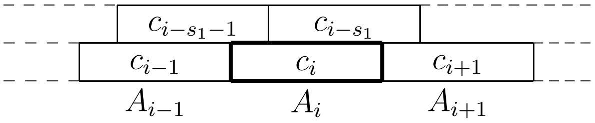

Figure 4.2: The dependence of covering events

independent classes B_j, $1 \le j \le 6$ of A_i's can be formed, where $B_j \cap B_\ell = \emptyset$, and $\bigcup_{j=1}^{6} B_j = \{A_1, \dots, A_w\}$.

Let A_i's be the vertices of an undirected graph, in which two vertices are connected if they are in conflict, i.e., if they depend on the same c_ℓ. The maximum degree of each vertex is at most 6, and the graph is neither a cycle graph with an odd number of vertices, nor a complete graph, so it can be colored with 6 colors (cf. Brooks (1941)). Events within one color class are independent of each other. B_j is the class of A_i's of color j. At least one of the B_j's has $\frac{w}{6}$ elements or more.

$$\begin{aligned}
\mathbf{P}\left\{\bigcap_{i=1}^{w} A_i\right\} &= \mathbf{P}\left\{\bigcap_{j=1}^{6} \{A_i : A_i \in B_j\}\right\} \\
&\le \min_{j=1,\dots,6} \mathbf{P}\left\{\bigcap_{A_i \in B_j} A_i\right\} \\
&\le \left(1 - \left(1 - \frac{w}{n}\right)^M\right)^{\frac{w}{6}},
\end{aligned}$$

therefore

$$\begin{aligned}
\mathbf{P}\{\text{false synchronization}\} &\le T\binom{T-1}{2M-1} n^{2M} \left(1 - \left(1 - \tfrac{w}{n}\right)^M\right)^{\frac{w}{6}} \\
&\le T^{2M} n^{2M} \left(1 - \left(1 - \tfrac{w}{n}\right)^M\right)^{\frac{w}{6}} \\
&= \exp\left(2M \ln T + 2M \ln n + \tfrac{w}{6} \ln\left(1 - \left(1 - \tfrac{w}{n}\right)^M\right)\right) \\
&\le \exp\left(2M \ln T + 2M \ln n - \tfrac{w}{6}\left(1 - \tfrac{w}{n}\right)^M\right).
\end{aligned}$$

□

Decoding error occurs if there are less than k successfully transmitted packets (uncovered 1's in the protocol sequence) of an active user.

Lemma 4.9 (Györfi and Győri (2005)). *For asynchronous access and non-binary packets, if* $w \geq \frac{k}{p}$ *then*

$$\mathbf{P}\{\text{false decoding}\} \leq \exp\left(-\frac{(wp-k)^2}{3wp} + (2M-2)\ln n + (2M-1)\ln T\right), \tag{4.13}$$

where

$$p = \left(1 - \frac{w}{n}\right)^{M-1}.$$

Proof. Let us select some arbitrarily shifted protocol sequences (users), such that there are at most M active ones in every time slot, and call one of them tagged user. Let p be the probability of the event that in a fixed segment the tagged user has an uncovered 1 (i.e., its 1 is not covered by the other $M-1$ users). Similarly to the proof of Lemma 4.6 the probability that there is a protocol sequence which has less than k uncovered positions (other positions are covered by the other $M-1$ protocol sequences) is at most

$$\mathbf{P}\{\text{false decoding}\} \leq T\tbinom{T-1}{2M-2} n^{2M-2} \sum_{i<k} \binom{w}{i} p^i (1-p)^{w-i},$$

and by applying Bernstein's inequality (Lemma B.2) for upper bounding the tail of binomially distributed random variable we get

$$\mathbf{P}\{\text{false decoding}\} \leq \exp\left(-\frac{(wp-k)^2}{3wp} + (2M-2)\ln n + (2M-1)\ln T\right).$$

□

Proof of Theorem 4.6. If T randomly chosen protocol sequences of length n and weight w satisfy the requirements of identification and synchronization, and the decoding of the sent messages is always possible, then the protocol sequences and codes of users can be applied for T users in communication on a multiple access collision channel. Obviously,

$$\mathbf{P}\{\text{bad code}\} \leq \mathbf{P}\{\text{false ident.}\} + \mathbf{P}\{\text{false synch.}\} + \mathbf{P}\{\text{false decoding}\},$$

and we need

$$\mathbf{P}\{\text{bad code}\} < 1,$$

since then there is a good code. This gives an upper bound on minimum frame size n. Thus, it is enough if the following probabilities tend to 0

$$\begin{aligned} \mathbf{P}\{\text{false identification}\} &\to 0, \\ \mathbf{P}\{\text{false synchronization}\} &\to 0, \\ \mathbf{P}\{\text{false decoding}\} &\to 0. \end{aligned}$$

For the decodability property we get by taking the logarithm of (4.13)

$$-\frac{(wp-k)^2}{3wp} + (2M-2)\ln n + (2M-1)\ln T < 0. \tag{4.14}$$

If the middle term is temporarily ignored, the solution of this inequality with respect to positive weight w is

$$w > \frac{k}{p}(1+\alpha)\left(1+\sqrt{1-\frac{1}{(1+\alpha)^2}}\right),$$

where

$$\alpha = \frac{3}{2}\frac{(2M-1)\ln T}{k}.$$

As $\alpha \to 0$, we have the following asymptotic inequality for w:

$$w \gtrsim \frac{k}{p}.$$

Let the length of the segments be M, so $n = Mw$, and now p depends only on M

$$p = \left(1-\tfrac{1}{M}\right)^{M-1}.$$

If we choose the weight of the protocol sequences w to

$$w = (1+\delta)\frac{k}{p} \tag{4.15}$$

for an arbitrary constant $\delta > 0$, the exponent in (4.13) becomes

$$-k\left(\frac{\delta^2}{3(1+\delta)} - \frac{(2M-2)\ln\left((1+\delta)\frac{kM}{p}\right)}{k} - \frac{(2M-1)\ln T}{k}\right)$$

which tends to $-\infty$ when $k \to \infty$ and $\frac{\ln T}{k} \to 0$, that is why for such a weight w

$$\mathbf{P}\{\text{false decoding}\} \to 0.$$

By the choice of (4.15) the exponent in (4.11) becomes

$$-k\left((1+\delta)\left(1-\tfrac{1}{M}\right) - \frac{(2M+1)\ln T}{k} - \frac{2M\ln\left((1+\delta)\frac{kM}{p}\right)}{k}\right)$$

and the exponent in (4.12) becomes

$$-k\left(\frac{1+\delta}{6}\left(1-\tfrac{1}{M}\right)-\frac{2M\ln T}{k}-\frac{2M\ln\left((1+\delta)\frac{kM}{p}\right)}{k}\right)$$

and both of them tend to $-\infty$ when $k\to\infty$ and $\frac{\ln T}{k}\to 0$, that is why

$$\mathbf{P}\{\text{false identification}\}\to 0,$$

and

$$\mathbf{P}\{\text{false synchronization}\}\to 0,$$

so there exists a good code $\mathcal{C}$. As the reasoning above is true for all arbitrarily small $\delta>0$, the next asymptotic upper bound on the minimum weight w is true:

$$w\lesssim\frac{k}{p}=k\left(1-\tfrac{1}{M}\right)^{1-M}.$$

Finally, we have shown the following asymptotic upper bound on the minimum frame size n:

$$n(T,M,k)\lesssim kM\left(1-\tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T,M)\gtrsim\left(1-\tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M\to\infty$, then

$$n(T,M,k)\lesssim kM\mathrm{e},$$

and for the sum-rate

$$R_{\text{sum}}(T,M)\gtrsim\mathrm{e}^{-1}.$$

□

From Theorem 4.4 and 4.6 we have the following:

Corollary 4.2 (Györfi and Győri (2005)). *For asynchronous access and non-binary packets, if M is fixed, $T\to\infty, |I|\to\infty, k\to\infty, |I|>\mathrm{e}k, \frac{\log T}{\log|I|}\to 0$ and $\frac{\log T}{k}\to 0$, then*

$$n(T,M,k)\simeq kM\left(1-\tfrac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T, M) \simeq \left(1 - \tfrac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$*, then*

$$n(T, M, k) \simeq kM\mathrm{e},$$

and for the sum-rate

$$R_{sum}(T, M) \simeq \mathrm{e}^{-1}.$$

4.5 Constructions

Chapter 5

Slow Frequency Hopping

5.1 Channel model

In a slow frequency-hop packet radio network, the total RF bandwidth is divided into L subbands, and the time is divided into intervals called slots. Transmission of a packet must take place wholly within a packet slot. The RF signal from a given transmitter is hopped from slot to slot by changing the carrier frequency. The sequence of carrier frequencies used by a signal is known as its frequency-hopping pattern or protocol sequence. The duration between two consecutive hop epochs is called the hop interval. In our case the hop interval is the time slot.

The requirement for slotted frequency hopping is that the hop intervals for all transmitters must be aligned at all receivers. For many applications, this requirement cannot be met, and only unslotted hopping is feasible. Within slotted hopping the frames of the hop-sequences can be synchronous or asynchronous. Assume that there is no noise in the system. Pursley (1987) made an excellent survey on the general problem.

More formally, assume L frequency bands, so the channel consists of L parallel multiple access channels without feedback, where the message to be sent over a common communications channel is in the form of "packets", that are sent afteranother in time-frequency slots selected from a time-frame of length n slots. If, in a particular time-frequency slot, a user is sending a packet and there are no other users sending packets partially overlapping with that slot, then the channel output is this packet value, otherwise the channel output in that slot is the collision symbol. There is no feedback available to inform the senders of the channel outputs in previous slots.

Take the slot duration as the unit of time. Assume the number of potential users $T < \infty$. The arrivals of the messages are according to a birth process with intensity $\lambda_m = c(T - m)$, where m is the number of active users (for $m = 1, 2, \ldots, T$).

Each user has a protocol sequence. Let $q_i = (q_{i,1}, \ldots, q_{i,n})$ be the protocol sequence of user i such that

$$0 \leq q_{i,j} \leq L - 1.$$

If $q_{i,j} = \ell$ then user i can send a packet during the j^{th} slot at frequency ℓ.

Assume that each message of length k is encoded by an (n, k) shortened Reed–Solomon code ($k < n < Q$). In the j^{th} slot at the selected frequency the j^{th} encoded packet of the Reed–Solomon codeword is sent ($j = 1, \ldots, n$). For a message from a given user (called the tagged user) decoding is possible if there are at most $n - k$ collisions (erasures), and the decoder knows which are the n chosen slots for the actual encoded message. The decoder should separate its packets from the others. Such separation is possible by the addresses of the users: the address of the user is stored in the head of the packet. Note that the decoder should know the serial numbers of the packets sent, which is an additional overhead of size $\log n$.

5.2 Bounds for non-binary packets

We consider the case when the input alphabet I contains more than two elements. Here k information packets are encoded, so the number of different messages is $|I|^k$, therefore the sum-rate is defined as

$$R_{\text{sum}} = \frac{kM}{nL}$$

Theorem 5.1. *For non-binary packets, if M is fixed, $T \to \infty, |I| \to \infty$, and $\frac{\log T}{\log |I|} \to 0$, then*

$$n(T, M, L, k) \gtrsim k \left(1 - \tfrac{1}{L}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T, M, L) \lesssim \frac{M}{L} \left(1 - \tfrac{1}{L}\right)^{M-1}.$$

Proof. For the minimum code length, entropy based lower bound is given. Now, the proof is the same as the proof of Theorem 4.4, but n should be replaced by nL.

The code words of the slow frequency hopping channel with L subbands and length n can be mapped to the code words of the collision channel of length nL with the restriction that there are exactly one 1 in each segment of length L. In the following we are dealing with the mapped code words.

For the minimum code length, entropy based lower bound is given. For a deterministic channel, the entropy of the channel input block can not be greater than the entropy of the output block of the channel. If the codes can solve the tasks of identification and decoding, then the entropy of the output block is equal to the entropy of the input block. If M users out of T send packets into the channel and each message takes values from a set of size $S = |I|^k$, then the input random variable can take $\binom{T}{M}(|I|^k)^M$ different values. (Note, that minimum block length needed for *at most* M users is greater or equal to the minimum block length needed for *exactly* M users, therefore it is enough to consider here the latter scenario.) Assume that the input variable is uniformly distributed, so the entropy of the input random variable is $\log\binom{T}{M}|I|^{kM}$. For the output random variable we can apply the sum of the componentwise entropy as upper bound. All together we have the following inequality

$$\log\left(\binom{T}{M}|I|^{kM}\right) = H(O_1,\dots,O_n) \le \sum_{i=1}^{nL} H(O_i) \le nL\max_i H(O_i),$$

where O_i corresponds to the i^{th} position of the output. Let w_i be the number of protocol sequences which have 1's at the i^{th} position. The entropy $H(O_i)$ is the highest possible if O_i is uniformly distributed on all $a \in I$. In this case the distribution of O_i can be calculated in the following way:

$$\begin{aligned}
\mathbf{P}\{O_i = \emptyset\} &= \frac{\binom{T-w_i}{M}}{\binom{T}{M}} := p_0 \\
\mathbf{P}\{O_i = a\} &= \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}|I|} := \frac{p_1}{|I|}, \quad \forall a \in I \\
\mathbf{P}\{O_i = *\} &= 1 - \frac{\binom{T-w_i}{M}}{\binom{T}{M}} - \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}} = 1 - p_0 - p_1.
\end{aligned}$$

The entropy of O_i can be upper bounded as

$$\begin{aligned}
H(O_i) &\le -p_0\log p_0 - |I|\cdot\frac{p_1}{|I|}\log\frac{p_1}{|I|} - (1-p_0-p_1)\log(1-p_0-p_1) \\
&= -p_0\log p_0 - p_1\log p_1 - (1-p_0-p_1)\log(1-p_0-p_1) + p_1\log|I| \\
&\le \log 3 + p_1\log|I| \\
&= \log 3 + \frac{w_i\binom{T-w_i}{M-1}}{\binom{T}{M}}\log|I|
\end{aligned}$$

$$
\begin{aligned}
&= \log 3 + \frac{w_i M}{T-M+1} \cdot \frac{(T-w_i)\cdots(T-w_i-M+2)}{T\cdots(T-M+2)} \log|I| \\
&\leq \log 3 + \frac{w_i M}{T} \frac{1}{1-\frac{M-1}{T}} \left(1-\frac{w_i}{T}\right)^{M-1} \log|I| \\
&\leq \log 3 + \frac{M}{L} \frac{1}{1-\frac{M-1}{T}} \left(1-\frac{1}{L}\right)^{M-1} \log|I| \\
&\simeq \log 3 + \frac{M}{L} \left(1-\frac{1}{L}\right)^{M-1} \log|I|,
\end{aligned}
$$

where we used that $\frac{w_i M}{T}\left(1-\frac{w_i}{T}\right)^{M-1}$ takes its maximum at $w_i = \frac{T}{L}$, if there is a restriction that the sum of L neighboring w_i's is T. The calculation above implies that

$$
\log\left(\binom{T}{M}|I|^{kM}\right) \leq nL\left(\tfrac{M}{L}\left(1-\tfrac{1}{L}\right)^{M-1}\log|I| + \log 3\right).
$$

For the minimum code length we get (by using $\left(\frac{T}{M}\right)^M \leq \binom{T}{M}$)

$$
\begin{aligned}
n &\gtrsim \frac{M\log\frac{T}{M} + kM\log|I|}{M\left(1-\frac{1}{L}\right)^{M-1}\log|I| + L\log 3} \\
&= \frac{\frac{M\log\frac{T}{M}}{\log|I|} + kM}{M\left(1-\frac{1}{L}\right)^{M-1} + \frac{L\log 3}{\log|I|}}.
\end{aligned}
$$

If $|I| \to \infty$ and $\frac{\log T}{\log|I|} \to 0$, then

$$
n(T,M,L,k) \gtrsim k\left(1-\tfrac{1}{L}\right)^{1-M},
$$

and for the sum-rate

$$
R_{\text{sum}}(T,M,L) = \frac{kM}{nL} \lesssim \frac{M}{L}\left(1-\tfrac{1}{L}\right)^{M-1}.
$$

□

In order to get an upper bound on the minimum block length $n(T,M,L,k)$ randomly chosen protocol sequences are used, and as an inner code a Reed–Solomon code $\mathcal{C}_i = \mathcal{C}$ of parameters (n,k) is applied over GF$(|I|)$ $(n \leq |I|)$.

For asynchronous access the upper bound on the minimum length of the protocol sequences is at least the one for synchronous case, so it is enough to consider only the asynchronous case.

Theorem 5.2. *For synchronous or asynchronous access and non-binary packets, if M is fixed, $T \to \infty$, $k \to \infty$, $\frac{\log T}{k} \to 0$ and $|I| > \mathrm{e}k$, then*

$$n(T, M, L, k) \lesssim k \left(1 - \tfrac{1}{L}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T, M, L) \gtrsim \frac{M}{L} \left(1 - \tfrac{1}{L}\right)^{M-1}.$$

In the case of asynchronous access the coding method has to ensure the synchronization in addition to the identification and decoding. The decoding depends on the codes of the users, and the others on the protocol sequences of the users.

The detection is done by a two phase algorithm. In the first step a sliding window is used whose length equals to the block length. The successfully transmitted packets and the collision symbol on the output of the channel are transformed to a bit 1, and the $\emptyset$ symbol to bit 0. The resulted binary matrix is actually the Boolean sum of the protocol sequences of the active users. If, starting at a position, this binary matrix covers the protocol sequence of a user, then it is declared as active (*identification*) beginning at this position (*synchronization*). In the second step it is already known which users are active in this block, and the task is to decode their messages from the successfully transmitted packets (*decoding*). Obviously, three different types of errors can happen: false identification, false synchronization, and false decoding.

Remark. During the design of the protocol sequences it is supposed that the decoding algorithm does *not* have a memory (stateless). We have synchronization error only when a protocol sequence is covered by the *beginning* of its shifted version and some other protocol sequences. During the application of these protocol sequences we use a decoding algorithm with memory (stateful). If a user is declared as active beginning at a given position, then he will be active in the next n time slots, so the algorithm need not to check its coverage in the next n time slots. Consequently, it does not cause synchronization problem if a protocol sequence is covered by the *end* of its shifted version and some other protocol sequences.

Let us choose T protocol sequences randomly. In each time slot there is exactly one 1 whose position is uniformly distributed and independent of the others.

Firstly, we consider the identification task.

Lemma 5.1.

$$\mathbf{P}\{\text{false identification}\} \le \exp\left((2M+1)\ln T + 2M\ln n - n\left(1-\tfrac{1}{L}\right)^{M}\right) \tag{5.1}$$

Proof. Let us fix some arbitrarily shifted interfering protocol sequences (users) such that there are at most M active ones in every time slot. (This can result in at most $2M$ users in a block.) Choose another (tagged) protocol sequence distinctly from the others. The proof is exactly the same as the proof of Lemma 3.1, but we have here $2M$ active users in a block instead of M, so

$$\begin{aligned}&\mathbf{P}\{\text{false identification}\}\\ &\quad\le \exp\left((2M+1)\ln T + 2M\ln n + n\ln\left(1-\left(1-\tfrac{1}{L}\right)^{M}\right)\right),\end{aligned}$$

and then let us apply that $\ln(1-x) \le -x$, $\forall x \in \mathbb{R}$. □

Now, we consider the synchronization task.

Lemma 5.2.

$$\mathbf{P}\{\text{false synchronization}\} \le \exp\left(2M\ln T + 2M\ln n - n\left(1-\tfrac{1}{L}\right)^{M}\right) \tag{5.2}$$

Proof. Let us choose a tagged protocol sequence and suppose that it is also among the active ones but with some shift, and fix some arbitrarily shifted protocol sequences (users) such that there are at most M active ones in every time slot. (This can result in at most $2M$ users in a block.) The proof is exactly the same as the proof of Lemma 3.4, but we have here $2M$ active users in a block instead of M, so

$$\begin{aligned}&\mathbf{P}\{\text{false synchronization}\}\\ &\quad\le \exp\left(2M\ln T + 2M\ln n + n\ln\left(1-\left(1-\tfrac{1}{L}\right)^{M}\right)\right),\end{aligned}$$

and then let us apply that $\ln(1-x) \le -x$, $\forall x \in \mathbb{R}$. □

Decoding error occurs if there are less than k successfully transmitted packets (uncovered 1's in the protocol sequence) of an active user.

Lemma 5.3. *For asynchronous access and non-binary packets, if* $n \ge \frac{k}{p}$ *then*

$$\mathbf{P}\{\text{false decoding}\} \le \exp\left(-\frac{(np-k)^2}{3np} + (2M-2)\ln n + (2M-1)\ln T\right), \tag{5.3}$$

where

$$p = \left(1-\tfrac{1}{L}\right)^{M-1}.$$

Proof. Let us select some arbitrarily shifted protocol sequences (users), such that there are at most M active ones in every time slot, and call one of them tagged user. Let p be the probability of the event that in a fixed segment the tagged user has an uncovered 1 (i.e., its 1 is not covered by the other $M-1$ users). Similarly to the proof of Lemma 4.6 the probability that there is a protocol sequence which has less than k uncovered positions (other positions are covered by the other $M-1$ protocol sequences) is at most

$$\mathbf{P}\{\text{false decoding}\} \leq T\tbinom{T-1}{2M-2} n^{2M-2} \sum_{i<k} \binom{n}{i} p^i (1-p)^{n-i},$$

and by applying Bernstein's inequality (Lemma B.2) for upper bounding the tail of binomially distributed random variable we get

$$\mathbf{P}\{\text{false decoding}\} \leq \exp\left(-\frac{(np-k)^2}{3np} + (2M-2)\ln n + (2M-1)\ln T \right).$$

□

Proof of Theorem 5.2. If T randomly chosen protocol sequences of length n satisfy the requirements of identification and synchronization, and the decoding of the sent messages is always possible, then the protocol sequences and codes of users can be applied for T users in communication on a slow frequency hopping channel. Obviously,

$$\mathbf{P}\{\text{bad code}\} \leq \mathbf{P}\{\text{false ident.}\} + \mathbf{P}\{\text{false synch.}\} + \mathbf{P}\{\text{false decoding}\},$$

and we need

$$\mathbf{P}\{\text{bad code}\} < 1,$$

since then there is a good code. This gives an upper bound on minimum block size n. Thus, it is enough if the following probabilities tend to 0

$$\begin{aligned} \mathbf{P}\{\text{false identification}\} &\to 0, \\ \mathbf{P}\{\text{false synchronization}\} &\to 0, \\ \mathbf{P}\{\text{false decoding}\} &\to 0. \end{aligned}$$

For the decodability property we get by taking the logarithm of (5.3)

$$-\frac{(np-k)^2}{3np} + (2M-2)\ln n + (2M-1)\ln T < 0. \tag{5.4}$$

If the middle term is temporarily ignored, the solution of this inequality with respect to positive weight n is

$$n > \frac{k}{p}(1+\alpha)\left(1+\sqrt{1-\frac{1}{(1+\alpha)^2}}\right),$$

where

$$\alpha = \frac{3}{2}\frac{(2M-1)\ln T}{k}.$$

As $\alpha \to 0$, we have the following asymptotic inequality for n:

$$n \gtrsim \frac{k}{p}.$$

If we choose the length of the protocol sequences n to

$$n = (1+\delta)\frac{k}{p} \tag{5.5}$$

for an arbitrary constant $\delta > 0$, the exponent in (5.3) becomes

$$-k\left(\frac{\delta^2}{3(1+\delta)} - \frac{(2M-2)\ln\left((1+\delta)\frac{kM}{p}\right)}{k} - \frac{(2M-1)\ln T}{k}\right)$$

which tends to $-\infty$ when $k \to \infty$ and $\frac{\ln T}{k} \to 0$, that is why for such a length n

$$\mathbf{P}\{\text{false decoding}\} \to 0.$$

By the choice of (5.5) the exponent in (5.1) becomes

$$-k\left((1+\delta)\left(1-\tfrac{1}{L}\right) - \frac{(2M+1)\ln T}{k} - \frac{2M\ln\left((1+\delta)\frac{k}{p}\right)}{k}\right)$$

and the exponent in (5.2) becomes

$$-k\left((1+\delta)\left(1-\tfrac{1}{L}\right) - \frac{2M\ln T}{k} - \frac{2M\ln\left((1+\delta)\frac{k}{p}\right)}{k}\right)$$

and both of them tend to $-\infty$ when $k \to \infty$ and $\frac{\ln T}{k} \to 0$, that is why

$$\mathbf{P}\{\text{false identification}\} \to 0,$$

and

$$\mathbf{P}\{\text{false synchronization}\} \to 0,$$

so there exists a good code $\mathcal{C}$. As the reasoning above is true for all arbitrarily small $\delta > 0$, the next asymptotic upper bound on the minimum block length n is true:

$$n \lesssim \frac{k}{p} = k\left(1 - \tfrac{1}{L}\right)^{1-M}.$$

Finally, we have shown the following asymptotic upper bound on the minimum block length n:

$$n(T, M, L, k) \lesssim k\left(1 - \tfrac{1}{L}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T, M, L) \gtrsim \frac{M}{L}\left(1 - \tfrac{1}{L}\right)^{M-1}.$$

□

Corollary 5.1. *For synchronous access and non-binary packets, if M is fixed, $T \to \infty, |I| \to \infty, k \to \infty, |I| > \mathrm{e}k, \frac{\log T}{\log |I|} \to 0$ and $\frac{\log T}{k} \to 0$, then*

$$n(T, M, L, k) \simeq k\left(1 - \tfrac{1}{L}\right)^{1-M},$$

and for the sum-rate

$$R_{sum}(T, M, L) \simeq \frac{M}{L}\left(1 - \tfrac{1}{L}\right)^{M-1}.$$

5.3 Constructions

Chapter 6

Collision Channel with Ternary Feedback

6.1 Collision Channel with Feedback

In multi-user communications the problem is how to serve many senders if one common communication channel is given. The classical solution is a kind of multiplexing, i.e., either time-division multiplexing, or frequency-division multiplexing. For partially active senders, always there are a large number of senders, each which has nothing to send most of the time. In this communication situation the multiplexing is inefficient. One such situation, namely the problem of communicating from remote terminals on various islands of Hawaii via a common radio channel to the main central computer, led to the invention by Abramson (1970) of the first formal random-access algorithm, now commonly called pure ALOHA and to the design of a radio linked computer network, called ALOHANET (cf. Abramson (1985)).

In pure ALOHA, a transmitter always transmits a packet at the random moment it is presented (arrived) to the transmitter. If during the transmission of this packet there is no other overlapping transmission by an other transmitter, then the packet is transmitted successfully, and the transmitter is informed about this success. Otherwise, there is a "collision", the transmitter hasn't got ACK that is assumed to destroy all the packets that overlap. When collision happens, the packets must be retransmitted. In order to avoid a repetition of the same collision, pure ALOHA specifies that after a collision each transmitter involved randomly selects a waiting time before it again retransmits its packet. For the analysis of a random access scheme, one usually assumes that the packets have the same size, the transmission time of a packet is called slot, it is the time unit, and we have Poisson traffic, i.e., the arrival moments of the packets form a stationary Poisson process with intensity λ with respect to the slot. Abramson (1970) defined the "throughput" as the fraction of time on the channel occupied by successfully transmitted packets. He showed that the maximum "throughput" is $\frac{1}{2e} \approx 0.184$. Unfortunately, he assume a "statistically equilibrium", i.e., that the time moments of transmissions and retransmissions form a stationary Poisson process, too, which impossible, for any given Poisson $\lambda > 0$ arrivals and for any distribution of the waiting time, the time moments of transmissions and retransmissions cannot form a stationary process, so the conditions of the "throughput" computation cannot be satisfied.

In slotted ALOHA, the time axis is partitioned into slots and each packet to be transmitted fits into one slot. Using the same "statistically equilibrium" assumption, Roberts (1975) showed that the maximum "throughput" is $\frac{1}{e} \approx 0.368$. According to the slotted ALOHA protocol, whenever a packet arrives at a transmitter, that packet is transmitted in the next slot. Whenever a collision occurs in a slot, each packet involved in the collision is said to be backlogged and remains backlogged until it is successfully transmitted. For some fixed $1 > p > 0$, each backlogged packet is retransmitted in the subsequent slot with probability p. It can be seen that, for Poisson arrival, the number of backlogged packets forms a homogeneous Markov chain. We sketch that for any (small) $\lambda > 0$, this Markov chain cannot be stable, i.e., it cannot have unique limit distribution. Let X_n be the number of packets being ready to transmission or retransmission at the end of the n-th slot, V_n the number of successful transmission in the n-th slot, and Y_n the number of arrivals in the n-th slot. Thus, X_n follows the evolution

$$X_{n+1} = (X_n - V_{n+1})^+ + Y_{n+1}.$$

It can be seen that the transition probabilities are of the following form:

$$\begin{aligned} p_{k,k-1} &= kp(1-p)^{k-1}e^{-\lambda} \\ p_{k,k+i} &= kp(1-p)^{k-1}\frac{\lambda^{i+1}}{(i+1)!}e^{-\lambda} + (1-kp(1-p)^{k-1})\frac{\lambda^i}{i!}e^{-\lambda},\ i \geq 0. \end{aligned}$$

If D_k denotes the conditional expectation of the one-step increase of the chain given the state k, then

$$D_k = \mathbf{E}\{X_{n+1} - X_n | X_n = k\} = \lambda - kp(1-p)^{k-1}.$$

It implies that, for any $\lambda > 0$ and for any $1 > p > 0$, D_k will be positive for all sufficiently large k. Thus, if the system becomes sufficiently backlogged, it drifts in the direction of becoming more and more backlogged. Kaplan

(1979) gives a simple and elegant proof that this chain is unstable. From this derivation it is clear that p should be changing, should be the function of a good estimate of k. Hajek and VanLoon (1982) have introduced a class of algorithms in which p is updated in each slot as a function of the previous p and the feedback information. They showed that such functions can be chosen with the property that for $\lambda < 1/e$ the resulting chain is stable.

In this chapter we consider the multiple-access collision channel with ternary feedback. An unlimited number of users are allowed to transmit packets of a fixed length whose duration is taken as a time unit and called slot. Stations can begin to transmit packets only at times $n \in \mathbb{N}$, $\mathbb{N} = \{0, 1, 2, 3 \dots\}$. A *slot* is a time interval $[n, n + 1)$. The destination for the packet contents is a single common receiver. All users send their packets through a common channel. Senders of different packets cannot interchange information. Thus it is convenient to suppose that there are infinitely many non cooperating users and that the packet arrivals can be modelled as a Poisson process in time with, say, intensity λ.

When two or more users send a packet in the same time slot, these packets "collide" and the packet information is lost, i.e., the receiver cannot determine the packet contents, and retransmission will be necessary. However, all users, also those who were not transmitting can learn —from the *ternary feedback* just before time instant $n + 1$— the story of time slot $[n, n + 1)$:

- feedback 0 means an idle slot,
- feedback 1 means successful transmission by a single user,
- feedback of the collision symbol $*$ means that collision happened.

A *conflict resolution protocol* (or *random multiple access algorithm*) is a retransmission scheme for the packets in a collision. Such a scheme must insure the eventual successful transmission of all these packets. A conflict resolution protocol has two components: the channel-access protocol (CAP) and the collision resolution algorithm (CRA).

The CAP is a distributed algorithm that determines, for each transmitter, when a newly arrived packet at that transmitter is sent for the first time. The simplest CAP, both conceptually and practically, is the *free-access protocol* in which a transmitter sends a new packet in the first slot following its arrival. The *blocked-access protocol* is that in which a transmitter sends a new packet in the first slot following the resolution of all collisions that had occured prior to the arrival of the packet.

The CRA can be defined as an algorithm (distributed in space and time) that organizes the retransmission of the colliding packets in such a way that every packet is eventually transmitted successfully with finite delay and all transmitters become aware of this fact.

The time span from the slot where an initial collision occurs up to and including the slot from which all transmitters recognize that all packets involved in the above initial collision have been successfully received is called collision resolution interval (CRI).

We consider blocked access which means that during the resolution of one particular collision, all users which were *not* involved in it are not allowed to transmit. This "blocking period" is called an *epoch*. Thus, an epoch consists of the time slots needed by the conflict resolution protocol to (at least partially) resolve one collision. The users involved in the initial collision will be called the *active users* of that epoch.

Clearly, because of the Poisson arrival of messages, i.e., of new users, message packets waiting for transmission will accumulate during the epoch. These packets will all be transmitted in the time slot following the epoch, together with packets arriving in that time slot, and possibly also packets which were not resolved previously. It is of course important that the maximum transmission delay, i.e., the maximal expected time between the generation and the successful transmission of a given packet, must be finite. (Because of the probabilistic (Poisson) model for the message arrivals, this is the best one can do.)

The supremum of the set of intensities λ for which a certain protocol still gives raise to a finite delay is called its *throughput*. The *capacity* of a certain collision channel is the supremum of achievable throughputs, taken over all possible protocols.

6.2 Tree Algorithms for Collision Resolution

Independently of each other, Capetanakis (1979), and Tsybakov and Mihailov (1978), (1980) introduced the first CRA which resulted in stable conflict resolution protocol. Capetanakis called it tree algorithm, while Tsybakov and Mihailov called it stack algorithm, so we abridge by TCRA.

Let N denote the number of active transmitters. According to the algorithm, all active transmitters send the packets in the next slot. If there was no active transmitter ($N = 0$) then the feedback is 0 and the algorithm terminates. If there was exactly one active transmitter ($N = 1$) then the feedback is 1 and the transmission was successful, so, again, the algorithm terminates. Otherwise $N \geq 2$, the feedback is the collision symbol $*$. After this collision, all transmitters involved flip a binary coin. Those flipping 0 retransmit in the next slot, those flipping 1 retransmit in the next slot after

the collision (if any) among those flipping 0 has been resolved.
Example. In order to illustrate the algorithm, assume that $N = 4$, and denote the four packets by A, B, C and D. Figure 6.1 shows the packets sent in a slot, when the flipping values are according to Table 6.1.

packets:	A B C D	B C	C	B	A D		A D	A D	A	D	
slots:	1	2	3	4	5	6	7	8	9	10	11

Figure 6.1: Packets sent according to TCRA

slot	packet	flipping values
1	A	1
1	B	0
1	C	0
1	D	1
2	B	1
2	C	0
5	A	1
5	D	1
7	A	0
7	D	0
8	A	0
8	D	1

Table 6.1: Flipping values

Observe that the algorithm doesn't terminates at slot 10. The reason is that no one can be sure that there was not other active transmitter, say E. This turns out only in slot 11.

The algorithm can be represented by a binary rooted search tree. This is why Capetanakis called it by tree algorithm. Collisions correspond to intermediate nodes, while empty slots and successful slots correspond to terminal nodes. Gallager (1985) introduced a simple implementation: when a transmitter flips 1 after a collision in which he is involved, he sets a counter to 1, then increments it by one for each subsequent collision slot and decrements by one for each subsequent collision-free slot. When the counter reaches 0, the transmitter retransmit in the next slot. Additionally, all active and non-active transmitters must know when the original collision (if any) has been resolved as this determines when new packets may be sent. For this purpose, it suffices for each transmitter to have a second counter which is set to 1 just prior to the first slot, then incremented by one for each subsequent collision slot and decremented by one for each subsequent collision-free slot. When this second counter reaches 0, the original collision (if any) has been resolved.

Tsybakov and Mihailov (1978), and Tsybakov and Vvedenskaya (1980) interpreted the same algorithm by a stack.

In order to analyze the tree algorithm, let X denote the number of packets sent in the first slot of the CRI, and let Y be the length (in slots) of the same CRI, i.e., the collision resolution time resolving X conflicts. Introduce the notation

$$L_N = \mathbf{E}\{Y|X = N\},$$

then L_N is the conditional expectation of the collision resolution time, given the multiplicity of the conflict N. Obviously,

$$L_0 = 1$$

and

$$L_1 = 1.$$

When $N \geq 2$, there is a collision in the first slot. If i of the colliding transmitters flip 0, then the expected CRI length is

$$1 + L_i + L_{N-i},$$

and the probability of this flipping is

$$\binom{N}{i} 2^{-N},$$

therefore

$$\begin{aligned} L_N &= 1 + \sum_{i=0}^{N} \binom{N}{i} 2^{-N} (L_i + L_{N-i}) = \\ &= 1 + \sum_{i=0}^{N} \binom{N}{i} 2^{-N} L_i + \sum_{i=0}^{N} \binom{N}{i} 2^{-N} L_{N-i} = \\ &= 1 + \sum_{i=0}^{N} \binom{N}{i} 2^{-N} L_i + \sum_{i=0}^{N} \binom{N}{N-i} 2^{-N} L_i = \\ &= 1 + 2 \sum_{i=0}^{N} \binom{N}{i} 2^{-N} L_i. \end{aligned}$$

Thus

$$L_N(1-2^{-N+1}) = 1 + 2^{-N+1}\sum_{i=0}^{N-1}\binom{N}{i}L_i. \tag{6.1}$$

Using this recursion we can calculate L_N, for example, Table 6.2 shows some figures. Massey (1981) bounded the oscillation of L_N/N. Hajek (1980) indicated first that L_N/N does not converge, and then Mathys and Flajolet (1985) proved its asymptotic behavior. Janssen and de Jong (2000) clarified the asymptotics of L_N/N:

N	L_N	L_N/N	$L(N)$
0	1		1
1	1	1	2.338
2	5	2.5	4.864
3	7.667	2.555	7.674
4	10.524	2.631	10.545
5	13.419	2.684	13.427
6	16.313	2.719	16.312
7	19.201	2.743	19.198
8	22.085	2.761	22.083
9	24.969	2.774	24.969
10	27.853	2.785	27.854

Table 6.2: Values of L_N for TCRA

$$L_N/N = \frac{2}{\ln 2} + A\sin(2\pi\log N + \varphi) + O(N^{-1}),$$

where

$$A = 3.127\cdot 10^{-6}, \qquad \varphi = 0.9826.$$

These imply that

$$2.8853869 \le \liminf_{N\to\infty}\frac{L_N}{N} \le \limsup_{N\to\infty}\frac{L_N}{N} \le 2.8853932.$$

Introduce the notation

$$L(z) = \sum_{i=0}^{\infty} L_i\frac{z^i}{i!}e^{-z}.$$

$L(z)$ is called the Poisson transform of the sequence $\{L_N\}$ (cf. Szpankowski (2001)). In the next sections we assume that new packets arrive according to a Poisson process, and in the analysis $L(z)$ plays an important role. If N is a random variable with Poisson(z) distribution then

$$L(z) = \mathbf{E}\{L_N\}.$$

For the tree algorithm, the coin flipping can be interpreted by the random arrival time, too. If a packet arrived in the interval $[0, 1]$ and this arrival time is uniformly distributed, then the bits of its binary expansion can be considered as flipping bits. Put $N = N_1 + N_2$, where N_1 and N_2 are independent random variables with Poisson($z/2$) distribution. N_1 and N_2 are the numbers of the arrivals in the first and second half of $[0, 1]$, respectively. Then for $N \leq 1$

$$L_N = 1,$$

otherwise

$$L_N = 1 + L_{N_1} + L_{N_2}.$$

From these relation we can derive an equation for $L(z)$:

$$\begin{aligned} L(z) &= \mathbf{E}\{L_N\} \\ &= \mathbf{E}\{I_{\{N\leq 1\}}\} + \mathbf{E}\{I_{\{N\geq 2\}}(1 + L_{N_1} + L_{N_2})\} \\ &= \mathbf{E}\{I_{\{N\leq 1\}}(1 - (1 + L_{N_1} + L_{N_2}))\} \\ &\quad + \mathbf{E}\{1 + L_{N_1} + L_{N_2}\} \\ &= -2\mathbf{E}\{I_{\{N\leq 1\}}\} + \mathbf{E}\{1 + L_{N_1} + L_{N_2}\} \\ &= -2(e^{-z} + ze^{-z}) + 1 + 2L(z/2) \end{aligned}$$

where we applied that for $N \leq 1$, $N_1 \leq 1$ and $N_2 \leq 1$, therefore $L_{N_1} = L_{N_2} = 1$. Thus

$$L(z) = 1 - 2(1 + z)e^{-z} + 2L(z/2). \qquad (6.2)$$

The recursive equation (6.2) gives an easy and quick way of calculation of $L(z)$ numerically.

Algorithm 6.1. *For any fixed $z > 0$ choose k_0 such that $\hat{z} := z/2^{k_0} < 10^{-5}$, and apply the following iteration:*

$$L(2^k\hat{z}) = 1 - 2(1 + 2^k\hat{z})e^{-2^k\hat{z}} + 2L(2^{k-1}\hat{z}), \qquad (1 \leq k \leq k_0)$$

then in the k_0^{th} step we get $L(z)$. In order to get a good initial value for $L(\hat{z})$,

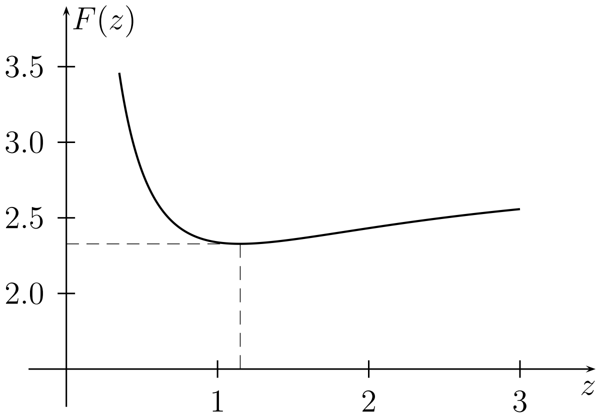

Figure 6.2: Values of $F(z)$ for small z

we use the second order Taylor polynomial approximation:

$$\begin{aligned}
L(\hat{z}) &= \left(L_0 + L_1\hat{z} + L_2\frac{\hat{z}^2}{2} + \cdots\right)\left(1 - \hat{z} + \frac{\hat{z}^2}{2}\cdots\right) \\
&\approx L_0 + (L_1 - L_0)\hat{z} + \left(\frac{L_2}{2} - L_1 + \frac{L_0}{2}\right)\hat{z}^2 \\
&= 1 + (1-1)\hat{z} + \left(\frac{5}{2} - 1 + \frac{1}{2}\right)\hat{z}^2 \\
&= 1 + 2\hat{z}^2.
\end{aligned}$$

Let us introduce

$$F(z) = \frac{L(z)}{z},$$

the proportional average number of time slots $F(z)$ required to resolve the conflict among colliding users in case of Poisson(z) collisions (cf. Figure 6.2).

$$\min_z F(z) = \min_z \frac{L(z)}{z} = \frac{L(1.15)}{1.15} = 2.32822469 \tag{6.3}$$

Using Algorithm 6.1, one can calculate $F(z)$ for $1 \leq z \leq 2$, and based on these values we give an algorithm to evaluate $F(z)$ for large z, and shows its oscillation.

Algorithm 6.2. *From the equation (6.2) it follows that*

$$F(2z) = \frac{1 - 2(1+2z)e^{-2z}}{2z} + F(z),$$

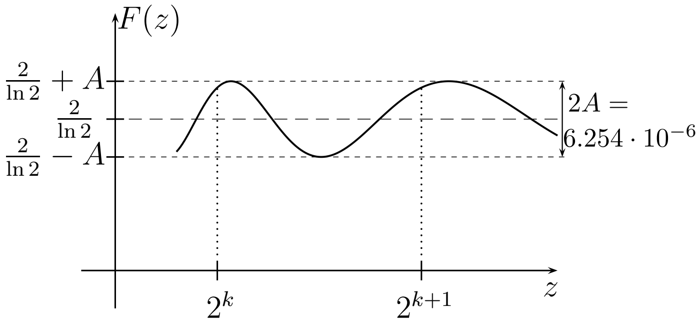

Figure 6.3: The oscillation in $F(z)$

so if $F(z)$ is given for $1 \le z \le 2$ (e.g., using Algorithm 6.1), then $F(2z)$ can be calculated for $1 \le z \le 2$, therefore $F(z)$ is given for $2 \le z \le 4$ this way. So, by induction

$$F\left(2^k z\right) = \sum_{i=1}^{k} \frac{1-2(1+2^i z)e^{-2^i z}}{2^i z} + F(z)$$

for any $k \ge 2$. For large enough k we can write that for any $1 \le z \le 2$

$$F\left(2^k z\right) \simeq G(z) + F(z), \tag{6.4}$$

where $k \ge k_0$, and

$$G(z) = \sum_{i=1}^{\infty} \frac{1-2(1+2^i z)e^{-2^i z}}{2^i z} \simeq \sum_{i=1}^{k_0} \frac{1-2(1+2^i z)e^{-2^i z}}{2^i z}.$$

If $k \ge k_0 = 30$, then this approximation error is of order 10^{-9}.

Equation (6.4) gives us an easy way of studying $F(z)$ (see Figure 6.3). We found that

$$2.8853869 \le \liminf_{z\to\infty} \frac{L(z)}{z} \le \limsup_{z\to\infty} \frac{L(z)}{z} \le 2.8853932. \tag{6.5}$$

In the next section we give a detailed analysis of the asymptotics (6.5).

Massey (1981) improved the TCRA. He observed that when a current slot is empty following a collision, all transmitters know that all the packets which collided in the past slot will be retransmitted in the future slot. Thus, all transmitters know in advance that the future slot will contain a collision, therefore it is wasteful actually to retransmit these packets in the future slot. The transmitters can "pretend" that this collision has taken place and immediately flip their binary coins and continue with the TCRA. We refer to this CRA as modified tree collision resolution algorithm (MTCRA). He proved that, for $N \geq 4$,

$$L^*_N \leq 2.664 \cdot N - 1,$$

and calculated some values of L^*_N:

N	L^*_N	L^*_N/N	$L^*(N)$
0	1		1
1	1	1	2.187
2	4.5	2.25	4.466
3	7	2.333	7.030
4	9.643	2.411	9.668
5	12.314	2.463	12.324
6	14.985	2.497	14.986
7	17.651	2.522	17.649
8	20.314	2.539	20.313
9	22.977	2.553	22.976
10	25.640	2.564	25.640

Table 6.3: Values of L^*_N for MTCRA

For the modified tree algorithm, the coin flipping can be interpreted also such that $N = N_1 + N_2$, where N_1 and N_2 are independent random variables with Poisson $(z/2)$ distribution. Then for $N \leq 1$

$$L^*_N = 1,$$

otherwise

$$L^*_N = 1 + L^*_{N_1} + L^*_{N_2} - I_{\{N_1=0\}}.$$

From these relation we can derive an equation for $L^*(z)$:

$$\begin{aligned}
L^*(z) &= \mathbf{E}\{L^*_N\} \\
&= \mathbf{E}\{I_{\{N\leq 1\}}\} + \mathbf{E}\{I_{\{N\geq 2\}}(1 + L^*_{N_1} + L^*_{N_2} - I_{\{N_1=0\}})\} \\
&= \mathbf{E}\{I_{\{N\leq 1\}}\} + \mathbf{E}\{I_{\{N\geq 2\}}(1 + L^*_{N_1} + L^*_{N_2})\} - \mathbf{E}\{I_{\{N\geq 2, N_1=0\}}\} \\
&= \mathbf{E}\{I_{\{N\leq 1\}}(1 - (1 + L^*_{N_1} + L^*_{N_2}))\} \\
&\quad + \mathbf{E}\{1 + L^*_{N_1} + L^*_{N_2}\} - \mathbf{E}\{I_{\{N_1+N_2\geq 2, N_1=0\}}\}
\end{aligned}$$

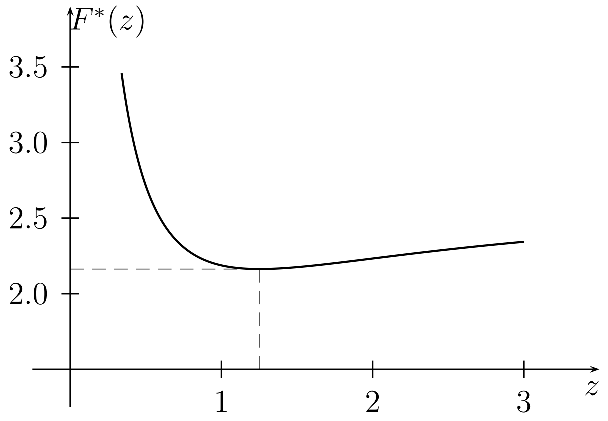

Figure 6.4: Values of $F^*(z)$ for small z

$$
\begin{aligned}
&= -2\mathbf{E}\{I_{\{N\le 1\}}\} + \mathbf{E}\{1 + L^*_{N_1} + L^*_{N_2}\} - \mathbf{E}\{I_{\{N_2\ge 2, N_1=0\}}\} \\
&= -2(e^{-z} + ze^{-z}) + 1 + 2L^*(z/2) - \left(1 - e^{-z/2} - \tfrac{z}{2}e^{-z/2}\right) e^{-z/2} \\
&= 1 - \left(1 + \tfrac{3}{2}z\right) e^{-z} - e^{-z/2} + 2L^*(z/2)
\end{aligned}
$$

where we applied that for $N \le 1$, $N_1 \le 1$ and $N_2 \le 1$, therefore $L_{N_1} = L_{N_2} = 1$. Thus

$$L^*(z) = 1 - \left(1 + \tfrac{3}{2}z\right) e^{-z} - e^{-z/2} + 2L^*(z/2). \tag{6.6}$$

Again, this gives a recursive way of calculating $L^*(z)$ if we take into account that for small values of z

$$L^*(z) \simeq 1 + 1.75z^2.$$

Let us introduce $F^*(z) = \frac{L^*(z)}{z}$. From equation (6.6) it follows that

$$F^*(2z) = \frac{1 - (1+3z)e^{-2z} - e^{-z}}{2z} + F^*(z),$$

so if $F^*(z)$ is given for $1 \le z \le 2$, then $F^*(2z)$ can be calculated for $1 \le z \le 2$, therefore $F^*(z)$ is given for $2 \le z \le 4$ this way (cf. Figure 6.4).

$$\min_z F^*(z) = \min_z \frac{L^*(z)}{z} = \frac{L^*(1.25)}{1.25} = 2.1632265 \tag{6.7}$$

Generally,

$$F^*\left(2^k z\right) = \sum_{i=1}^{k} \frac{1 - (1 + \frac{3}{2}2^i z)e^{-2^i z} - e^{-2^{i-1}z}}{2^i z} + F^*(z)$$

for any $k \geq 2$. As this sum is convergent, for large enough k we can write that

$$F^*\left(2^k z\right) \simeq G^*(z) + F^*(z), \tag{6.8}$$

where $1 \leq z \leq 2$ and $k \geq k_0$

$$G^*(z) \simeq \sum_{i=1}^{k_0} \frac{1 - (1 + \frac{3}{2} 2^i z) e^{-2^i z} - e^{-2^{i-1} z}}{2^i z}.$$

Again, for $k_0 = 30$, the approximation error is of order 10^{-9}. (6.8) gives us an easy way of studying $F^*(z) = \frac{L^*(z)}{z}$. We found that

$$2.664040 \leq \liminf_{z\to\infty} \frac{L^*(z)}{z} \leq \liminf_{z\to\infty} \frac{L^*(z)}{z} \leq 2.664045. \tag{6.9}$$

Similarly to $L(z)$, it is possible to express $L^*(z)$ by a non-recursive formula, too.

$$\begin{aligned} L^*(z) &= 1 + \sum_{j=2}^{\infty} (-1)^j \frac{z^j}{j!} \frac{\frac{3}{2} j - 1 - 2^{-j}}{1 - 2^{1-j}} \\ &= (1+z) e^{-z} + \sum_{N=2}^{\infty} \frac{z^N}{N!} e^{-z} \left(1 + \sum_{j=2}^{N} (-1)^j \binom{N}{j} \frac{\frac{3}{2} j - 1 - 2^{-j}}{1 - 2^{1-j}} \right) \end{aligned}$$

from which we get a closed formula for L_N^*. $L_N^* = 1$ if $N \leq 1$ and

$$L_N^* = 1 + \sum_{j=2}^{N} (-1)^j \binom{N}{j} \frac{\frac{3}{2} j - 1 - 2^{-j}}{1 - 2^{1-j}}$$

if $N \geq 2$.

If the multiplicity of the collision is large then the tree has many intermediate nodes, so L_N can be decreased if the tree has Q-ary root, otherwise has binary nodes. Given N, let $L_N^{(Q)}$ denote the conditional expectation of the collision resolution time, and let $L^{(Q)}(z)$ be the Poisson transform of the sequence $\{L_N^{(Q)}\}$. Similarly to the previous calculation, if N is a random variable with Poisson (z) distribution then the coin flipping can be interpreted such that $N = N_1 + \cdots + N_Q$, where $N_1, \ldots, N_Q$ are independent random variables with Poisson (z/Q) distribution. Then for $N \leq 1$

$$L_N^{(Q)} = 1,$$

otherwise

$$L_N^{(Q)} = 1 + L_{N_1} + \cdots + L_{N_Q},$$

therefore

$$
\begin{aligned}
&L^{(Q)}(z)\\
&\quad = \mathbf{E}\{L_N^{(Q)}\}\\
&\quad = \mathbf{E}\{I_{\{N\le 1\}}\} + \mathbf{E}\{I_{\{N\ge 2\}}(1 + L_{N_1} + \cdots + L_{N_Q})\}\\
&\quad = \mathbf{E}\{I_{\{N\le 1\}}(1 - (1 + L_{N_1} + \cdots + L_{N_Q}))\} + \mathbf{E}\{1 + L_{N_1} + \cdots + L_{N_Q}\}\\
&\quad = -Q\mathbf{E}\{I_{\{N\le 1\}}\} + \mathbf{E}\{1 + L_{N_1} + \cdots + L_{N_Q}\}\\
&\quad = -Q(e^{-z} + ze^{-z}) + 1 + QL(z/Q).
\end{aligned}
$$

It implies that

$$L^{(Q)}(z) = 1 - Q(1+z)e^{-z} + QL(z/Q). \tag{6.10}$$

6.3 The oscillation of $L(z)/z$ and of $L_N - L(N)$

Hajek (1980) indicated first that L_N/N does not converge, Massey (1981) bounded the oscillation of L_N/N, and then Mathys and Flajolet (1985) showed its asymptotic behavior in an implicit way. Janssen and de Jong (2000) clarified the exact asymptotics of L_N/N:

$$L_N/N = \frac{2}{\ln 2} + A\sin(2\pi \log_2 N + \varphi) + O(N^{-1}),$$

where

$$A = 3.127 \cdot 10^{-6}, \qquad \varphi = 0.9826.$$

These imply that

$$2.8853869 \le \liminf_{N\to\infty} \frac{L_N}{N} \le \limsup_{N\to\infty} \frac{L_N}{N} \le 2.8853932.$$

The question naturally arises whether

$$\limsup_{N\to\infty} \frac{L_N}{N} \overset{?}{=} \limsup_{z\to\infty} \frac{L(z)}{z}$$

Firstly we show a result of Janssen and de Jong (2000) which is on the oscillation of $\frac{L_N}{N}$. In its proof a similar technique is used as in the proof of Theorem 6.1 in the sequel.

Lemma 6.1 (Janssen and de Jong (2000)).

$$\frac{L(z)}{z} \simeq \frac{2}{\ln 2} + A\sin(2\pi \log_2 z + \varphi), \tag{6.11}$$

where

$$A = 3.127 \cdot 10^{-6}, \qquad \varphi = 0.9826.$$

Proof. Janssen and de Jong (2000) proved this lemma by Fourier analysis. Here we give a different proof. The technique being used here is by Mellin transform, and it will be applied also in the proof of Theorem 6.1. (Reader can find an excellent survey on Mellin transform in Flajolet et al. (1995), and some application of Mellin transform to similar problems in Knuth (1973) pages 131–134, and Jacquet, Regnier (1986).)

Based on Gulko and Kaplan (1985), Mathys and Flajolet (1985) the formula for L_N can be written in a nonrecursive way: $L_0 = L_1 = 1$, and for $N \geq 2$,

$$L_N = 1 + 2\sum_{j=0}^{\infty} \left(2^j \left(1 - (1-2^{-j})^N\right) - N(1-2^{-j})^{N-1}\right) \tag{6.12}$$

Actually, it can be easily seen as follows.

$$1 - (1-2^{-j})^N - N2^{-j}(1-2^{-j})^{N-1}$$

is the probability of two or more successes in N Bernoulli trials, where the probability of success is 2^{-j}. This is the probability that a particular node of the tree at level j contains two or more active users. The average number of such nodes at level j is then

$$2^j \left(1 - (1-2^{-j})^N - N2^{-j}(1-2^{-j})^{N-1}\right),$$

and each of them has two children at level $j+1$. Average number of nodes in the tree can be calculated by summing up for all possible j this quantity multiplied by two plus 1 for the root node.

Let us calculate the Poisson transform of L_N. By (6.12)

$$\begin{aligned} L(z) &= \sum_{N=0}^{\infty} L_N \frac{z^N}{N!} e^{-z} \\ &= \sum_{N=0}^{\infty} \frac{z^N}{N!} e^{-z} + 2\sum_{N=2}^{\infty}\sum_{j=0}^{\infty} \left(2^j \left(1 - (1-2^{-j})^N\right) - N(1-2^{-j})^{N-1}\right) \frac{z^N}{N!} e^{-z} \end{aligned}$$

$$
\begin{aligned}
&= 1+2\sum_{j=0}^{\infty}2^j\sum_{N=2}^{\infty}\left(1-(1-2^{-j})^N-N2^{-j}(1-2^{-j})^{N-1}\right)\frac{z^N}{N!}e^{-z}\\
&= 1+2\sum_{j=0}^{\infty}2^j\Bigg((e^z-1-z)-\left(e^{(1-2^{-j})z}-1-(1-2^{-j})z\right)\\
&\quad -\sum_{N=2}^{\infty}N2^{-j}(1-2^{-j})^{N-1}\frac{z^N}{N!}\Bigg)e^{-z}\\
&= 1+2\sum_{j=0}^{\infty}2^j\left(e^z-e^{(1-2^{-j})z}-2^{-j}z-\sum_{N=1}^{\infty}2^{-j}z(1-2^{-j})^N\frac{z^N}{N!}\right)e^{-z}\\
&= 1+2\sum_{j=0}^{\infty}2^j\left(e^z-e^{(1-2^{-j})z}-2^{-j}z-2^{-j}z\left(e^{(1-2^{-j})z}-1\right)\right)e^{-z}\\
&= 1+2\sum_{j=0}^{\infty}2^j\left(e^z-e^{(1-2^{-j})z}-2^{-j}ze^{(1-2^{-j})z}\right)e^{-z}\\
&= 1+2\sum_{j=0}^{\infty}2^j\left(1-e^{-2^{-j}z}-2^{-j}ze^{-2^{-j}z}\right) \qquad (6.13)
\end{aligned}
$$

We use the Mellin transform technique (cf. (2001)). The Mellin transform of a complex valued function $f(x)$ defined over positive reals is

$$\mathcal{M}[f(x);s]=F(s)=\int_0^{\infty}x^{s-1}f(x)\,dx, \qquad a<\Re(s)<b$$

where (a,b) is the fundamental (convergence) strip and $\Re(\cdot)$ ($\Im(\cdot)$) denotes the real (imaginary) part of its argument. The inversion formula is

$$f(x)=\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}x^{-s}F(s)\,ds, \qquad a<c<b,$$

where c is an arbitrary real number from the fundamental strip (a,b). The following Mellin transforms will be used.

$$\mathcal{M}[1-e^{-x}-xe^{-x};s]=-(s+1)\Gamma(s), \qquad -2<\Re(s)<0, \qquad (6.14)$$

where $\Gamma(\cdot)$ denotes the complete gamma function that is in Euler's limit form (cf. Szpankowski (2001) page 41):

$$\Gamma(s)=\lim_{n\to\infty}\frac{n^s n!}{s(s+1)(s+2)\cdots(s+n)}. \qquad (6.15)$$

Equation (6.14) can be verified by applying the inversion integral or if we consider some elementary Mellin transforms (cf. Szpankowki (2001) page 401):

$$\begin{aligned} \mathcal{M}[e^{-x}; s] &= \Gamma(s), \qquad 0 < \Re(s) < \infty, & (6.16)\\ \mathcal{M}[e^{-x} - 1; s] &= \Gamma(s), \qquad -1 < \Re(s) < 0. & (6.17) \end{aligned}$$

One of the basic properties of the Mellin transform is that if

$$\mathcal{M}[f(x); s] = F(s), \qquad a < \Re(s) < b,$$

then

$$\mathcal{M}[\alpha x^{\beta} f(\gamma x); s] = \alpha\gamma^{-s} F(s+\beta), \qquad a - \beta < \Re(s) < b - \beta. \tag{6.18}$$

That is why from (6.16) and (6.18) we have

$$\mathcal{M}[xe^{-x}; s] = \Gamma(s+1), \qquad -1 < \Re(s) < \infty. \tag{6.19}$$

Equation (6.14) follows from (6.17), (6.19) and from the fact that it does not have a pole at -1.

So, by using the Mellin transform on (6.13) we get that

$$\begin{aligned} \mathcal{M}[L(z) - 1; s] &= 2\sum_{j=0}^{\infty} 2^j \mathcal{M}[1 - e^{-2^{-j}z} - 2^{-j}ze^{-2^{-j}z}; s]\\ &= 2\sum_{j=0}^{\infty} 2^j \left(2^{-j}\right)^{-s} (-(s+1)\Gamma(s))\\ &= 2\sum_{j=0}^{\infty} \left(2^{s+1}\right)^j (-(s+1)\Gamma(s))\\ &= -2\frac{(s+1)\Gamma(s)}{1-2^{s+1}}, \end{aligned}$$

where $-2 < \Re(s) < -1$ (in the last step $\Re(s) < -1$ is needed for the convergence). Let us choose $c := -3/2$. From the inversion formula it follows that

$$\begin{aligned} L(z) - 1 &= \frac{1}{2\pi i}\int\limits_{c-i\infty}^{c+i\infty} z^{-s}\left(-2\frac{(s+1)\Gamma(s)}{1-2^{s+1}}\right) ds\\ &= -\frac{2}{2\pi i}\int\limits_{c-i\infty}^{c+i\infty} \frac{(s+1)\Gamma(s)z^{-s}}{1-2^{s+1}}\, ds & (6.20) \end{aligned}$$

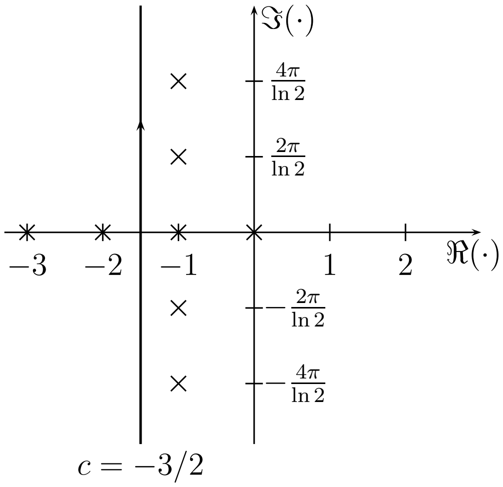

Figure 6.5: Poles of $\mathcal{M}[L(z) - 1; s]$ and the line integral for the inversion formula

The line integral in (6.20) can be evaluated by using Cauchy's residue theorem (cf. Figure 6.5). For this calculation some residues are needed. $\Gamma(s)$ has simple poles at $s = 0, -1, -2, \ldots$. The residue in the pole at 0 is

$$\begin{aligned} \operatorname*{res}_{s=0} \Gamma(s) &= \lim_{s \to 0} s\Gamma(s) \\ &= \lim_{s \to 0} \Gamma(s+1) \\ &= 1. \end{aligned}$$

$\frac{1}{1-2^{s+1}}$ has simple poles at the roots of equation $2^{s+1} = 1$, so if $s + 1 = \frac{\ln 1 + 2k\pi i}{\ln 2}$, namely $s = -1 + \frac{2k\pi i}{\ln 2}$

$$\begin{aligned} \operatorname*{res}_{s=s_0} \frac{1}{1-2^{s+1}} &= \lim_{s \to s_0} \frac{1}{(1-2^{s+1})'} \\ &= \lim_{s \to s_0} \frac{1}{-2^{s+1} \ln 2} \\ &= -\frac{1}{\ln 2}, \end{aligned}$$

for all $s_0 \in \left\{-1 + \frac{2k\pi i}{\ln 2},\ k \in \mathbb{Z}\right\}$, and

$$\begin{aligned} \operatorname*{res}_{s=-1} \frac{(s+1)\Gamma(s)}{1-2^{s+1}} &= \operatorname*{res}_{s=-1} \frac{\Gamma(s+2)}{s(1-2^{s+1})} \\ &= -\operatorname*{res}_{s=-1} \frac{1}{1-2^{s+1}} \end{aligned}$$

$$= \frac{1}{\ln 2}.$$

If we close the integration contour of the inversion integral in the right half plane (and negate the result because of the negative direction of the integration contour), we get

$$\begin{aligned}
L(z) - 1 &= -\frac{2}{2\pi i}\int\limits_{c-i\infty}^{c+i\infty} \frac{(s+1)\Gamma(s)z^{-s}}{1-2^{s+1}}\,ds \\
&= \frac{2}{2\pi i}\Biggl(2\pi i \operatorname*{res}_{s=0} \frac{(s+1)\Gamma(s)z^{-s}}{1-2^{s+1}} \\
&\qquad + 2\pi i \sum_{k=-\infty}^{\infty} \operatorname*{res}_{s=-1+\frac{2k\pi i}{\ln 2}} \frac{(s+1)\Gamma(s)z^{-s}}{1-2^{s+1}}\Biggr) \\
&= 2\left(\frac{1}{1-2} + \frac{z}{\ln 2} + \sum_{k\neq 0} \frac{\frac{2k\pi i}{\ln 2}\Gamma\left(-1+\frac{2k\pi i}{\ln 2}\right) z^{1-\frac{2k\pi i}{\ln 2}}}{-\ln 2}\right) \\
&= -2 + \frac{2z}{\ln 2} - \frac{2z}{\ln 2}\sum_{k\neq 0} \tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right) z^{-\frac{2k\pi i}{\ln 2}} \\
&= -2 + \frac{2z}{\ln 2}\Biggl(1 - \sum_{k\neq 0} \tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2 z}\Biggr),
\end{aligned}$$

therefore

$$L(z) = \frac{2z}{\ln 2}\Biggl(1 - \sum_{k\neq 0} \tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2 z}\Biggr) - 1 \tag{6.21}$$

As the gamma function decays exponentially fast over imaginary lines, for $\frac{L(z)}{z}$ a sharp approximation can be given by (6.21) if we take into account just the first two terms (for $k = \pm 1$) of the sum, i.e., the approximation error is of order 10^{-12}.

$$\frac{L(z)}{z} \approx \frac{2}{\ln 2}\Biggl(1 - \sum_{k=\pm 1} \tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2 z}\Biggr) - \frac{1}{z}.$$

The last term is negligible asymptotically, so

$$\frac{L(z)}{z} \approx \frac{2}{\ln 2}\Biggl(1 - \sum_{k=\pm 1} \tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2 z}\Biggr)$$

$$
\begin{aligned}
&= \frac{2}{\ln 2}\Bigg(1-\sum_{k=\pm 1}\tfrac{2k\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2k\pi i}{\ln 2}\right)\cdot \\
&\quad \cdot\left(\cos\left(-2k\pi\log_2 z\right)+i\sin\left(-2k\pi\log_2 z\right)\right)\Bigg) \\
&= \frac{2}{\ln 2}\Bigg(1+\tfrac{2\pi i}{\ln 2}\Gamma\left(-1-\tfrac{2\pi i}{\ln 2}\right)\left(\cos\left(2\pi\log_2 z\right)+i\sin\left(2\pi\log_2 z\right)\right) \\
&\quad -\tfrac{2\pi i}{\ln 2}\Gamma\left(-1+\tfrac{2\pi i}{\ln 2}\right)\left(\cos\left(2\pi\log_2 z\right)-i\sin\left(2\pi\log_2 z\right)\right)\Bigg) \\
&= \frac{2}{\ln 2}\Bigg(1+\tfrac{2\pi}{\ln 2}\Big(i\left(\Gamma\left(-1-\tfrac{2\pi i}{\ln 2}\right)-\Gamma\left(-1+\tfrac{2\pi i}{\ln 2}\right)\right)\cos\left(2\pi\log_2 z\right) \\
&\quad -\left(\Gamma\left(-1-\tfrac{2\pi i}{\ln 2}\right)+\Gamma\left(-1+\tfrac{2\pi i}{\ln 2}\right)\right)\sin\left(2\pi\log_2 z\right)\Big)\Bigg)
\end{aligned}
$$

Using the well-known equality $\Gamma(\overline{z})=\overline{\Gamma(z)}$, and let

$$\Gamma\left(-1+\tfrac{2\pi i}{\ln 2}\right)=x'+iy'=-3.31727\cdot 10^{-8}+4.973654\cdot 10^{-8}i,$$

then we get

$$
\begin{aligned}
\frac{L(z)}{z} &\approx \frac{2}{\ln 2}\Big(1+\tfrac{4\pi}{\ln 2}\left(y'\cos\left(2\pi\log_2 z\right)-x'\sin\left(2\pi\log_2 z\right)\right)\Big) && (6.22)\\
&= \frac{2}{\ln 2}\left(1+\tfrac{4\pi}{\ln 2}\sqrt{x'^2+y'^2}\sin\left(2\pi\log_2 z+\operatorname{arctg}\left(-\tfrac{y'}{x'}\right)\right)\right) \\
&= \frac{2}{\ln 2}+A\sin(2\pi\log_2 z+\varphi), && (6.23)
\end{aligned}
$$

where

$$A=3.127\cdot 10^{-6},\qquad \varphi=0.9826.$$

□

Mathys (1984) proved that $L_N-L(N)=O(1)$. Next we extend it showing its oscillation.

Theorem 6.1.

$$L_N-L(N)\simeq A\cos(2\pi\log_2 N+\varphi),$$

where

$$A=1.29\cdot 10^{-4},\qquad \varphi=0.698.$$

Proof. By (6.12) and (6.13),

$$\begin{aligned}
L_N - L(N) &= 1 + 2\sum_{j=0}^{\infty} \left(2^j \left(1 - (1-2^{-j})^N\right) - N(1-2^{-j})^{N-1}\right) \\
&\quad - 1 - 2\sum_{j=0}^{\infty} \left(2^j(1 - e^{-2^{-j}N}) - Ne^{-2^{-j}N}\right) \\
&= 2\sum_{j=0}^{\infty} 2^j \left(e^{-2^{-j}N} - (1-2^{-j})^N\right) \\
&\quad + 2\sum_{j=0}^{\infty} N \left(e^{-2^{-j}N} - (1-2^{-j})^{N-1}\right) \\
&= 2\sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \left(1 - \left(e^{2^{-j}}(1-2^{-j})\right)^N\right) \\
&\quad + 2\sum_{j=0}^{\infty} N \left(e^{-2^{-j}} e^{-2^{-j}(N-1)} - (1-2^{-j})^{N-1}\right) \\
&=: 2A + 2B
\end{aligned}$$

For getting lower and upper bounds we use the following inequalities. If $0 \le x \le 1$, then

$$\begin{array}{rcccl}
1 + x + \frac{x^2}{2} & \le & e^x & \le & 1 + x + \frac{x^2}{2} + \frac{x^3}{2} \\
1 - \frac{x^2}{2} - \frac{x^3}{2} & \le & e^x(1-x) & \le & 1 - \frac{x^2}{2} - \frac{x^4}{4} \ \le \ 1 - \frac{x^2}{2} \\
1 - x & \le & e^{-x} & \le & 1 - x + \frac{x^2}{2}
\end{array}$$

and if $a > b \ge 0$, then

$$(a-b)Nb^{N-1} \ \le \ a^N - b^N \ \le \ (a-b)Na^{N-1}.$$

Lower bound:

$$\begin{aligned}
A &\ge \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \left(1 - \left(1 - \frac{2^{-2j}}{2}\right)^N\right) \\
&\ge \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \frac{2^{-2j}}{2} N \left(1 - \frac{2^{-2j}}{2}\right)^{N-1} \\
&\ge \frac{1}{2}\sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} \left(1 - (N-1)\frac{2^{-2j}}{2}\right)
\end{aligned}$$

$$
\begin{aligned}
&\geq \frac{1}{2}\sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N}\left(1 - N\frac{2^{-2j}}{2}\right)\\
&= \frac{1}{2}\sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} - \frac{1}{4}\sum_{j=0}^{\infty} 2^{-3j} N^2 e^{-2^{-j}N}\\
&=: I_1 + I_2,
\end{aligned}
$$

and

$$
\begin{aligned}
B &\geq \sum_{j=0}^{\infty} N\left((1-2^{-j})e^{-2^{-j}(N-1)} - (1-2^{-j})^{N-1}\right)\\
&= \sum_{j=0}^{\infty} N e^{-2^{-j}(N-1)}\left(1 - \left(e^{2^{-j}}(1-2^{-j})\right)^{N-1}\right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)}\\
&\geq \sum_{j=0}^{\infty}(N-1) e^{-2^{-j}(N-1)}\left(1 - \left(1 - \frac{2^{-2j}}{2}\right)^{N-1}\right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)}\\
&\geq \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j}(N-1)^2 e^{-2^{-j}(N-1)}\left(1 - \frac{2^{-2j}}{2}\right)^{N-1} - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)}\\
&\geq \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j}(N-1)^2 e^{-2^{-j}(N-1)}\left(1 - (N-1)\frac{2^{-2j}}{2}\right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)}\\
&= \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j}(N-1)^2 e^{-2^{-j}(N-1)} - \frac{1}{4}\sum_{j=0}^{\infty} 2^{-4j}(N-1)^3 e^{-2^{-j}(N-1)}\\
&\quad - \sum_{j=0}^{\infty} 2^{-j}(N-1) e^{-2^{-j}(N-1)} - \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)}\\
&=: J_1 + J_2 + J_3 + J_4.
\end{aligned}
$$

Upper bound:

$$
\begin{aligned}
A &\leq \sum_{j=0}^{\infty} 2^{j} e^{-2^{-j}N}\left(1 - \left(1 - \frac{2^{-2j}}{2} - \frac{2^{-3j}}{2}\right)^{N}\right)\\
&\leq \sum_{j=0}^{\infty} 2^{j} e^{-2^{-j}N}\frac{2^{-2j}}{2}(1+2^{-j})N\\
&= \frac{1}{2}\sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N}(1+2^{-j})
\end{aligned}
$$

$$\begin{aligned} &= \frac{1}{2}\sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} + \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j} N e^{-2^{-j}N} \\ &=: I_1 + \widetilde{I}_2, \end{aligned}$$

and

$$\begin{aligned} B &\leq \sum_{j=0}^{\infty} N\left(\left(1 - 2^{-j} + \frac{2^{-2j}}{2}\right) e^{-2^{-j}(N-1)} - (1-2^{-j})^{N-1}\right) \\ &= \sum_{j=0}^{\infty} N\left(e^{-2^{-j}(N-1)} - (1-2^{-j})^{N-1}\right) \\ &\quad -\sum_{j=0}^{\infty} 2^{-j} N\left(1 - \frac{2^{-j}}{2}\right) e^{-2^{-j}(N-1)} \\ &= \sum_{j=0}^{\infty} N e^{-2^{-j}(N-1)}\left(1 - \left(e^{2^{-j}}(1-2^{-j})\right)^{N-1}\right) \\ &\quad -\sum_{j=0}^{\infty} 2^{-j} N\left(1 - \frac{2^{-j}}{2}\right) e^{-2^{-j}(N-1)} \\ &\leq \sum_{j=0}^{\infty} N e^{-2^{-j}(N-1)}\left(1 - \left(1 - \frac{2^{-2j}}{2} - \frac{2^{-3j}}{2}\right)^{N-1}\right) \\ &\quad -\sum_{j=0}^{\infty} 2^{-j}(N-1)\left(1 - \frac{2^{-j}}{2}\right) e^{-2^{-j}(N-1)} \\ &\leq \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j} N(N-1)(1+2^{-j}) e^{-2^{-j}(N-1)} \\ &\quad -\sum_{j=0}^{\infty} 2^{-j}(N-1)\left(1 - \frac{2^{-j}}{2}\right) e^{-2^{-j}(N-1)} \\ &= \frac{1}{2}\sum_{j=0}^{\infty} 2^{-2j}(N-1)^2 e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-2j}(N-1) e^{-2^{-j}(N-1)} \\ &\quad + \frac{1}{2}\sum_{j=0}^{\infty} 2^{-3j}(N-1)^2 e^{-2^{-j}(N-1)} + \frac{1}{2}\sum_{j=0}^{\infty} 2^{-3j}(N-1) e^{-2^{-j}(N-1)} \\ &\quad -\sum_{j=0}^{\infty} 2^{-j}(N-1) e^{-2^{-j}(N-1)} \end{aligned}$$

$$=: \quad J_1 + 2\widetilde{J}_2 + \widetilde{J}_3 + \widetilde{J}_4 + J_3$$

It can be shown that if $N \to \infty$, then $I_2, \widetilde{I}_2, J_2, J_4, \widetilde{J}_2, \widetilde{J}_3, \widetilde{J}_4$ all tend to 0. Both from the upper and the lower bounds the same terms remain, so we have derived the following asymptotical equation which should be analyzed further.

$$L_N - L(N) = 2(A+B) = 2(I_1 + J_1 + J_3) + o(1)$$

It can be further simplified by showing that $2I_1 + J_3 \to 0$ if $N \to \infty$. Let us lower bound it firstly,

$$\begin{aligned} 2I_1 + J_3 &= \sum_{j=0}^{\infty} 2^{-j} N \left(e^{-2^{-j}} - 1\right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)} \\ &\geq \sum_{j=0}^{\infty} 2^{-j} N \left(1 - 2^{-j} - 1\right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)} \\ &= -\sum_{j=0}^{\infty} 2^{-2j} N e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)} \end{aligned}$$

and then upper bound it

$$\begin{aligned} 2I_1 + J_3 &\leq \sum_{j=0}^{\infty} 2^{-j} N \left(1 - 2^{-j} + \frac{2^{-2j}}{2} - 1\right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)} \\ &= -\sum_{j=0}^{\infty} 2^{-2j} N \left(1 - \frac{2^{-j}}{2}\right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)}. \end{aligned}$$

Notice that both bounds tend to 0. That is why $L_N - L(N)$ asymptotically equals to

$$\begin{aligned} L_N - L(N) &= 2J_1 + J_3 + o(1) \\ &= \sum_{j=0}^{\infty} 2^{-2j} (N-1)^2 e^{-2^{-j}(N-1)} - \sum_{j=0}^{\infty} 2^{-j} (N-1) e^{-2^{-j}(N-1)} + o(1) \\ &=: \Delta(N-1) + o(1) \end{aligned}$$

By using Mellin transform technique the difference $\Delta(N)$ can be expressed in terms of the gamma function. From (6.16) and (6.18) it follows that

$$\mathcal{M}[x^2 e^{-x}; s] = \Gamma(s+2), \qquad -2 < \Re(s) < \infty.$$

Thus, with also using (6.19) we have

$$\begin{aligned}
\mathcal{M}[\Delta(N);s] &= \sum_{j=0}^{\infty}\mathcal{M}[2^{-2j}N^2e^{-2^{-j}N};s]-\sum_{j=0}^{\infty}\mathcal{M}[2^{-j}Ne^{-2^{-j}N};s]\\
&= \sum_{j=0}^{\infty}\left(2^{-j}\right)^{-s}\Gamma(s+2)-\sum_{j=0}^{\infty}\left(2^{-j}\right)^{-s}\Gamma(s+1)\\
&= \frac{\Gamma(s+2)}{1-2^s}-\frac{\Gamma(s+1)}{1-2^s},
\end{aligned}$$

where $-1 < \Re(s) < 0$ (in the last step $\Re(s) < 0$ is needed for the convergence). Let us choose $c := -1/2$. From the inversion formula it follows that

$$\Delta(N)=\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}N^{-s}\left(\frac{\Gamma(s+2)}{1-2^s}-\frac{\Gamma(s+1)}{1-2^s}\right)ds$$

This line integral can be evaluated by using Cauchy's residue theorem (cf. Figure 6.6). If we close the integration contour of the inversion integral in the right half plane (and negate the result because of the negative direction of the integration contour), we get

$$\Delta(N)=-\frac{1}{2\pi i}\left(2\pi i\sum_{k=-\infty}^{\infty}\operatorname*{res}_{s=\frac{2k\pi i}{\ln 2}}N^{-s}\left(\frac{\Gamma(s+2)}{1-2^s}-\frac{\Gamma(s+1)}{1-2^s}\right)\right)$$

$$\begin{aligned}
L_N-L(N) &= \Delta(N-1)+o(1)\\
&\simeq \frac{1}{\ln 2}+\frac{1}{\ln 2}\sum_{k\neq 0}\Gamma\left(2+\tfrac{2k\pi i}{\ln 2}\right)e^{-2k\pi i\log_2(N-1)}\\
&\quad -\left(\frac{1}{\ln 2}+\frac{1}{\ln 2}\sum_{k\neq 0}\Gamma\left(1+\tfrac{2k\pi i}{\ln 2}\right)e^{-2k\pi i\log_2(N-1)}\right)\\
&= \frac{1}{\ln 2}\sum_{k\neq 0}\left(\Gamma\left(2+\tfrac{2k\pi i}{\ln 2}\right)-\Gamma\left(1+\tfrac{2k\pi i}{\ln 2}\right)\right)e^{-2k\pi i\log_2(N-1)}
\end{aligned}$$

Approximation by only the terms of $k = \pm 1$ is perfect, i.e., the approximation error is of order 10^{-9}.

$$L_N-L(N)\simeq A\cos(2\pi\log_2(N-1)+\varphi),$$

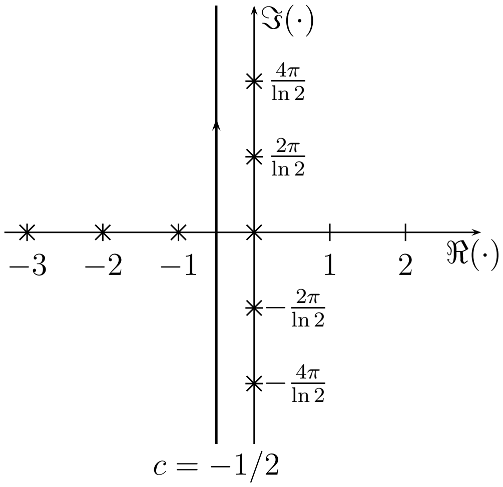

Figure 6.6: Poles of $\mathcal{M}[\Delta(N); s]$ and the line integral for the inversion formula

where

$$\begin{aligned} A &= \frac{2}{\ln 2}\Bigg(\Big(\Re\left(\Gamma\left(2+\tfrac{2\pi i}{\ln 2}\right)-\Gamma\left(1+\tfrac{2\pi i}{\ln 2}\right)\right)\Big)^2 \\ &\quad +\Big(\Im\left(\Gamma\left(2+\tfrac{2\pi i}{\ln 2}\right)-\Gamma\left(1+\tfrac{2\pi i}{\ln 2}\right)\right)\Big)^2\Bigg)^{1/2} \\ &= 1.29\cdot 10^{-4}, \end{aligned}$$

and

$$\varphi = \operatorname{arctg}\frac{\Im\left(\Gamma\left(2+\frac{2\pi i}{\ln 2}\right)-\Gamma\left(1+\frac{2\pi i}{\ln 2}\right)\right)}{\Re\left(\Gamma\left(2+\frac{2\pi i}{\ln 2}\right)-\Gamma\left(1+\frac{2\pi i}{\ln 2}\right)\right)} = 0.698.$$

Thus

$$|L_N - L(N)| \lesssim A = 1.29\cdot 10^{-4}$$

In order to finish the proof of Theorem 6.1 we have to show that if $N \to \infty$

$$\cos(2\pi\log_2 N+\varphi)-\cos(2\pi\log_2(N-1)+\varphi)=o(1).$$

It is easy since

$$\begin{aligned} &\cos(2\pi\log_2 N+\varphi)-\cos(2\pi\log_2(N-1)+\varphi) \\ &\quad = (\cos(2\pi\log_2 N)-\cos(2\pi\log_2(N-1)))\cos\varphi \qquad (6.24) \\ &\qquad - (\sin(2\pi\log_2 N)-\sin(2\pi\log_2(N-1)))\sin\varphi \qquad (6.25) \end{aligned}$$

Then the first term of (6.24) can be written as

$$\begin{aligned}
&\cos(2\pi \log_2 N) - \cos(2\pi \log_2(N-1)) \\
&\quad = -2\sin\left(\pi\left(\log_2 N + \log_2(N-1)\right)\right)\sin\left(\pi\left(\log_2 N - \log_2(N-1)\right)\right) \\
&\quad = -2\sin\left(\pi \log_2(N(N-1))\right)\sin\left(\pi \log_2\left(1+\tfrac{1}{N-1}\right)\right)
\end{aligned}$$

As the function $\log_2(\cdot)$ is continuous in 1 and function $\sin(\cdot)$ is continuous in 0

$$\lim_{N\to\infty} \sin\left(\pi \log_2\left(1+\tfrac{1}{N-1}\right)\right) = 0,$$

that is why (6.24) is $o(1)$. With similar reasoning it can be easily seen that (6.25) is also $o(1)$. So, we have proved that

$$L_N - L(N) \simeq A\cos(2\pi \log_2 N + \varphi).$$

□

Notice, that Theorem 6.1 implies that if $N \to \infty$, then $\frac{L_N - L(N)}{N^\alpha} \to 0$, for any $\alpha > 0$, and so

$$\limsup_{N\to\infty} \frac{L_N}{N} = \limsup_{z\to\infty} \frac{L(z)}{z}.$$

6.4 Capetanakis Algorithm

Next we show three blocked access channel access protocols, three CAPs, each of them apply TCRA or MTCRA.

Partition the time axis into collision resolution intervals (CRI) of random length. The length of the first CRI is the collision resolution time of the packets arrived in the first slot. The length of the n^{th} CRI is the collision resolution time of the packets arrived in the $(n-1)^{\text{th}}$ CRI.

Theorem 6.2. *For the Capetanakis-algorithm, assume that the arrivals of the packets are according to a Poisson process $\{Z(t) : t \geq 0\}$ with intensity $\lambda > 0$. If*

$$\lambda < \frac{1}{\limsup\limits_{z\to\infty} \frac{L(z)}{z}},$$

then the sequence of lengths of CRIs forms a stable Markov chain.

Proof. Let X_n be the number of packets arrived in the n-th CRI, and let Y_n be the length of the n-th CRI. It is easy to see that $\{Y_n\}$ is a homogeneous, irreducible and aperiodic Markov chain, so the Foster Theorem (Theorem B.4) formulates sufficient conditions for the stability. For any $k \geq 1$,

$$\begin{aligned}
\mathbf{E}\{Y_{n+1} \mid Y_n = k\} &= \frac{\mathbf{E}\{Y_{n+1} I_{\{Y_n=k\}}\}}{\mathbf{P}\{Y_n = k\}} \\
&= \frac{\mathbf{E}\{\sum_{i=0}^{\infty} Y_{n+1} I_{\{Y_n=k, X_n=i\}}\}}{\mathbf{P}\{Y_n = k\}} \\
&= \frac{\sum_{i=0}^{\infty} \mathbf{E}\{Y_{n+1} \mid Y_n = k, X_n = i\}\mathbf{P}\{X_n = i, Y_n = k\}}{\mathbf{P}\{Y_n = k\}} \\
&= \sum_{i=0}^{\infty} \mathbf{E}\{Y_{n+1} \mid X_n = i, Y_n = k\}\mathbf{P}\{X_n = i \mid Y_n = k\} \\
&= \sum_{i=0}^{\infty} \mathbf{E}\{Y_{n+1} \mid X_n = i\}\mathbf{P}\{X_n = i \mid Y_n = k\} \\
&= \sum_{i=0}^{\infty} L_i \frac{(\lambda k)^i}{i!} e^{-\lambda k} \\
&= L(\lambda k).
\end{aligned}$$

Put

$$D = \limsup_{z \to \infty} \frac{L(z)}{z},$$

then there is $\epsilon > 0$ such that

$$\lambda < \frac{1}{D + \epsilon},$$

and there exists $z_0 > 0$ such that for any $z \geq z_0$,

$$L(z) \leq (D + \epsilon) \cdot z - 1.$$

Then on the one hand for $\lambda k < z_0$,

$$\mathbf{E}\{Y_{n+1} \mid Y_n = k\} = L(\lambda k) \leq \sup_{z < z_0} L(z) < \infty,$$

and on the other hand for $\lambda k \geq z_0$

$$\mathbf{E}\{Y_{n+1} \mid Y_n = k\} = L(\lambda k) \leq (D + \epsilon)\lambda k - 1 \leq k - 1,$$

and so we verified the conditions of Theorem B.4 with $I = z_0/\lambda$, $C = \sup_{z<z_0} L(z)$, $d = 1$. □

REMARK. For TCRA, (6.5) and this theorem implies the stability if

$$\lambda < \frac{1}{2.88545} = 0.3465 \ .$$

For MTCRA, (6.9) and this theorem implies the stability if

$$\lambda < \frac{1}{2.664} = 0.3753 \ .$$

6.5 Gallager-algorithm

Capetanakis called the previous algorithm as fixed tree algorithm. He observed that in case of long CRI, approximately half of the nodes of the tree are intermediate nodes, therefore the collision resolution time can be decreased if the root node is not binary. He improved the fixed tree algorithm by the so called dynamic tree algorithm, where the degree of the root depends on an estimate of the number of new packets. Gallager (1978) keeps the CRI small. For his channel access protocol, we have a second time increment of size Δ. The i-th arrival epoch is $[i\Delta, (i+1)\Delta)$. The i-th CRI is the collision resolution interval of the packets arrived in the i-th arrival epoch, so a new packet that arrived during the i-th arrival epoch is transmitted in the first utilizable slot following the $(i-1)$-th CRI. Let Y_n be the length of the n-th CRI then the waiting time of the resolution of the $(n+1)$-th arrival epoch is denoted by W_{n+1} and can be calculated as follows:

$$W_{n+1} = (W_n - \Delta + Y_{n+1})^+. \tag{6.26}$$

Theorem 6.3. *For the Gallager-algorithm, assume that the arrivals of the packets are according to a Poisson process $\{Z(t) : t \geq 0\}$ with intensity $\lambda > 0$. For TCRA, if*

$$\lambda < \max_z \frac{z}{L(z)},$$

then $\{W_n\}$ is a stable Markov process.

Proof. Under the conditions of the theorem, $\{Y_n\}$ are independent and identically distributed random variables, so $\{W_n\}$ defined by the evolution equation (6.26) is a homogeneous, irreducible, aperiodic Markov process and a sufficient condition of the stability can be derived from Theorem B.6:

$$\mathbf{E}\{Y_1\} < \Delta.$$

Let X_n be the number of packets arrived in the n-th arrival epoch, then X_n is Poisson distributed with parameter $\lambda\Delta$. Thus for TCRA,

$$\begin{aligned}\mathbf{E}\{Y_1\} &= \sum_{i=0}^{\infty}\mathbf{E}\{Y_1 \mid X_0 = i\}\mathbf{P}\{X_0 = i\}\\ &= \sum_{i=0}^{\infty} L_i\mathbf{P}\{X_0 = i\}\\ &= L(\lambda\Delta),\end{aligned}$$

therefore $\{W_n\}$ is stable if

$$L(\lambda\Delta) < \Delta,$$

i.e.,

$$\lambda < \frac{\lambda\Delta}{L(\lambda\Delta)}.$$

Put z^* such that

$$\frac{z^*}{L(z^*)} = \max_z \frac{z}{L(z)},$$

and Δ_λ such that $\lambda\Delta_\lambda = z^*$ then we get that

$$\lambda < \frac{z^*}{L(z^*)}.$$

□

REMARK. If in the Gallager algorithm, we use TCRA then from (6.3) the stability condition is

$$\lambda < 0.4295,$$

and from (6.7) for MTCRA,

$$\lambda < 0.4622.$$

6.6 Part and try algorithm

Tsybakov and Mihailov (1980) introduced the part and try algorithm. Let us divide the frame size Δ into two subframe of length $\frac{\Delta}{2}$. Let the number of arriving packets in a given frame be N from which N_1 arrives in the first subframe and N_2 in the second one ($N = N_1 + N_2$). In the first slot of the CRI every packets are transmitted which arrived in the arrival epoch of length Δ. If the first slot is empty or success, then collision resolution is done and the length of CRI is 1. If there is a collision in the first slot, then only the packets arrived in the first subframe of length $\frac{\Delta}{2}$ is transmitted. If the second slot is empty, then $N_2 \geq 2$, and from the third slot we do collision resolution for N_2 packets recursively with this algorithm and we are using the modification of Massey. This needs all together $\widetilde{L}_{N_2} + 1$ slots. If the second slot is success, then $N_2 \geq 1$, and from the third slot we do collision resolution for N_2 packets recursively with this algorithm. This needs $\widetilde{L}_{N_2} + 2$ slots. Finally, if the second slot is collision, then we do collision resolution for N_1 packets from the second frame which needs $\widetilde{L}_{N_1} + 1$ slots. (In this case we do not resolve the N_2 packets. The next arrival epoch is initiated at $\frac{\Delta}{2}$. So, $\widetilde{L}_N = 1$ if $N \leq 1$, and

$$\widetilde{L}_N = \begin{cases} 1 + \widetilde{L}_{N_2}, & N_1 = 0 \\ 2 + \widetilde{L}_{N_2}, & N_1 = 1 \\ 1 + \widetilde{L}_{N_1}, & N_1 \geq 2 \end{cases}$$

if $N \geq 2$. Thus

$$\begin{aligned} \widetilde{L}_N &= I_{\{N\leq 1\}} + I_{\{N\geq 2\}} \left(1 + I_{\{N_1=0\}} L_{N_2} + I_{\{N_1=1\}}(1 + L_{N_2}) + I_{\{N_1\geq 2\}} L_{N_1}\right) \\ &= 1 + I_{\{N_2\geq 2\}} I_{\{N_1=0\}} L_{N_2} + I_{\{N_2\geq 1\}} I_{\{N_1=1\}}(1 + L_{N_2}) + I_{\{N_1\geq 2\}} L_{N_1} \\ &= 1 + \left(1 - I_{\{N_2\leq 1\}}\right) I_{\{N_1=0\}} L_{N_2} + \left(1 - I_{\{N_2=0\}}\right) I_{\{N_1=1\}}(1 + L_{N_2}) \\ &\quad + \left(1 - I_{\{N_1\leq 1\}}\right) L_{N_1} \\ &= 1 + I_{\{N_1=0\}} L_{N_2} - I_{\{N_2\leq 1\}} I_{\{N_1=0\}} + I_{\{N_1=1\}}(1 + L_{N_2}) - 2I_{\{N_2=0\}} I_{\{N_1=1\}} \\ &\quad + L_{N_1} - I_{\{N_1\leq 1\}} \\ &= 1 + I_{\{N_1\leq 1\}} L_{N_2} - I_{\{N\leq 1\}} - I_{\{N_2=0\}} I_{\{N_1=1\}} + L_{N_1} - I_{\{N_1=0\}} \end{aligned} \tag{6.27}$$

and

$$\begin{aligned} \widetilde{L}(\lambda\Delta) &= \mathbf{E}\{\widetilde{L}_N\} \\ &= 1 + \left(e^{-\frac{\lambda\Delta}{2}} + \frac{\lambda\Delta}{2} e^{-\frac{\lambda\Delta}{2}}\right) \widetilde{L}\left(\tfrac{\lambda\Delta}{2}\right) - e^{-\lambda\Delta} - \lambda\Delta e^{-\lambda\Delta} \\ &\quad - e^{-\frac{\lambda\Delta}{2}} \frac{\lambda\Delta}{2} e^{-\frac{\lambda\Delta}{2}} + L\left(\tfrac{\lambda\Delta}{2}\right) - e^{-\frac{\lambda\Delta}{2}} \\ &= 1 - e^{-\frac{\lambda\Delta}{2}} - \frac{3\lambda\Delta}{2} e^{-\lambda\Delta} - e^{-\lambda\Delta} + \left(1 + e^{-\frac{\lambda\Delta}{2}} + \frac{\lambda\Delta}{2} e^{-\frac{\lambda\Delta}{2}}\right) \widetilde{L}\left(\tfrac{\lambda\Delta}{2}\right). \end{aligned}$$

An alternative way of calculating $\widetilde{L}_N$ is the following:

$$\widetilde{L}_N = 1 + 2^{-N}\widetilde{L}_N + N2^{-N}(1+\widetilde{L}_{N-1}) + \sum_{i=2}^{N}\binom{N}{i}2^{-N}\widetilde{L}_i$$

if $N \geq 2$ (otherwise $\widetilde{L}_N = 1$).

This algorithm does not resolve necessarily all the packets which arrived in the arrival epoch of length Δ. Let us denote by $\widetilde{W}_N$ the time shift (in the unity of Δ) which should be applied to the present arrival epoch to get the next one, if there have been arrived N packets in the present arrival epoch. $\widetilde{W}_N = 1$, if $N \leq 1$ and

$$\begin{aligned}
\widetilde{W}_N &= I_{\{N_1\leq 1\}}\frac{\widetilde{W}_{N_2}+1}{2} + I_{\{N_1\geq 2\}}\frac{\widetilde{W}_{N_1}}{2} \\
&= I_{\{N_1\leq 1\}}\frac{\widetilde{W}_{N_2}+1}{2} + \left(1 - I_{\{N_1\leq 1\}}\right)\frac{\widetilde{W}_{N_1}}{2} \\
&= I_{\{N_1\leq 1\}}\frac{\widetilde{W}_{N_2}}{2} + \frac{\widetilde{W}_{N_1}}{2},
\end{aligned}$$

if $N \geq 2$.

Let us calculate the expected value of $\widetilde{W}_N$.

$$\begin{aligned}
\widetilde{W}(\lambda\Delta) &= \mathbf{E}\{\widetilde{W}_N\} \\
&= \left(e^{-\frac{\lambda\Delta}{2}} + \frac{\lambda\Delta}{2}e^{-\frac{\lambda\Delta}{2}}\right)\frac{1}{2}\widetilde{W}\left(\tfrac{\lambda\Delta}{2}\right) + \frac{1}{2}\widetilde{W}\left(\tfrac{\lambda\Delta}{2}\right) \\
&= \frac{1}{2}\left(1 + e^{-\frac{\lambda\Delta}{2}} + \frac{\lambda\Delta}{2}e^{-\frac{\lambda\Delta}{2}}\right)\widetilde{W}\left(\tfrac{\lambda\Delta}{2}\right)
\end{aligned}$$

Note, that the expected value of the length of the arrival epoch processed is $\Delta\widetilde{W}(\lambda\Delta)$.

An alternative way of calculating $\widetilde{W}_N$ is the following:

$$\widetilde{W}_N = 2^{-N}\frac{\widetilde{W}_N+1}{2} + N2^{-N}\frac{\widetilde{W}_{N-1}+1}{2} + \sum_{i=2}^{N}2^{-N}\binom{N}{i}\frac{\widetilde{W}_i}{2}$$

Similarly to the analysis of the Gallager algorithm, let Y_n be the length of the n-th CRI and let Δ_n be the length of n-th arrival epoch processed, then the waiting time of the resolution of the $(n+1)$-th arrival epoch is denoted by W_{n+1} and can be calculated as follows:

$$W_{n+1} = (W_n - \Delta_n + Y_{n+1})^{+}. \tag{6.28}$$

N	$\widetilde{L}_N$	$\widetilde{W}_N$
0	1	1
1	1	1
2	4	0.833
3	5.833	0.643
4	6.476	0.505
5	6.670	0.416
6	6.836	0.357
7	7.029	0.313
8	7.218	0.278
9	7.389	0.251
10	7.541	0.228

Table 6.4: Values of $\widetilde{L}_N$ and $\widetilde{W}_N$

Theorem 6.4. *For the part-and-try algorithm, assume that the arrivals of the packets are according to a Poisson process $\{Z(t) : t \geq 0\}$ with intensity $\lambda > 0$. For TCRA, if*

$$\lambda < \max_z \frac{z\widetilde{W}(z)}{\widetilde{L}(z)},$$

then $\{W_n\}$ is a stable Markov process.

Proof. Under the conditions of the theorem, $\{Y_n\}$ are independent and identically distributed random variables, so $\{W_n\}$ defined by the evolution equation (6.28) is a homogeneous, irreducible, aperiodic Markov process and a sufficient condition of the stability can be derived from Theorem B.6:

$$\mathbf{E}\{Y_1\} < \mathbf{E}\{\Delta_n\}.$$

Let X_n be the number of packets arrived in the n-th arrival epoch, then X_n is Poisson distributed with parameter $\lambda\Delta$. Thus for TCRA,

$$\begin{aligned}\mathbf{E}\{Y_1\} &= \sum_{i=0}^{\infty} \mathbf{E}\{Y_1 \mid X_0 = i\}\mathbf{P}\{X_0 = i\} \\ &= \sum_{i=0}^{\infty} \widetilde{L}_i \mathbf{P}\{X_0 = i\} \\ &= \widetilde{L}(\lambda\Delta),\end{aligned}$$

moreover

$$\mathbf{E}\{\Delta_n\} = \Delta\widetilde{W}(\lambda\Delta),$$

therefore $\{W_n\}$ is stable if

$$\widetilde{L}(\lambda\Delta) < \Delta\widetilde{W}(\lambda\Delta),$$

i.e.,

$$\lambda < \frac{\lambda\Delta\widetilde{W}(\lambda\Delta)}{\widetilde{L}(\lambda\Delta)}.$$

Put z^* such that

$$\frac{z^*\widetilde{W}(z^*)}{\widetilde{L}(z^*)} = \max_z \frac{z\widetilde{W}(z)}{\widetilde{L}(z)},$$

and Δ_λ such that $\lambda\Delta_\lambda = z^*$ then we get that

$$\lambda < \frac{z^*\widetilde{W}(z^*)}{\widetilde{L}(z^*)}$$

which has its maximum at $z^* = \lambda\Delta = 1.266$ (where $\Delta = 2.6$) resulting in the bound

$$\lambda < 0.48711.$$

□

If the coin is biased with probability p, then from (6.27) we get

$$\begin{aligned}
\widetilde{L}(\lambda\Delta) &= 1 + e^{-p\lambda\Delta}(1 + p\lambda\Delta)\widetilde{L}\left((1-p)\lambda\Delta\right) - e^{-\lambda\Delta} - \lambda\Delta e^{-\lambda\Delta} \\
&\quad - e^{-(1-p)\lambda\Delta}\lambda\Delta e^{-p\lambda\Delta} + \widetilde{L}(p\lambda\Delta) - e^{-p\lambda\Delta} \\
&= 1 - e^{-\lambda\Delta}\left(1 + (1+p)\lambda\Delta\right) - e^{-p\lambda\Delta} \\
&\quad + e^{-p\lambda\Delta}(1 + p\lambda\Delta)\widetilde{L}\left((1-p)\lambda\Delta\right) + \widetilde{L}(p\lambda\Delta)
\end{aligned}$$

For the arrival epoch shift

$$\begin{aligned}
\widetilde{W}_N &= I_{\{N_1 \le 1\}}\Big((1-p)\widetilde{W}_{N_2} + p\Big) + I_{\{N_1 \ge 2\}} p\widetilde{W}_{N_1} \\
&= I_{\{N_1 \le 1\}}(1-p)\widetilde{W}_{N_2} + p\widetilde{W}_{N_1},
\end{aligned}$$

if $N \ge 2$, which gives for the expected value

$$\widetilde{W}(\lambda\Delta) = e^{-p\lambda\Delta}(1 + p\lambda\Delta)(1-p)\widetilde{W}\left((1-p)\lambda\Delta\right) + p\widetilde{W}(p\lambda\Delta).$$

In the biased case intensity has its maximum at $z^* = \lambda\Delta = 1.2725$ ($\Delta = 2.61$) when $p = 0.4756$ and value

$$\lambda = 0.48757.$$

Chapter 7

Multiple Access Adder Channel

7.1 Channel Model

The binary adder channel (Figure 7.1) is a special case of the multiple-access channel. Here the channel input alphabet is binary ($\mathcal{B} = \{0, 1\}$), while the output alphabet is the set of nonnegative integer numbers ($\mathbb{N}$). The channel is deterministic: the output is the sum of the inputs, where the summation is the usual summation over $\mathbb{N}$ (and it is *not* the mod 2 summation). The channel transition probabilities can be written as

$$\mathbf{P}\left(Y = y | X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t\right) = \begin{cases} 1 & \text{if } y = \sum_{i=1}^{t} x_i; \\ 0 & \text{otherwise.} \end{cases}$$

Using the vectorial form we can say, that the received vector is (almost sure) the sum of the sent codeword vectors:

$$\mathbf{Y} = \sum_{i=1}^{t} \mathbf{x}_{M_i}^{(i)},$$

where $\mathbf{x}_j^{(i)}$ is the codeword for the j^{th} message of the i^{th} user, and the random variable M_i denotes the message of the i^{th} user.

The rate region of this channel has been determined by Chang and Weldon (1979). For $t = 2$ it is shown in Figure 7.2.

For a deterministic channel, like the adder channel, it is interesting to define the class of **uniquely decipherable** (u.d.) codes. Code $\mathcal{C}$ is a u.d. code for the adder channel, if the messages of the users can be recovered without error from the output of the channel ($P_e(\mathcal{C}) = 0$). I.e. if the users send different messages, then the received sum vector must be different. To

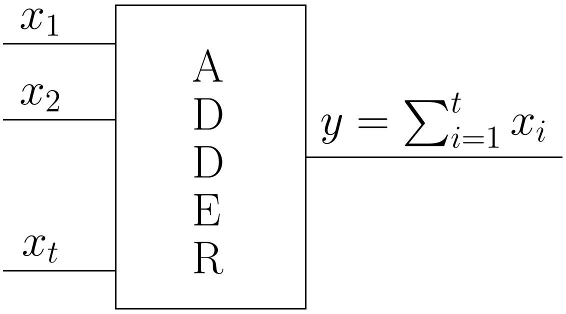

Figure 7.1: The binary adder channel ($x_1, x_2, \ldots, x_t \in \mathcal{B}, y \in \mathbb{N}$)

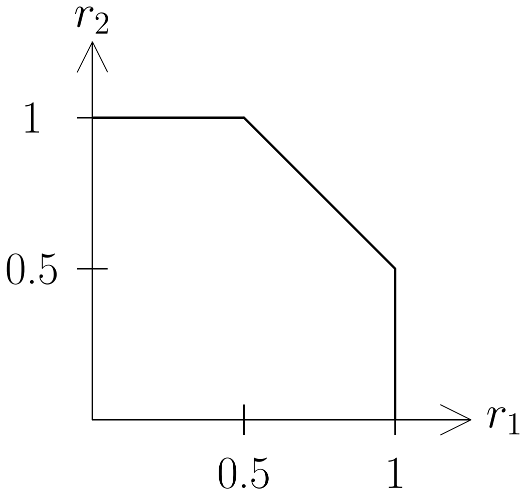

Figure 7.2: The rate region of the binary adder channel for $t = 2$ users

formulate this, let us denote the actual message of the i^{th} user with m_i. We call message-constellation the vector formed by the messages of the users:

$$\mathbf{m} = (m_1, m_2, \ldots, m_t) \in \bigotimes_{i=1}^{t} \{1, 2, \ldots, |C_i|\}.$$

For the adder channel, given the message-constellation $\mathbf{m}$, the channel output vector is deterministic, and is denoted by $\mathbf{S}(\mathbf{m})$. In the case of the adder channel

$$\mathbf{S}(\mathbf{m}) = \sum_{i=1}^{t} \mathbf{x}_{m_i}^{(i)}.$$

Definition 7.1. *A multiple-access code is a uniquely decipherable (u.d.) code for the adder channel, if the received vector is unique considering all message-constellations:*

$$\mathbf{S}(\mathbf{m}_1) = \mathbf{S}(\mathbf{m}_2) \iff \mathbf{m}_1 = \mathbf{m}_2 \qquad \forall \mathbf{m}_1, \mathbf{m}_2 \in \bigotimes_{i=1}^{t} \{1, 2, \ldots, |C_i|\}.$$

Khachatrian (2000) has written an excellent survey on u.d. code constructions for the adder channel, with various rates.

In this article, we will deal only with the class of u.d. codes with symmetric rates. If we consider the sum rate

$$r_{\text{sum}} = \sum_{i=1}^{t} r_i$$

of arbitrary u.d. codes for the adder channel, then from the results of Chang and Weldon (1979) it follows, that its maximum for large t is

$$r_{\text{sum}} \sim \frac{1}{2} \log t.$$

They have also given a u.d. code construction with two codewords per user with asymptotically the same rate. This means, that to asymptotically maximize the sum rate, it is enough to consider codes with two codewords per user:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_t\},$$
$$C_i = \left\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\right\} \qquad \forall i \in [t].$$

In this case, the rates for all users are the same, namely

$$r_i = \frac{1}{n} \qquad \forall i \in [t],$$

and because of this, we can even use the code length instead of the rate vector to compare codes.

As we have just mentioned, Chang and Weldon (1979) and Lindström (1964) have given u.d. code constructions for the binary adder channel. Chang and Weldon's construction is, in fact, a statistical design that was given by Cantor and Mills (1966). Both constructions will be shown later in Sections 7.9 and 7.10.

We will also deal with the **signature coding** problem. Consider an alarming or signaling system: there are t stations, and some of them want to send an alarm signal: they send their codeword to the channel. The others, who do not want to signal an alarm, turn off their transmitter to conserve power. We can consider these stations as they are sending a zero vector to the channel.

The task of the receiver is to detect the alarming stations. This scenario is much like the general two codeword coding problem with one additional constraint: one of the codewords in all component codes should be the zero vector:

$$\mathcal{C} = \Big\{ \{\mathbf{0}, \mathbf{x}^{(1)}\}, \{\mathbf{0}, \mathbf{x}^{(2)}\}, \dots, \{\mathbf{0}, \mathbf{x}^{(t)}\} \Big\}.$$

In the followings, for the sake of simplicity, when dealing with signature codes we will not mention the zero codeword, which is included in all component codes. We will consider, that there is only one codeword per user. Thus we can write the last equation quite simply:

$$\mathcal{C} = \left\{ \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(t)} \right\}$$

Corresponding to the semantics of the signature coding, that there are some alarming and some quiet users, we will use the set U of alarming users instead of the message constellation vector $\mathbf{m}$ we used before. For the binary adder channel the channel output vector given the set $U \subseteq [t]$ of active users is

$$\mathbf{S}(U) = \sum_{i \in U} \mathbf{x}^{(i)}.$$

We can define the class of u.d. signature codes in the same way as before: the channel output must be different for all different set of alarming users.

Definition 7.2. *A signature code for the adder channel is u.d., if the received vector is unique considering all possible sets of active users:*

$$\mathbf{S}(U) = \mathbf{S}(V) \iff U = V \qquad \forall U, V \subseteq [t].$$

7.2 User Models

Modeling the behavior of the users is as important as to model the channel itself. There are two fundamental user models, the permanent activity model and the partial activity model.

The **permanent activity** model is a rather simple one. Here the users considered to be active all the time, i.e. they always have some information to send. In the previous sections we considered this case.

In many real-life examples (think about a mobile phone network, for example) the users are not always active, they are just **partially active**, i.e. sometimes they have information to send, but most of the time they do not have. There are many possibilities to model a user population with this behavior. One interesting scenario is the m-out-of-t model: in this model we assume, that at most m users out of the t total ones are active at any instant, but this active set can vary arbitrarily with time as long as its size does not exceed m.

To make a good formalism for this partial activity case, we consider that the zero vector is added to each component code as a new codeword. The active users simply send the appropriate codeword belonging to their messages, while the inactive ones send the zero vector. This model corresponds to real life, where inactivity means simply turning off the transmitter.

This motivates the use of signature codes, introduced in the previous section. It signature codes, for the active users there are only one usable codeword in their component codes. Thus we cannot transfer messages directly, but we can solve signaling and alarming tasks. For real information transfer, we will show a simple scenario a bit later.

The point in using m-out-of-t model is that, if m is significantly smaller than t, which is usually a realistic case, then an m-out-of-t code can be significantly shorter than a conventional t user multiple-access code.

For real information transmission with signature codes, one can use the following simple scenario. Consider a population of t users with m-out-of-t partial activity model. (Note, that the following construction works also for the permanent activity model.) We will create a u.d. multiple-access code with three codewords per user, one codeword is the all zero one, for signaling inactivity, while the other two are for the messages of the active users. Let us take a u.d. signature code $\mathcal{C}^*$ for $2t$ virtual users out of which at most m are active simultaneously (a code for the $2t$-out-of-m model):

$$\mathcal{C}^* = \Big\{ \{\mathbf{0}, \mathbf{x}^{(1)}\}, \{\mathbf{0}, \mathbf{x}^{(2)}\}, \ldots, \{\mathbf{0}, \mathbf{x}^{(2t)}\} \Big\}.$$

Create the component codes of the new multiple-access code by distributing two codewords from $\mathcal{C}^*$ for each of the t real users, and additionally, put the all zero codeword into the component codes:

$$C_i = \{\mathbf{0}, \mathbf{x}^{(2i-1)}, \mathbf{x}^{(2i)}\} \qquad \forall i \in [t].$$

This way we have got a multiple-access code with two message codewords and one inactivity codeword for t users:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_t\}.$$

Each user can transmit messages with the nonzero codewords, or signal inactivity with the zero one. If the number of simultaneously active users does not exceed m, then the code is u.d. Moreover, the partial activity of the users is exploited in the code, so the codeword length can be far below the length of a code for the permanent activity case.

7.3 Permanent Activity Case

In this section we survey some well known results regarding the u.d. coding problem of the multiple-access adder channel for the permanent user activity case. We will present some bounds on the minimal length of u.d. codes, as well as some code constructions.

7.4 Equivalent Representations

There are many equivalent representations of the u.d. signature coding problem for the adder channel. In this section we show some of them.

Problem 7.1. ***(Coin Weighing Problem)*** *Given t coins, some of them are counterfeit, some are genuine, and we have to distinguish between them. The weight of the counterfeit coins are e.g. 9 grams, while the original ones weigh 10 grams. We have a scale, that weighs exactly. Give sets of coins, for which by measuring the weight of these sets, we can find the counterfeit coins. What is the minimal number of weighings required?*

We have to stress, that we consider here the problem, when the set of coins selected for the future weighings does not depend on the results of the previous weighings, so the sets to weigh is given before we start measuring at all. In terms of search theory this is called non-adaptive or parallel search. (There is another problem, when the next set to weigh can be selected according to the results of the previous weighings, but we do not deal with this so called adaptive search problem here.)

To show how this problem is related to the coding problem, first make some simplifications: let the weight of a counterfeit coin be one, and the weight of a genuine one be zero. Certainly this is still the same problem.

Now let us consider that each coin represents a user. Construct a codeword for each user: let the length of the codeword be equal to the number of weighings. Put 1 to the i^{th} position if the coin associated with this user participates in the i^{th} weighing. If the coin is not participating in that weighing, then put 0 there. If we give the zero vector as a common second codeword for all users, then we get a signature code.

Consider a given set of counterfeit coins, and consider that the users associated with the counterfeit coins send their non-zero codeword. We can consider, that the other users send the zero vector. The result of the weighings will be equal to the sum vector at the channel output. If we can determine the set of false coins from the output of the weighings, then we can determine the messages of the users from the sum vector, and vica versa. So the problems of planning the weighings and finding u.d. signature codes are equivalent.

Problem 7.2. ***(Combinatorial Detection Problem)*** *Given a finite set T construct a set $\mathcal{M}$ of subsets of it ($\mathcal{M} \subseteq \mathcal{P}(T)$, where $\mathcal{P}(T)$ is the powerset of T), in such a way that given an arbitrary $U \subseteq T$ we can determine U if we know the sizes of the intersections $|U \cap M|$ for all $M \in \mathcal{M}$. What is the minimal size of $\mathcal{M}$?*

To find the equivalence of this problem with the previous one is really easy. Here the sets in $\mathcal{M}$ denote the sets of coins weighed together. So it follows, that this combinatorial detection problem is equivalent to the u.d. coding problem. But we will give a more formal proof of it using the matrix representation.

The **matrix representation** is simply another representation of the last combinatorial detection problem. If $T = [t]$, then we can represent one subset $U \subseteq T$ with a binary column vector $\mathbf{u} = (u_1, u_2, \ldots, u_t)$:

$$[t] \supseteq U \mapsto \mathbf{u} \in \mathcal{B}^t \text{ where } u_i = \begin{cases} 1 & \text{if } i \in U; \\ 0 & \text{if } i \notin U \end{cases} \qquad \forall i \in [t]. \tag{7.1}$$

The combinatorial detection problem can be formulated as to find column vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n \in \mathbf{B}^t$ in such a way, that if we know the values ${\mathbf{y}_i}^\top \mathbf{u}$ for all $i \in [n]$, then we can determine the vector $\mathbf{u}$. To complete the matrix representation, create a matrix from the row vectors ${\mathbf{y}_i}^\top$ by simply writing them below each other:

$$\mathbf{C} = \begin{pmatrix} {\mathbf{y}_1}^\top \\ {\mathbf{y}_2}^\top \\ \vdots \\ {\mathbf{y}_n}^\top \end{pmatrix}.$$

Now the problem is to find a matrix $\mathbf{C}$ for which

$$\mathbf{Cu} = \mathbf{Cv} \iff \mathbf{u} = \mathbf{v} \qquad \forall \mathbf{u} \in \mathcal{B}^t, \forall \mathbf{v} \in \mathcal{B}^t, \tag{7.2}$$

or equivalently, introducing $\mathbf{w} = \mathbf{u} - \mathbf{v}$, the problem is to find a $\mathbf{C}$, for which

$$\mathbf{Cw} = \mathbf{0} \iff \mathbf{w} = \mathbf{0} \qquad \forall \mathbf{w} \in \{-1, 0, 1\}^t. \tag{7.3}$$

Now we show the mapping between this matrix representation of the combinatorial detection problem and the u.d. coding problem of the multiple-access adder channel.

Let us write the above matrix $\mathbf{C}$ as column vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)} \in \mathcal{B}^n$ written next to each other:

$$\mathbf{C} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(t)}\right).$$

Consider a signature code, where each user has a column vector $\mathbf{x}^{(i)}$ from above as codeword:

$$\mathcal{C} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}\}.$$

It is easy to see, that this code is a u.d. signature code for the adder channel for t users. Let $[t]$ denote the population of the t users. Let us denote the set of active users at a given instant with $U \subseteq T$. With these notations, the u.d. property in the case of the adder channel was the following (see Definition 7.2):

$$\mathbf{S}(U) = \mathbf{S}(V) \iff U = V \qquad \forall U, V \subseteq [t].$$

Let us represent sets U and V with vectors $\mathbf{u}$ and $\mathbf{v}$ in the way given by our mapping (7.1). Now the uniquely decipherable property becomes

$$\mathbf{Cu} = \mathbf{Cv} \iff \mathbf{u} = \mathbf{v} \qquad \forall \mathbf{u} \in \mathcal{B}^t, \forall \mathbf{v} \in \mathcal{B}^t.$$

which is exactly the same formula as (7.2) which was given as the condition on $\mathbf{C}$.

7.5 Trivial Bounds on $N(t)$

First we present here a very simple statement about the minimal code length for the binary adder channel with two codewords per user.

Definition 7.3. *The **minimal code length** $N(t)$ is the length of the shortest possible u.d. code for a given number of users:*

$$N(t) = \min\{n \colon \exists \mathcal{C} \text{ u.d. code with length } n \text{ for } t \text{ users}\}.$$

Definition 7.4. *The **minimal signature code length** $N_s(t)$ is the length of the shortest possible u.d. signature code for a given number of users:*

$$N_s(t) = \min\{n \colon \exists \mathcal{C} \text{ u.d. signature code with length } n \text{ for } t \text{ users}\}.$$

Theorem 7.1.

$$\frac{t}{\log(t+1)} \le N(t) \le N_s(t) \le t.$$

Proof. For the lower bound, we simply use enumeration: there are 2^t possible message-constellations for t users ($\mathbf{m} \in \{1,2\}^t$). On the other side, the received vector has components in $\{0,1,\ldots,t\}$, and in case of the shortest possible code it is of length $N(t)$. For the u.d. property, the number of possible received vectors cannot be smaller than the number of possible user subsets. Thus we have

$$(t+1)^{N(t)} \ge 2^t,$$

from which

$$N(t) \ge \frac{t}{\log(t+1)}.$$

(Here and from now, log represents the logarithm of base 2.)

The statement $N(t) \le N_s(t)$ follows simply from the definition: all u.d. signature code is also a u.d. code, so the minimal u.d. code length cannot be greater than the minimal u.d. signature code length.

For the upper bound, we show a trivial u.d. signature code construction for t users with codeword length $n = t$: simply let the codeword of the i^{th} user be $\mathbf{e}^{(i)}$ which is the i^{th} unit vector. Using the matrix representation, we can say, that for t users let $\mathbf{C} = \mathbf{I}_t$ (which is the identity matrix of size $t \times t$). Since this matrix is invertible, this signature code certainly has the u.d. property (7.3). So $N_s(t) \le t$. □

7.6 Upper Bound of Erdős and Rényi

Erdős and Rényi (1963) has presented a nontrivial asymptotic upper bound on the minimal signature code length for the adder channel:

Theorem 7.2. *(Erdős–Rényi (1963))*

$$\limsup_{t\to\infty} \frac{N_s(t)\log t}{t} \le \log 9.$$

Proof. The proof is based on random coding. We select a random signature code $\mathcal{C}$ of length $n+1$ in such a way, that the first n components of the codewords are i.i.d. uniformly distributed binary random variables, while the $(n+1)^{\text{th}}$ component is fixed to 1:

$$\mathbf{P}\left(x_i^{(j)} = 0\right) = \mathbf{P}\left(x_i^{(j)} = 1\right) = \frac{1}{2} \qquad \forall j \in [t], \forall i \in [n] \text{ (i.i.d.)},$$

$$x_{n+1}^{(j)} = 1 \qquad \forall j \in [t].$$

We will work with the probability of the event

$$\text{code } \mathcal{C} \text{ is not a u.d. signature code,} \tag{*}$$

and we will give an upper bound for it, which will tend to 0 as $t \to \infty$.

We have introduced $\mathbf{S}(U)$ as the channel output when the set of active users is U. Simply we have that if $\mathcal{C}$ is not a u.d. signature code, then there exist two different sets of active users, say U and V, for which $\mathbf{S}(U) = \mathbf{S}(V)$:

$$\mathbf{P}\big(\text{event } (*)\big) = \mathbf{P}\left(\bigcup_{U \neq V \subseteq [t]} \big\{\mathbf{S}(U) = \mathbf{S}(V)\big\} \right).$$

If there are two subsets U and V for which $\mathbf{S}(U) = \mathbf{S}(V)$, then there are also two disjoint subsets with the same property: $U \setminus V$ and $V \setminus U$ will suite. We also know, that $|U| = |V|$, since the $(n+1)^{\text{th}}$ component of the received vector is simply the number of active users, so if the received vectors are equal, then the sets of the active users must be of the same size:

$$\mathbf{P}\big(\text{event } (*)\big) = \mathbf{P}\left(\bigcup_{\substack{U \neq V \subseteq [t]: \\ |U|=|V|, U \cap V = \emptyset}} \big\{\mathbf{S}(U) = \mathbf{S}(V)\big\} \right).$$

We can use the so called union bound as an upper bound:

$$\begin{aligned} \mathbf{P}\big(\text{event } (*)\big) &\leq \sum_{\substack{U \neq V \subseteq [t]: \\ |U|=|V|, U \cap V = \emptyset}} \mathbf{P}\big(\mathbf{S}(U) = \mathbf{S}(V)\big) \\ &= \sum_{k=1}^{\lfloor \frac{t}{2} \rfloor} \sum_{\substack{U,V \subseteq [t]: \\ |U|=|V|=k, U \cap V = \emptyset}} \mathbf{P}\big(\mathbf{S}(U) = \mathbf{S}(V)\big), \end{aligned} \tag{7.4}$$

since the common size of the active subsets is at most $\lfloor \frac{t}{2} \rfloor$.

The components of the codewords in $\mathcal{C}$ has a simple distribution, so it is easy to calculate $\mathbf{P}\big(\mathbf{S}(U) = \mathbf{S}(V)\big)$ for some fixed disjoint U and V of the same size k:

$$\begin{aligned} \mathbf{P}\big(\mathbf{S}(U) = \mathbf{S}(V)\big) &= \mathbf{P}\left(\bigcap_{i=1}^{n} \bigcup_{\ell=0}^{k} \{S_i(U) = \ell \text{ and } S_i(V) = \ell\} \right) \\ &= \left(\sum_{\ell=0}^{k} \binom{k}{\ell} 2^{-k} \binom{k}{\ell} 2^{-k} \right)^{n}, \end{aligned}$$

since for disjoint U and V, the components of $S(U)$ and $S(V)$ are independent, and has binomial distribution.

Let us introduce

$$Q(k) = \sum_{\ell=0}^{k} \binom{k}{\ell} 2^{-k} \binom{k}{\ell} 2^{-k}. \tag{7.5}$$

We have

$$Q(k) = \sum_{\ell=0}^{k} \binom{k}{\ell} \binom{k}{k-\ell} 2^{-2k} = \binom{2k}{k} 2^{-2k}.$$

For $1 < k$

$$\binom{2k}{k} 2^{-2k} \leq \frac{1}{\sqrt{\pi k}}.$$

(C.f. Gallager (1968) Problem 5.8 pp. 530.) Thus

$$Q(k) \leq \frac{1}{\sqrt{\pi k}}. \tag{7.6}$$

Substituting this result into (7.4), we get

$$\begin{aligned} \mathbf{P}\big(\text{event } (*)\big) &\leq \sum_{k=1}^{\lfloor \frac{t}{2} \rfloor} \sum_{\substack{U,V \subseteq [t]: \\ |U|=k, |V|=k, \\ U \cap V = \emptyset}} \left(\frac{1}{\sqrt{\pi k}}\right)^n \\ &= \sum_{k=1}^{\lfloor \frac{t}{2} \rfloor} \binom{t}{k} \binom{t-k}{k} \left(\frac{1}{\sqrt{\pi k}}\right)^n. \end{aligned} \tag{7.7}$$

Split the summation into two parts:

$$k = 1, 2, \ldots, \left\lfloor \frac{t}{2\log^2 t} \right\rfloor,$$

and

$$k = \left\lfloor \frac{t}{2\log^2 t} \right\rfloor + 1, \left\lfloor \frac{t}{2\log^2 t} \right\rfloor + 2, \ldots, \left\lfloor \frac{t}{2} \right\rfloor.$$

For the first part, we use $\frac{1}{\sqrt{\pi k}} \leq \frac{1}{\sqrt{\pi}}$ and $\binom{t}{k}\binom{t-k}{k} \leq t^{2k}$:

$$\begin{aligned}\sum_{k=1}^{\left\lfloor \frac{t}{2\log^2 t}\right\rfloor} \binom{t}{k}\binom{t-k}{k}\left(\frac{1}{\sqrt{\pi k}}\right)^n &\leq \sum_{k=1}^{\left\lfloor \frac{t}{2\log^2 t}\right\rfloor} t^{2k}\left(\frac{1}{\sqrt{\pi}}\right)^n \\ &\leq \frac{(t^2)^{\frac{t}{2\log^2 t}}-1}{t^2-1}\left(\frac{1}{\sqrt{\pi}}\right)^n \\ &\leq t^{\frac{t}{\log^2 t}}\left(\frac{1}{\sqrt{\pi}}\right)^n \\ &\leq 2^{\frac{t}{\log t}-\frac{n}{2}\log\pi},\end{aligned}$$

which tends to zero as $t \to \infty$ if

$$\lim_{t\to\infty}\left(\frac{t}{\log t} - \frac{n}{2}\log\pi\right) = -\infty. \tag{7.8}$$

For the second part of the summation in (7.7), we use $\sum_{k=1}^{\left\lfloor \frac{t}{2}\right\rfloor}\binom{t}{k}\binom{t-k}{k} \leq 3^t$, which holds since selecting two subsets of $[t]$ is equivalent of partitioning it into three parts. So

$$\begin{aligned}\sum_{k=\left\lfloor \frac{t}{2\log^2 t}\right\rfloor+1}^{\left\lfloor \frac{t}{2}\right\rfloor} \binom{t}{k}\binom{t-k}{k}\left(\frac{1}{\sqrt{\pi k}}\right)^n &\leq \sum_{k=1}^{\left\lfloor \frac{t}{2}\right\rfloor}\binom{t}{k}\binom{t-k}{k}\left(\frac{1}{\sqrt{\frac{\pi t}{2\log^2 t}}}\right)^n \\ &\leq 3^t\left(\frac{1}{\sqrt{\frac{\pi t}{2\log^2 t}}}\right)^n \\ &= 2^{t\log 3-\frac{n}{2}\log\frac{\pi t}{2\log^2 t}},\end{aligned}$$

which tends to zero as $t \to \infty$ if

$$\lim_{t\to\infty}\left(t\log 3 - \frac{n}{2}\log\frac{\pi t}{2\log^2 t}\right) = -\infty. \tag{7.9}$$

Let us set

$$n = \left\lceil \frac{ct}{\log t}\right\rceil.$$

In the first condition (7.8) this yields

$$\begin{aligned}\lim_{t\to\infty}\left(\frac{t}{\log t}-\frac{n}{2}\log\pi\right) &\leq \lim_{t\to\infty}\left(\frac{t}{\log t}-\frac{ct\log\pi}{2\log t}\right) \\ &= \left(1-\frac{c\log\pi}{2}\right)\lim_{t\to\infty}\frac{t}{\log t},\end{aligned}$$

which is $-\infty$ if $c > \frac{2}{\log \pi} \approx 1.211$. In the second condition (7.9),

$$\begin{aligned}\lim_{t\to\infty}\left(t\log 3 - \frac{n}{2}\log\frac{\pi t}{2\log^2 t}\right) &\leq \lim_{t\to\infty}\left(t\log 3 - \frac{ct}{2\log t}\log\frac{\pi t}{2\log^2 t}\right)\\ &= \lim_{t\to\infty}\left(t\left(\log 3 - \frac{c}{2}\frac{\log \pi t - \log\left(2\log^2 t\right)}{\log t}\right)\right)\\ &= \left(\log 3 - \frac{c}{2}\right)\lim_{t\to\infty} t,\end{aligned}$$

which is $-\infty$ if $c > \log 9 \approx 3.170$.

So we have shown, that for all $\varepsilon > 0$ and $n = \left\lceil \frac{(\varepsilon+\log 9)t}{\log t}\right\rceil$,

$$\lim_{t\to\infty} \mathbf{P}\big(\text{event } (*)\big) = 0.$$

This means, that for t large enough, the random code of length $n+1$ we select is a u.d. signature code with positive probability. So for t large enough, there exists a u.d. signature code of this length:

$$N_s(t) \leq \left\lceil \frac{(\varepsilon+\log 9)t}{\log t}\right\rceil + 1 \leq \frac{(\varepsilon+\log 9)t}{\log t} + 2,$$

or equivalently

$$\limsup_{t\to\infty} \frac{N_s(t)\log t}{t} \leq \log 9.$$

□

7.7 Pippenger Theorem

Pippenger (1981) extended the definition of u.d. signature codes. A code $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ is called a u.d. signature code **with multiplicity**, if the sum

$$\mathbf{S}(\mathbf{u}) = \sum_{i=1}^{t} u_i \mathbf{x}_i$$

uniquely determines ($\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v}) \iff \mathbf{u} = \mathbf{v}$) the components of vector $\mathbf{u} = (u_1, u_2, \ldots, u_t) \in \{0, 1, \ldots, t\}^t$ satisfying

$$\sum_{i=1}^{t} u_i \leq t.$$

Theorem 7.3. *(Pippenger (1981)) For the length $N_{sm}(t)$ of the shortest possible u.d. signature code with multiplicity for t users,*

$$\limsup_{t\to\infty} \frac{N_{sm}(t)\log t}{t} \le 8.$$

Proof. Pippenger proved the existence of such codes by random coding. We select a random code $\mathcal{C}$ of length n for t users, and we will show, that the probability of the event, that

$$\text{code } \mathcal{C} \text{ is not a u.d. signature code with mulitplicity} \qquad (*)$$

is less than one for t large enough.

We select a random code $\mathcal{C}$ of length $n+1$ in such a way, that the first n components of the codewords are i.i.d. uniformly distributed binary random variables, while the $(n+1)^{\text{th}}$ component is fixed to 1:

$$\begin{aligned} &\mathbf{P}\left(x_i^{(j)}=0\right)=\mathbf{P}\left(x_i^{(j)}=1\right)=\frac{1}{2} && \forall j\in[t], \forall i\in[n] \text{ (i.i.d.)},\\ &x_{n+1}^{(j)}=1 && \forall j\in[t]. \end{aligned}$$

We need a code with property

$$\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v}) \iff \mathbf{u}=\mathbf{v} \qquad \forall \mathbf{u},\mathbf{v}\in\{0,1,2,\ldots,t\}^t\colon \sum_{i=1}^t u_i\le t, \sum_{i=1}^t v_i\le t,$$

so

$$\begin{aligned} \mathbf{P}\big(\text{event } (*)\big) &= \mathbf{P}\left(\bigcup_{\substack{\mathbf{u},\mathbf{v}\in\{0,1,2,\ldots,t\}^t\colon\\ \sum_{i=1}^t u_i\le t, \sum_{i=1}^t v_i\le t}} \big\{\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v})\big\}\right)\\ &= \mathbf{P}\left(\bigcup_{\substack{\mathbf{u},\mathbf{v}\in\{0,1,2,\ldots,t\}^t\colon\\ \mathbf{u}^\top\mathbf{v}=0, \sum_{i=1}^t u_i=\sum_{i=1}^t v_i\le t}} \big\{\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v})\big\}\right), \end{aligned}$$

We also know, that $\sum_{i=1}^t u_i=\sum_{i=1}^t v_i$, since the $(n+1)^{\text{th}}$ component of the vector $\mathbf{S}(\mathbf{u})$ is simply the number of active users, so if the received vectors are equal, then the sum of the vectors must be the same. We also used that if there is a vector pair $(\mathbf{u},\mathbf{v})$ which satisfy $\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v})$, then there is also an orthogonal vector pair satisfying it. E.g. $(\mathbf{u}', \mathbf{v}')$, where $\mathbf{u}' = \bigl(u_1', u_2', \ldots, u_t'\bigr)$, $\mathbf{v}' = \bigl(v_1', v_2', \ldots, v_t'\bigr)$, where

$$u_i' = u_i - \min\{u_i, v_i\}$$

and

$$v_i' = v_i - \min\{u_i, v_i\}.$$

Now applying the union bound one gets

$$\mathbf{P}\bigl(\text{event } (*)\bigr) \leq \sum_{\substack{\mathbf{u},\mathbf{v} \in \{0,1,2,\ldots,t\}^t : \\ \mathbf{u}^\top \mathbf{v} = 0, \sum_{i=1}^t u_i = \sum_{i=1}^t v_i \leq t}} \mathbf{P}\left(\bigl\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\bigr\}\right). \tag{7.10}$$

For a given $\mathbf{u}$ and $\mathbf{v}$ we can bound $\mathbf{P}\left(\bigl\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\bigr\}\right)$. For this we will use the following sets:

$$U = \bigl\{i \in \{1.2.\ldots,t\} \colon u_i \neq 0\bigr\}$$

and

$$V = \bigl\{i \in \{1.2.\ldots,t\} \colon v_i \neq 0\bigr\}.$$

Since $\mathbf{u}^\top \mathbf{v} = 0$, we know that $U \cap V = \emptyset$.

For each component k, we assign a subset of $U \cup V$ to each code:

$$f_k : \mathcal{C} \mapsto U \cap A_k(\mathcal{C}) \cup V \setminus A_k(\mathcal{C}),$$

where $A_k(\mathcal{C}) = \Bigl\{i \in \{1,2,\ldots,t\} \colon x_k^{(i)} \neq 0\Bigr\}$. We now state a lemma:

Lemma 7.1. *For codes $\mathcal{C}$ and $\mathcal{D}$ for which*

$$\mathbf{S}_\mathcal{C}(\mathbf{u}) = \mathbf{S}_\mathcal{C}(\mathbf{v}),$$
$$\mathbf{S}_\mathcal{D}(\mathbf{u}) = \mathbf{S}_\mathcal{D}(\mathbf{v})$$

and

$$A_k(\mathcal{C}) \neq A_k(\mathcal{D}),$$

$f_k(\mathcal{C})$ and $f_k(\mathcal{D})$ are incomparable in the sense that neither is contained within the other.

Note, that the third condition of the lemma can be interpreted as codes must differ in the k^{th} component for at least one user in $U \cup V$.

Proof. Imagine the contrary: if $f_k(\mathcal{C}) \subseteq f_k(\mathcal{D})$, or equivalently

$$U \cap A_k(\mathcal{C}) \cup V \setminus A_k(\mathcal{C}) \subseteq U \cap A_k(\mathcal{D}) \cup V \setminus A_k(\mathcal{D}),$$

then since $U \cap V = \emptyset$,

$$A_k(\mathcal{C}) \cap U \subseteq A_k(\mathcal{D}) \cap U, \tag{7.11}$$

and

$$A_k(\mathcal{C}) \cap V \supseteq A_k(\mathcal{D}) \cap V. \tag{7.12}$$

Moreover, if $A_k(\mathcal{C}) \neq A_k(\mathcal{D})$, then at least one of the coverings is of proper subset. Remember that

$$S_{\mathcal{C}k}(\mathbf{u}) = \sum_{i=1}^{t} u_i x_k^{(i)} = \sum_{i \in A_k(\mathcal{C})} u_i = \sum_{i \in A_k(\mathcal{C}) \cap U} u_i, \tag{7.13}$$

and similary

$$S_{\mathcal{C}k}(\mathbf{v}) = \sum_{i \in A_k(\mathcal{C}) \cap V} v_i, \tag{7.14}$$

$$S_{\mathcal{D}k}(\mathbf{u}) = \sum_{i \in A_k(\mathcal{D}) \cap U} u_i, \tag{7.15}$$

and

$$S_{\mathcal{D}k}(\mathbf{v}) = \sum_{i \in A_k(\mathcal{D}) \cap V} v_i. \tag{7.16}$$

From (7.11), (7.13) and (7.15) we have that $S_{\mathcal{C}k}(\mathbf{u}) \leq S_{\mathcal{D}k}(\mathbf{u})$, and similary, from (7.12), (7.14) and (7.16) we have that $S_{\mathcal{C}k}(\mathbf{v}) \geq S_{\mathcal{D}k}(\mathbf{v})$. Moreover, if $A_k(\mathcal{C}) \neq A_k(\mathcal{D})$, then at least one of the inequalities is strict. But since $\mathbf{S}_{\mathcal{C}}(\mathbf{u}) = \mathbf{S}_{\mathcal{C}}(\mathbf{v})$ and $\mathbf{S}_{\mathcal{D}}(\mathbf{u}) = \mathbf{S}_{\mathcal{D}}(\mathbf{v})$, this is a contradiction. □

Using this lemma, for a fixed $\mathbf{u}$ and $\mathbf{v}$ with $\mathbf{u}^\top \mathbf{v} = 0$, we can get the probability of $\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})$. Since the distribution over the codes is uniform, we can simply enumerate the suitable codes and divide it by the number of all possible codes:

$$\mathbf{P}\left(\{\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v})\}\right)=\frac{\left|\{\mathcal{C}\in(\mathcal{B}^n)^t:\mathbf{S}_{\mathcal{C}}(\mathbf{u})=\mathbf{S}_{\mathcal{C}}(\mathbf{v})\}\right|}{\left|\{(\mathcal{B}^n)^t\}\right|}.$$

Let $\ell=|U|+|V|$. Consider the k^{th} components of the codewords. For users not in $U\cup V$, the k^{th} components can be arbitrary, this does not affect $\mathbf{S}_{\mathcal{C}}(\mathbf{u})=\mathbf{S}_{\mathcal{C}}(\mathbf{v})$. This creates $2^{t-\ell}$ possibilities. For the k^{th} components of the codewords of users in $U\cup V$, we have at most as many possibilities as many incomparable subsets exists of $U\cup V$. This is $\binom{\ell}{\lfloor\ell/2\rfloor}$ by the theorem of Sperner (1928). Thus

$$\begin{aligned}\mathbf{P}\left(\{\mathbf{S}(\mathbf{u})=\mathbf{S}(\mathbf{v})\}\right)&\le\frac{\left(2^{t-\ell}\binom{\ell}{\lfloor\ell/2\rfloor}\right)^n}{2^{tn}}\\&=\left(2^{-\ell}\binom{\ell}{\lfloor\ell/2\rfloor}\right)^n\\&\le\left(\frac{1}{\sqrt{\pi\frac{\ell}{2}}}\right)^n\\&=2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}.\end{aligned}\tag{7.17}$$

Returning to (7.10), we get

$$\begin{aligned}\mathbf{P}\big(\text{event }(*)\big)&\le\sum_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\ldots,t\}^t:\\ \mathbf{u}^\top\mathbf{v}=0,\sum_{i=1}^t u_i=\sum_{i=1}^t v_i\le t}}2^{-\frac{n}{2}\log\frac{\pi|U\cup V|}{2}}\\&=\sum_{\ell=1}^t\sum_{u,v\colon u+v=\ell}\sum_{\substack{U,V\subseteq\{1,2,\ldots,t\}:\\|U|=u,|V|=v,\\U\cap V=\emptyset}}\sum_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\ldots,t\}^t:\\ \sum_{i=1}^t u_i=\sum_{i=1}^t v_i\le t\\ \{i\colon u_i\ne0\}=U\\ \{i\colon v_i\ne0\}=V}}2^{-\frac{n}{2}\log\frac{\pi\ell}{2}},\end{aligned}$$

where we enumerated the possible vectors $\mathbf{u}$ and $\mathbf{v}$ based on the size of their base set U and V. Thus

$$\begin{aligned}\mathbf{P}\left(\text{event }*\right)&\le\sum_{\ell=1}^t\sum_{u,v\colon u+v=\ell}\binom{t}{u}\binom{t-u}{v}\binom{t}{u}\binom{t}{v}2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}\\&\le\sum_{\ell=1}^t\binom{t}{\ell}2^\ell\sum_{u,v\colon u+v=\ell}\binom{t}{u}\binom{t}{v}2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}\end{aligned}$$

$$= \sum_{\ell=1}^{t} \binom{t}{\ell} 2^{\ell} \binom{2t}{\ell} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}. \tag{7.18}$$

Split the summation into two parts:

$$\ell = 1, 2, \ldots, \left\lfloor \frac{t}{\log^2 t} \right\rfloor,$$

and

$$\ell = \left\lfloor \frac{t}{\log^2 t} \right\rfloor + 1, \left\lfloor \frac{t}{\log^2 t} \right\rfloor + 2, \ldots, t.$$

For the first part we use $\binom{t}{\ell} \le t^{\ell}$ and $\binom{2t}{\ell} \le (2t)^{\ell}$:

$$\begin{aligned} \sum_{\ell=1}^{\left\lfloor \frac{t}{\log^2 t} \right\rfloor} \binom{t}{\ell} 2^{\ell} \binom{2t}{\ell} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}} &\le \sum_{\ell=1}^{\left\lfloor \frac{t}{\log^2 t} \right\rfloor} t^{2\ell} 2^{2\ell - \frac{n}{2}\log\frac{\pi\ell}{2}} \\ &\le \frac{t}{\log^2 t} t^{\frac{2t}{\log^2 t}} 2^{\frac{2t}{\log^2 t} - \frac{n}{2}\log\frac{\pi}{2}} \\ &= 2^{\log\frac{t}{\log^2 t} + \frac{2t}{\log^2 t}(1+\log t) - \frac{n}{2}\log\frac{\pi}{2}}, \end{aligned}$$

which tends to 0 as $t \to \infty$ if

$$\lim_{t\to\infty} \left(\log\frac{t}{\log^2 t} + \frac{2t(1+\log t)}{\log^2 t} - \frac{n}{2}\log\frac{\pi}{2} \right) = -\infty. \tag{7.19}$$

For the second part of the summation in (7.18), we use $\binom{t}{\ell} \le 2^{t}$, $2^{\ell} \le 2^{t}$ and $\binom{2t}{\ell} \le 2^{2t}$:

$$\begin{aligned} \sum_{\ell=\left\lfloor \frac{t}{\log^2 t} \right\rfloor + 1}^{t} \binom{t}{\ell} 2^{\ell} \binom{2t}{\ell} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}} &\le \sum_{\ell=\left\lfloor \frac{t}{\log^2 t} \right\rfloor + 1}^{t} 2^{4t - \frac{n}{2}\log\frac{\pi\ell}{2}} \\ &\le \sum_{\ell=1}^{t} 2^{4t - \frac{n}{2}\log\frac{\pi t}{2\log^2 t}} \\ &\le t 2^{4t - \frac{n}{2}\log\frac{\pi t}{2\log^2 t}} \\ &= 2^{\log t + 4t - \frac{n}{2}\log\frac{\pi t}{2\log^2 t}}, \end{aligned}$$

which tends to 0 as $t \to \infty$ if

$$\lim_{t\to\infty} \left(\log t + 4t - \frac{n}{2}\log\frac{\pi t}{2\log^2 t} \right) = -\infty. \tag{7.20}$$

Let us set

$$n = \left\lceil \frac{ct}{\log t} \right\rceil .$$

In the first condition (7.19) this yields

$$\begin{aligned}
&\lim_{t\to\infty}\left(\log\frac{t}{\log^2 t}+\frac{2t(1+\log t)}{\log^2 t}-\frac{n}{2}\log\frac{\pi}{2}\right)\\
&\qquad\le \lim_{t\to\infty}\left(\log\frac{t}{\log^2 t}+\frac{2t(1+\log t)}{\log^2 t}-\frac{ct}{2\log t}\log\frac{\pi}{2}\right)\\
&\qquad= \lim_{t\to\infty}\left(\frac{t}{\log t}\left(\frac{\log\frac{t}{\log^2 t}}{\frac{t}{\log t}}+\frac{2(1+\log t)}{\log t}-\frac{c}{2}\log\frac{\pi}{2}\right)\right)\\
&\qquad= \left(2-\frac{c}{2}\log\frac{\pi}{2}\right)\lim_{t\to\infty}\frac{t}{\log t},
\end{aligned}$$

which is $-\infty$ if $c > \frac{4}{\log\frac{\pi}{2}} \approx 6.140$. In the second condition (7.20),

$$\begin{aligned}
&\lim_{t\to\infty}\left(\log t+4t-\frac{n}{2}\log\frac{\pi t}{2\log^2 t}\right)\\
&\qquad\le \lim_{t\to\infty}\left(\log t+4t-\frac{ct}{2\log t}\log\frac{\pi t}{2\log^2 t}\right)\\
&\qquad= \lim_{t\to\infty}\left(t\left(\frac{\log t}{t}+4-\frac{c}{2\log t}\log\frac{\pi t}{2\log^2 t}\right)\right)\\
&\qquad= \left(4-\frac{c}{2}\right)\lim_{t\to\infty} t,
\end{aligned}$$

which is $-\infty$ if $c > 8$.

So we have shown, that for all $\varepsilon > 0$ and $n = \left\lceil \frac{(8+\varepsilon)t}{\log t} \right\rceil$,

$$\lim_{t\to\infty} \mathbf{P}\big(\text{event } (*)\big) = 0.$$

This means, that for t large enough, the random code of length n we select is a u.d. signature code with multiplicity with positive probability. So there exists such a code of that length:

$$N_{tm}(t) \le \left\lceil \frac{(8+\varepsilon)t}{\log t} \right\rceil \le \frac{(8+\varepsilon)t}{\log t} + 1,$$

or equivalently

$$\limsup_{t\to\infty} \frac{N_{tm}(t)\log t}{t} \le 8.$$

□

7.8 A Lower Bound

From our trivial lower bound (Theorem 7.1) and the upper bound of Erdős and Rényi (Theorem 7.2), for the minimal length of u.d. codes for the adder channel we have

$$1 \leq \liminf_{t\to\infty} \frac{N(t)\log t}{t} \leq \limsup_{t\to\infty} \frac{N_s(t)\log t}{t} \leq \log 9,$$

while the truth is that

$$\lim_{t\to\infty} N(t)\frac{\log t}{t} = \lim_{t\to\infty} N_s(t)\frac{\log t}{t} = 2.$$

Here we present an improved lower bound for the limes inferior. The upper bound for the limes superior will follow from the construction of Lindström (see Section 7.10).

Theorem 7.4. *(Chang–Weldon (1979))*

$$\liminf_{t\to\infty} \frac{N(t)\log t}{t} \geq 2.$$

For the proof, we will need a lemma which is following from a bound of the discrete entropy (cf. e.g. Cover–Thomas (1991) Theorem 9.7.1 pp. 235.):

Lemma 7.2. *Let X have an arbitrary distribution over the integers:*

$$\mathbf{P}(X = i) = p_i \qquad \forall i \in \mathbb{Z},$$
$$\sum_{i\in\mathbb{Z}} p_i = 1.$$

If X has variance $Var(X)$ and Shannon–entropy $\mathbf{H}(X) = -\sum_{i\in\mathbb{Z}} p_i \log p_i$, then

$$\mathbf{H}(X) \leq \frac{1}{2}\log\left(2\pi e\left(Var(X) + \frac{1}{12}\right)\right).$$

Proof of Theorem 7.4. We will bound the same entropy in two different ways, and this will yield the bound. Consider an arbitrary u.d. code for the adder channel. Let us define the message-constellation $\mathbf{M}$ as a random vector variable with uniform distribution over all the possible constellations:

$$\mathbf{P}(\mathbf{M} = \mathbf{m}) = \frac{1}{2^t} \qquad \forall \mathbf{m} \in \{1,2\}^t.$$

Since the code is u.d., the received vector $\mathbf{Y}$ must be different for all different $\mathbf{M}$, so

$$\mathbf{H}(\mathbf{Y}) = \mathbf{H}(\mathbf{M}) = t. \tag{7.21}$$

On the other hand, the entropy of $\mathbf{Y}$ can be upper bounded by the sum of the entropies of its components. Each component of the vector $\mathbf{Y}$ has a binomial distribution. To show this, considering only the j^{th} bit of the codewords, and split the user population into three groups. For the users in the first group both codewords has 0 in the j^{th} position. For the users in the second group, one codeword is 0 and the other is 1 at the j^{th} position, while for the third group both codewords are 1 at the j^{th} position. If we denote the number of users in the first group with a_j, in the second group with b_j and in the third group with c_j the we can write

$$\mathbf{P}\left(Y_j = s\right) = \frac{2^{a_j}\binom{b_j}{s-c_j}2^{c_j}}{2^{a_j+b_j+c_j}} = \binom{b_j}{s-c_j}2^{-b_j} \qquad \forall s\colon c_j \leq s \leq b_j + c_j,$$

where 2^{a_j} is the number of possible constellations for the users in the first group, $\binom{b_j}{s-c_j}$ is the number of possible constellations for users in the second group (select exactly $s - c_j$ users out of the b_i ones who have both zero and one at position j), and 2^{c_j} is the number of possible constellations for users in the third group. $2^{a_j+b_j+c_j}$ is the total number of possible constellations.

Now we can write

$$\mathbf{H}\left(\mathbf{Y}\right) \leq \sum_{j=1}^{n} \mathbf{H}\left(Y_j\right),$$

and using Lemma 7.2,

$$\mathbf{H}\left(\mathbf{Y}\right) \leq \sum_{j=1}^{n} \frac{1}{2}\log\left(2\pi e\left(\mathrm{Var}\left(Y_j\right) + \frac{1}{12}\right)\right),$$

Since the variance of the binomial distribution with parameters $\left(b_j, \frac{1}{2}\right)$ is $\frac{b_j}{4}$, we have

$$\mathbf{H}\left(\mathbf{Y}\right) \leq \sum_{j=1}^{n} \frac{1}{2}\log\left(2\pi e\left(\frac{b_j}{4} + \frac{1}{12}\right)\right),$$

and since $\frac{b_j}{4} + \frac{1}{12} \leq \frac{t}{4} + \frac{1}{12} \leq \frac{t}{2}$,

$$\mathbf{H}\left(\mathbf{Y}\right) \leq \frac{n \log \pi e t}{2}. \tag{7.22}$$

Putting (7.21) and (7.22) together we get that

$$\frac{n \log \pi e t}{t} \geq 2,$$

which also holds for the code of minimal possible length $N(t)$:

$$\frac{N(t)\log t}{t} \geq 2\frac{\log \pi e t}{\log t},$$

thus

$$\liminf_{t\to\infty} \frac{N(t)\log t}{t} \geq 2.$$

□

7.9 Chang and Weldon's Construction

In this section we show the u.d. code construction of Chang and Weldon (1979) for the binary multiple-access adder channel. Their code construction is for $t_k = (k+2)2^{(k-1)}$ users, where k is an arbitrary natural number. Their code length is $n_k = 2^k$. If we put these together, we get

$$\lim_{k\to\infty} \frac{n_k \log t_k}{t_k} = 2,$$

from which

$$\limsup_{k\to\infty} \frac{N(t_k)\log t_k}{t_k} \leq 2.$$

First, we present a difference matrix representation of codes with two codewords per user. Given an arbitrary code

$$\mathcal{C} = \Big\{\{\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}\}, \{\mathbf{x}_1^{(2)}, \mathbf{x}_2^{(2)}\}, \ldots, \{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\}\Big\},$$

we can create a so called difference matrix $\mathbf{D}$ by writing the differences of the two vectors in the component codes next to each other:

$$\mathbf{D} = \left(\mathbf{x}_2^{(1)} - \mathbf{x}_1^{(1)}, \mathbf{x}_2^{(2)} - \mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_2^{(t)} - \mathbf{x}_1^{(t)}\right)$$

It is easy to see, that given a difference matrix $\mathbf{D}$ (a matrix with all elements from $\{-1, 0, 1\}$) we can construct at least one code, for which this is the difference matrix. E.g. for all user $i \in [t]$ and for all code bit $j \in [n]$ let

$$x_{1j}^{(i)} = \begin{cases} 1 & \text{if } D_{ij} = -1; \\ 0 & \text{if } D_{ij} = 0; \\ 0 & \text{if } D_{ij} = 1, \end{cases} \quad \text{and } x_{2j}^{(i)} = \begin{cases} 0 & \text{if } D_{ij} = -1; \\ 0 & \text{if } D_{ij} = 0; \\ 1 & \text{if } D_{ij} = 1. \end{cases}$$

Certainly we could also use 1 in both codewords if $D_{ij} = 0$.

It is easy to show (with the same reasoning given in Section 7.4 at the matrix representation), that that the u.d. property can be expressed in the following property of matrix $\mathbf{D}$:

$$\mathbf{D}\mathbf{w} = \mathbf{0} \iff \mathbf{w} = \mathbf{0} \qquad \forall \mathbf{w} \in \{-1, 0, 1\}^t. \tag{7.23}$$

We have trivial u.d. difference matrices for $t = 1$ ($\mathbf{D}_0$) and for $t = 3$ ($\mathbf{D}_1$):

$$\mathbf{D}_0 = \begin{pmatrix} 1 \end{pmatrix}, \mathbf{D}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{pmatrix}.$$

Moreover, we can find a recursive construction for u.d. difference matrices $\mathbf{D}_k$ for $t = (k+2)2^{k-1}$ users. This is stated as the next theorem.

Theorem 7.5. *(Chang–Weldon (1979)) The $\mathbf{D}_k$ matrices defined below are u.d. difference matrices for $t_k = (k+2)2^{k-1}$ users with code length $n_k = 2^k$:*

$$\mathbf{D}_0 = \begin{pmatrix} 1 \end{pmatrix}$$
$$\forall k \geq 1 : \mathbf{D}_k = \begin{pmatrix} \mathbf{D}_{k-1} & \mathbf{D}_{k-1} & \mathbf{I}_{2^{k-1}} \\ \mathbf{D}_{k-1} & -\mathbf{D}_{k-1} & \mathbf{0}_{2^{k-1}} \end{pmatrix}$$

Proof. It is easy to see by induction, that $\mathbf{D}_k$ has $t_k = (k+2)2^{k-1}$ columns, since the two matrices $\mathbf{D}_{k-1}$ have two times $(k+1)2^{k-2}$ columns, and the additional identity and zero matrices have 2^{k-1} columns. The sum is really $t_k = (k+2)2^{k-1}$.

Now let us suppose, that for a given $\mathbf{w} \in \{-1, 0, 1\}^{t_k}$,

$$\mathbf{D}_k \mathbf{w} = \mathbf{0}.$$

We decompose vector $\mathbf{w}$ of length t_k into three vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ of length $(k+1)2^{k-2}$, $(k+1)2^{k-2}$, and 2^{k-1} respectively:

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{pmatrix}.$$

Now using the recursive definition of $\mathbf{D}_k$ we get

$$\begin{pmatrix} \mathbf{D}_{k-1} & \mathbf{D}_{k-1} & \mathbf{I}_{2^{k-1}} \\ \mathbf{D}_{k-1} & -\mathbf{D}_{k-1} & \mathbf{0}_{2^{k-1}} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$

which gives two equations:

$$\mathbf{D}_{k-1}\mathbf{w}_1 + \mathbf{D}_{k-1}\mathbf{w}_2 + \mathbf{w}_3 = \mathbf{0},$$
$$\mathbf{D}_{k-1}\mathbf{w}_1 - \mathbf{D}_{k-1}\mathbf{w}_2 = \mathbf{0}.$$

From this we have

$$2\mathbf{D}_{k-1}\mathbf{w}_1 = -\mathbf{w}_3$$

and since $\mathbf{w}_3$ has components in $\{-1, 0, 1\}$, it follows that $\mathbf{w}_3 = \mathbf{0}$, and therefore

$$\begin{aligned}\mathbf{D}_{k-1}\mathbf{w}_1 &= \mathbf{0}\\ \mathbf{D}_{k-1}\mathbf{w}_2 &= \mathbf{0}.\end{aligned}$$

But $\mathbf{D}_{k-1}$ has property (7.23) by induction, so

$$\begin{aligned}\mathbf{w}_1 &= \mathbf{0},\\ \mathbf{w}_2 &= \mathbf{0}.\end{aligned}$$

follows, and the proof is complete. □

Chang and Weldon (1979) have also given a decoding algorithm for their code. It is the following: let us suppose, that user i sends message m_i, so the received vector $\mathbf{y}$ is

$$\mathbf{y} = \sum_{i=1}^{t} \mathbf{x}_{m_i}^{(i)}$$

Introduce vector $\mathbf{u} = \big(u_1, u_2, \ldots, u_t\big) \in \mathcal{B}^{t_k}$ which has 0 in the positions corresponding to users sending their "message 1" and 1 in the positions corresponding to users sending their "message 2", so

$$u_i = \begin{cases} 0 & \text{if } m_i = 1;\\ 1 & \text{if } m_i = 2 \end{cases} \qquad \forall i \in [t_k].$$

Then we can write

$$\mathbf{y} = \mathbf{D}_k\mathbf{u} + \sum_{i=1}^{t} \mathbf{x}_1^{(i)},$$

and therefore

$$\mathbf{D}_k\mathbf{m} = \mathbf{y} - \sum_{i=1}^{t} \mathbf{x}_1^{(i)}. \tag{7.24}$$

Now we make a decomposition:

$$\mathbf{y} - \sum_{i=1}^{t} \mathbf{x}_1^{(i)} = \begin{pmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{pmatrix} = \begin{pmatrix}\mathbf{D}_{k-1} & \mathbf{D}_{k-1} & \mathbf{I}_{2^{k-1}}\\ \mathbf{D}_{k-1} & -\mathbf{D}_{k-1} & \mathbf{0}_{2^{k-1}}\end{pmatrix}\begin{pmatrix}\mathbf{u}_1\\ \mathbf{u}_2\\ \mathbf{u}_3\end{pmatrix},$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ has 2^{k-1} rows, $\mathbf{u}_1$ and $\mathbf{u}_2$ has $(k+1)2^{k-2}$ rows, and $\mathbf{u}_3$ has 2^{k-1} rows. This equation yields

$$\mathbf{v}_1 + \mathbf{v}_2 = 2\mathbf{D}_{k-1}\mathbf{u}_1 + \mathbf{u}_3.$$

Then

$$\mathbf{v}_1 + \mathbf{v}_2 \equiv \mathbf{u}_3 \qquad \bmod 2,$$

and from this we can get $\mathbf{u}_3$, since it is a binary vector. We have also

$$\mathbf{D}_{k-1}\mathbf{u}_1 = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2 - \mathbf{u}_3),$$
$$\mathbf{D}_{k-1}\mathbf{u}_2 = \frac{1}{2}(\mathbf{v}_1 - \mathbf{v}_2 - \mathbf{u}_3),$$

where the right hand side of the equations are known vectors. These are two instances of the same problem but for $k-1$. We can recursively continue at formula (7.24). At last, for the case $k=0$ we will have a trivial equation for $\mathbf{u}$ since $\mathbf{D}_0 = (1)$.

7.10 Lindström's Construction

The last construction was for general u.d. multiple access codes for the adder channel. For u.d. signature codes, Lindström (1964; 1964) presented a construction. His code is asymptotically optimal, since it sets an upper bound on the minimal u.d. signature code length which is equal to the lower bound in Theorem 7.4.

We will construct a u.d. signature code for a given n. The number of users in the code will be denoted by $t(n)$, and it holds, that

$$t(n) \sim \frac{n \log n}{2}.$$

If we need a code for t users, select the smallest n for which $t(n) \geq t$, create a code for this n, keep only the codewords for t users, and simply drop other codewords.

To show that this code is assymptotically optimal, we will invert the relation between n and $t(n)$, and give an upper bound for the length of the shortest possible code for t users. This way we will get the next theorem.

Theorem 7.6. *(Lindström (1964)) The construction of Lindström gives u.d. signature codes of arbitrary length, and for code length n, the number of users is $t(n)$ users, for which*

$$\lim_{n\to\infty} \frac{t(n)}{n \log n} = \frac{1}{2}.$$

From this, for the length $N(t)$ of the shortest possible u.d. signature code for t users

$$\limsup_{t\to\infty} \frac{N(t)\log t}{t} \le 2$$

follows.

Proof. In the construction we will use the binary representation of positive integers: if $a \in \mathbb{N}$ is a positive integer, then let its binary representation be $a_{\lfloor \log a \rfloor} \dots a_1 a_0$:

$$a = \sum_{i=0}^{\lfloor \log a \rfloor} 2^i a_i.$$

The set of the nonzero positions in the binary form of a is detoted by $D(a)$:

$$D(a) = \{i\colon a_i = 1\}.$$

Then we can write

$$a = \sum_{i \in D(a)} 2^i.$$

Let us introduce

$$\alpha(a) = |D(a)|,$$

and define

$$a \subseteq b \iff D(a) \subseteq D(b);$$
$$a \cap b = \sum_{i \in D(a)\cap D(b)} 2^i.$$

For a given n, for each $r \in \{1, 2, \dots, n\}$ construct a matrix $\mathbf{C}^{(r)}$ of size $r \times \alpha(r)$ in the following way: for row i where $i \subseteq r$ select the $\alpha(r)$ elements $c_{ij}^{(r)} \in \mathcal{B}$ such that

$$\sum_{i \subseteq r} (-1)^{\alpha(i)+1} c_{ij}^{(r)} = 2^{j-1}. \tag{7.25}$$

This is possible, since there are $2^{\alpha(r)}$ terms in the sum, and in one half of these terms (-1) has even exponent, in the other half it has an odd exponent. This yields a $(+1)$ coefficient for $c_{ij}^{(r)}$ exactly $2^{\alpha(r)-1}$ times, and a (-1) coefficient exactly the same times. Thus selecting $c_{ij}^{(r)} = 0$ for the terms with negative coefficients, and selecting $c_{ij}^{(r)} = 1$ for the positive ones, we get a sum of $2^{\alpha(r)-1}$ for $i \subseteq r$. Since $2^{\alpha(r)-1}$ is an upper bound for 2^{j-1} (note that $j-1 \leq \alpha(r)-1$), so selecting only some of the $c_{ij}^{(r)}$s with $(+1)$ coefficients to 1, and leaving all other $c_{ij}^{(r)}$ to be 0 is a possible solution.

For all the other rows (where $i \nsubseteq r$), let

$$c_{ij}^{(r)} = c_{i \cap r\, j}^{(r)},$$

which is known from the above equation (7.25), since $i \cap r \subseteq r$.

Create a code submatrix $\mathbf{C}^{(r)}$ as

$$\mathbf{C}^{(r)} = \begin{pmatrix} c_{11}^{(r)} & c_{12}^{(r)} & \cdots & c_{1t_r}^{(r)} \\ c_{21}^{(r)} & c_{22}^{(r)} & \cdots & c_{2t_r}^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1}^{(r)} & c_{n2}^{(r)} & \cdots & c_{nt_r}^{(r)} \end{pmatrix}.$$

Create the code matrix $\mathbf{C}$ as concatenating these $\mathbf{C}^{(r)}$ matrices for all $r = 1, 2, \ldots, n$:

$$\mathbf{C} = \left(\mathbf{C}^{(1)} \middle| \mathbf{C}^{(2)} \middle| \ldots \middle| \mathbf{C}^{(n)}\right).$$

To get the code $\mathcal{C}$, consider the columns of the code matrix $\mathbf{C}$ as the codewords. Since $\mathbf{C}$ will have n rows, the codewords are of length n. Calculate the number of columns, which is the number of users in the code, as the following

$$t(n) = \sum_{r=1}^{n} t_r = \sum_{r=1}^{n} \alpha(r). \tag{7.26}$$

The expression $\sum_{r=1}^{n} \alpha(r)$ is the number of "1"s in the binary representations of all numbers between 1 and n. To calculate this sum, divide it to summation segments in the following way. Let

$$g_1 > g_2 > \ldots > g_{\alpha(n)}$$

denote the elements of $D(n)$. Then as we have already mentioned,

$$n = \sum_{i=1}^{\alpha(n)} 2^{g_i}.$$

Thus

$$\sum_{r=1}^{n} \alpha(r) = \sum_{r=0}^{2^{g_1}-1} \alpha(r) + \sum_{r=2^{g_1}}^{2^{g_1}+2^{g_2}-1} \alpha(r) + \ldots + \sum_{r=2^{g_1}+2^{g_2}+\ldots+2^{g_{\alpha(n)-1}}}^{2^{g_1}+2^{g_2}+\ldots+2^{g_{\alpha(n)}}-1} \alpha(r) + \alpha(n).$$

Consider only one summation segment: in the sum

$$\sum_{r=2^{g_1}+2^{g_2}+\ldots+2^{g_{i-1}}}^{2^{g_1}+2^{g_2}+\ldots+2^{g_i}-1} \alpha(r)$$

we sum the number of “1”s in the binary representations of 2^{g_i} numbers. Each of these numbers have $i-1$ fixed “1” digits at the positions $g_1, g_2, \ldots, g_{i-1}$, and the last g_i digits take all possible values:

$$\begin{array}{lcccccccccccccccc}
\text{positions:} & & \underset{\smile}{g_1} & \cdots\cdots & & \underset{\smile}{g_2} & \cdots\cdots & & \underset{\smile}{g_{i-1}} & \cdots\cdots & & \underset{\smile}{g_i} & \cdots\cdots & & \underset{\smile}{1} & \underset{\smile}{0} \\
2^{g_1}+\cdots+2^{g_{i-1}}+2^{g_i}-1 & = & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 0 & 1 & \cdots & 1 & 1 \\
 & & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
2^{g_1}+\cdots+2^{g_{i-1}}+1 & = & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 \\
2^{g_1}+\cdots+2^{g_{i-1}} & = & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0
\end{array}$$

Here we can clearly determine the number of “1”s: this is $2^{g_i}\frac{g_i}{2}$ for the last g_i digits (half of all digits in this block is “1”, and $2^{g_i}(i-1)$ for the $i-1$ fixed digits.

Now returning to $\sum_{r=1}^{n} \alpha(r)$, this yields

$$\sum_{r=1}^{n} \alpha(r) = \sum_{i=1}^{\alpha(n)} 2^{g_i} \left(\frac{g_i}{2} + i - 1 \right) + \alpha(n),$$

where $g_1 > g_2 > \cdots > g_{\alpha(n)}$ are the elements of $D(n)$, satisfying

$$n = 2^{g_1} + 2^{g_2} + \cdots + 2^{g_{\alpha(n)}}.$$

We will show, that

$$t(n) = \sum_{r=1}^{n} \alpha(r) \sim \frac{n\lfloor \log n \rfloor}{2} \sim \frac{n \log n}{2}$$

as $n \to \infty$. The latter assymptotic equality is trivial. For the former, observe first that $\lfloor \log n \rfloor = g_1$, and then take the difference of the left and the right side of the assymptotics, divided by the right hand side:

$$\begin{aligned}\frac{\sum_{r=1}^{n}\alpha(r)-\frac{n\lfloor\log n\rfloor}{2}}{\frac{n\lfloor\log n\rfloor}{2}} &= \frac{\sum_{i=1}^{\alpha(n)}\left(\frac{g_i-g_1}{2}+i-1\right)2^{g_i}+\alpha(n)}{\frac{\sum_{i=1}^{\alpha(n)}g_1 2^{g_i}}{2}}\\ &= \frac{\sum_{i=1}^{\alpha(n)}(g_i-g_1+2i-2)2^{g_i}+2\alpha(n)}{\sum_{i=1}^{\alpha(n)}g_1 2^{g_i}} =\\ &= \frac{\sum_{i=1}^{\alpha(n)}(i-1-f_i)2^{-f_i+1-i}+\frac{2\alpha(n)}{2^{g_1}}}{\sum_{i=1}^{\alpha(n)}g_1 2^{-f_i+1-i}}\end{aligned}$$

where we introduced $f_i = g_1 - g_i - i + 1$, and divided by 2^{g_1}. Notice, that $f_i \geq 0$, and use that $(c-x)2^{-x}$ for $x \geq 0$ is maximized at $x = 0$ with value c, and minimized at $x = \log e + c$ with value $-\frac{\log e}{e}2^{-c}$. Then

$$\begin{aligned}\frac{\sum_{r=1}^{n}\alpha(r)-\frac{n\lfloor\log n\rfloor}{2}}{\frac{n\lfloor\log n\rfloor}{2}} &\leq \frac{\sum_{i=1}^{\alpha(n)}(i-1)2^{1-i}+\frac{2\alpha(n)}{2^{g_1}}}{g_1+\sum_{i=2}^{\alpha(n)}g_1 2^{-f_i+1-i}}\\ &\leq \frac{\text{bounded}}{g_1}\\ &= \frac{\text{bounded}}{\lfloor\log n\rfloor},\end{aligned}$$

which tends to 0 as $n \to \infty$. For the lower bound:

$$\begin{aligned}\frac{\sum_{r=1}^{n}\alpha(r)-\frac{n\lfloor\log n\rfloor}{2}}{\frac{n\lfloor\log n\rfloor}{2}} &\geq \frac{\sum_{i=1}^{\alpha(n)}-\frac{\log e}{e}2^{1-i}2^{1-i}+\frac{2\alpha(n)}{2^{-g_1}}}{2g_1}\\ &= \frac{-\frac{\log e}{e}\sum_{i=1}^{\alpha(n)}4^{1-i}+\frac{2\alpha(n)}{2^{-g_1}}}{2g_1}\\ &= \frac{\text{bounded}}{2g_1}\\ &= \frac{\text{bounded}}{2\lfloor\log n\rfloor}.\end{aligned}$$

Thus continuing (7.26), we get that this construction yields a maximal number of $t(n)$ users for a given n for which

$$t(n) \sim \frac{n\log n}{2},$$

or more precisely

$$\lim_{n\to\infty}\frac{t(n)}{n\log n} = \frac{1}{2}. \tag{7.27}$$

Now we will prove from this, that for the length $N(t)$ of the shortest possible u.d. code for t users

$$N(t) \lesssim \frac{2t}{\log t},$$

or precisely

$$\limsup_{t\to\infty} \frac{N(t)\log t}{t} \le 2.$$

To see this, we will set an arbitrary $\varepsilon > 0$, and show that

$$N'(t) = \frac{2(1+\varepsilon)t}{\log t}$$

is an upper bound for $N(t)$ if t is great enough.

Let $\delta = \sqrt{1+\varepsilon} - 1$ and let $\delta' = 1 - \frac{1}{1+\delta}$. Then we can write

$$N'(t) = \frac{2(1+\delta)t}{(1-\delta')\log t}.$$

From (7.27) we know, that there exists an n_0, for which

$$\forall n > n_0 \colon t(n) \ge \frac{(1-\delta')n\log n}{2}.$$

Since $N'(t) \to \infty$ as $t \to \infty$, there also exists a $t_0^{(1)}$, for which

$$\begin{aligned}\forall t > t_0^{(1)} \colon t(N'(t)) &\ge \frac{(1-\delta')N'(t)\log N'(t)}{2}\\ &= \frac{(1-\delta')\frac{2(1+\delta)t}{(1-\delta')\log t}\left(\log\frac{2(1+\delta)}{(1-\delta')} + \log t - \log\log t\right)}{2}\\ &= \frac{t(1+\delta)\left(\log\frac{2(1+\delta)}{(1-\delta')} + \log t - \log\log t\right)}{\log t}.\end{aligned}$$

It is easy to see, that this last formula is approximatelly equal to $(1+\delta)t$, so there exists a $t_0^{(2)}$ for which

$$\forall t > \max\{t_0^{(1)}, t_0^{(2)}\} \colon t(N'(t)) > t,$$

which means that if t is great enough, then for code length $N'(t)$ the above construction gives a u.d. signature code for more than t users. Then this

$N'(t)$ is an upper bound for the length of the shortest possible u.d. signature code for t users.

$$N(t) \le N'(t) = \frac{2(1+\varepsilon)t}{\log t}.$$

Since $\varepsilon > 0$ is arbitrary, this yields

$$\limsup_{t\to\infty} \frac{N(t)\log t}{t} \le 2.$$

What remains to see, is that the above constructed $\mathbf{C}$ is really a u.d. code matrix. Let us consider the contrary: suppose that $\mathbf{m} \ne \mathbf{n}$ is two constellations ($\mathbf{m}, \mathbf{n} \in \{0,1\}^t$) for which $\mathbf{S}(\mathbf{m}) = \mathbf{S}(\mathbf{n})$, or equivalently, $\mathbf{Cm} = \mathbf{Cn}$. Decompose the constellation vectors as

$$\mathbf{m} = \begin{pmatrix} \mathbf{m}^{(1)} \\ \mathbf{m}^{(2)} \\ \vdots \\ \mathbf{m}^{(n)} \end{pmatrix}, \ \mathbf{n} = \begin{pmatrix} \mathbf{n}^{(1)} \\ \mathbf{n}^{(2)} \\ \vdots \\ \mathbf{n}^{(n)} \end{pmatrix},$$

where $\mathbf{m}^{(r)}$ and $\mathbf{n}^{(r)}$ is of size t_r.

Since $\mathbf{m} \ne \mathbf{n}$, there is some r, for which $\mathbf{m}^{(r)} \ne \mathbf{n}^{(r)}$. Let ℓ be the maximal r with this property.

Let $\mathbf{y} = (x_1, x_2, \dots, x_n)^\top$ be defined by

$$x_i = \begin{cases} (-1)^{\alpha(i)+1} & \text{if } i \subseteq \ell; \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mathbf{Cm} = \mathbf{Cn}$, we have that $\mathbf{y}^\top\mathbf{Cm} - \mathbf{y}^\top\mathbf{Cn} = \mathbf{y}^\top\mathbf{C}(\mathbf{m}-\mathbf{n}) = \mathbf{0}$. Thus

$$\mathbf{0} = \mathbf{y}^\top\mathbf{C}(\mathbf{m}-\mathbf{n}) = \sum_{r=1}^{n} \mathbf{y}^\top\mathbf{C}^{(r)}(\mathbf{m}^{(r)} - \mathbf{n}^{(r)}),$$

where we separate the sum into three partitions: $r \supsetneq \ell$, $r = \ell$ and $r \not\supseteq \ell$:

$$\begin{aligned} \mathbf{y}^\top\mathbf{C}(\mathbf{m}-\mathbf{n}) = &\sum_{r \supsetneq \ell} \mathbf{y}^\top\mathbf{C}^{(r)}(\mathbf{m}^{(r)} - \mathbf{n}^{(r)}) \\ &+ \mathbf{y}^\top\mathbf{C}^{(\ell)}(\mathbf{m}^{(\ell)} - \mathbf{n}^{(\ell)}) \\ &+ \sum_{r \not\supseteq \ell} \mathbf{y}^\top\mathbf{C}^{(r)}(\mathbf{m}^{(r)} - \mathbf{n}^{(r)}). \end{aligned}$$

For $r \supsetneq \ell$, we know, that $r > \ell$, thus $\mathbf{m}^{(r)} - \mathbf{n}^{(r)} = \mathbf{0}$ (ℓ was defined as the largest index r for which $\mathbf{m}^{(r)}$ and $\mathbf{n}^{(r)}$ differs), so the sum of the first partition is $\mathbf{0}$.

For $r = \ell$,

$$\begin{aligned}\mathbf{y}^\top \mathbf{C}^{(\ell)}(\mathbf{m}^{(\ell)} - \mathbf{n}^{(\ell)}) &= \sum_{j=1}^{t_\ell} \left(\sum_{i \subseteq \ell} (-1)^{\alpha(i)+1} c_{ij}^{(\ell)} \right) (m_j^{(\ell)} - n_j^{(\ell)}) \\ &= \sum_{j=1}^{t_\ell} 2^{j-1} (m_j^{(\ell)} - n_j^{(\ell)}),\end{aligned}$$

by formula (7.25).

For $r \not\supseteq \ell$, there is some q for which $q \in D(\ell)$ and $q \notin D(r)$. Let the number corresponding set $\{q\}$ be $p = 2^q$. Then

$$\begin{aligned}&\mathbf{y}^\top \mathbf{C}^{(r)}(\mathbf{m}^{(r)} - \mathbf{n}^{(r)}) \\ &\quad = \sum_{j=1}^{t_r} \left(\sum_{i \subseteq \ell} (-1)^{\alpha(i)+1} c_{ij}^{(r)} \right) (m_j^{(r)} - n_j^{(r)}) \\ &\quad = \sum_{j=1}^{t_r} \left(\sum_{i \subseteq \ell - p} \left((-1)^{\alpha(i)+1} c_{ij}^{(r)} + (-1)^{\alpha(i+p)+1} c_{i+p\,j}^{(r)} \right) \right) (m_j^{(r)} - n_j^{(r)}),\end{aligned}$$

and since $\alpha(i+p) = \alpha(i) + 1$, moreover $q \notin r$, thus $c_{ij}^{(r)} = c_{i+p\,j}^{(r)}$ by the definition of $c_{ij}^{(r)}$. So the inner sum is 0, thus

$$\mathbf{y}^\top \mathbf{C}^{(r)}(\mathbf{m}^{(r)} - \mathbf{n}^{(r)}) = \mathbf{0}.$$

So we have

$$\begin{aligned}\mathbf{0} &= \mathbf{y}^\top \mathbf{C}(\mathbf{m} - \mathbf{n}) \\ &= \mathbf{y}^\top \mathbf{C}^{(\ell)}(\mathbf{m}^{(\ell)} - \mathbf{n}^{(\ell)}) \\ &= \sum_{j=1}^{t_\ell} 2^{j-1} (m_j^{(\ell)} - n_j^{(\ell)}),\end{aligned}$$

which means

$$\sum_{j=1}^{t_\ell} 2^{j-1} m_j^{(\ell)} = \sum_{j=1}^{t_\ell} 2^{j-1} n_j^{(\ell)}.$$

This is not possible, since

$$\mathbf{m}^{(\ell)} \in \{0,1\}^{t_\ell},$$
$$\mathbf{n}^{(\ell)} \in \{0,1\}^{t_\ell},$$
$$\mathbf{m}^{(\ell)} \neq \mathbf{n}^{(\ell)},$$

and we know, that the binary representation of integers is unique. □

7.11 Partial Activity m-out-of-t Model

In this section we present the signature coding results for the partial activity m-out-of-t model for the binary adder channel. Recall, that in this model there are t total users of the channel, out of which at most m are active at any instant. The inactive users send the zero vector from their component code, while the active ones send their other (non-zero) codeword. The received vector is the vectorial sum of the sent codewords, and from this we should recover the set of active users. We still do not show the zero codewords in the component codes, so we simply write that code $\mathcal{C}$ is

$$\mathcal{C} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(t)}\}.$$

We still use the simple notation $\mathbf{S}(U)$ for the received vector if the active users are those in set U:

$$\mathbf{S}(U) = \sum_{u\in U} \mathbf{x}^{(u)}.$$

Definition 7.5. *For the m-out-of-t model, the* ***minimal signature code length*** *$N(t,m)$ is the length of the shortest possible u.d. signature code for a given number t of total users and for a given maximal number m of simultaneously active ones:*

$$N(t,m) = \min\bigl\{n\colon \exists\mathcal{C} \text{ u.d. signature code with length } n \\ \text{for } t \text{ users out of which at most } m \text{ are active simultaneously}\bigr\}.$$

We show two bounds for $N(t,m)$ saying that for $1 \ll m \ll t$,

$$\frac{2m}{\log m}\log t \lesssim N(t,m) \lesssim \frac{4m}{\log m}\log t.$$

Based on Bose and Chowla's (1962) work, Lindström (1975) constructed a u.d. signature code with code length

$$n \sim m \log t.$$

We will show this construction soon. This is the best known so far, thus there is no asymptotically optimal code construction for the m-out-of-t model.

7.12 Bounds for U.D. Signature Codes

The next theorem gives the asymptotic upper bound on $N(t, m)$. This follows from a similar theorem of D'yachkov and Rykov in (1981), see the remark after Theorem 7.10 and 7.11. We present here a more simple proof.

Theorem 7.7. *(D'yachkov–Rykov (1981)) For $N(t, m)$ we have that*

$$\limsup_{m\to\infty}\limsup_{t\to\infty}\frac{N(t,m)\log m}{m\log t}\le 4.$$

Proof. Let the length of codewords be $n+1$, let the last component of each codeword be fixed to 1, and the rest of the components be randomly chosen from $\mathcal{B}=\{0,1\}$ with uniform distribution:

$$\begin{aligned}&\mathbf{P}\left(x_k^{(u)}=0\right)=\mathbf{P}\left(x_k^{(u)}=1\right)=\frac{1}{2} && \forall u\in\mathcal{U}, \forall k\colon 1\le k\le n \text{ (i.i.d.)},\\ &x_{n+1}^{(u)}=1 && \forall u\in\mathcal{U}.\end{aligned}$$

We will give an upper bound on the probability of the event

$$\text{code } \mathcal{C}=\left\{\mathbf{x}^{(1)},\mathbf{x}^{(2)},\ldots,\mathbf{x}^{(t)}\right\} \text{ is } \textit{not} \text{ a u.d. signature code} \tag{*}$$

and then show that for a given m and t, and for an n great enough this bound is less than 1. So the probability of randomly selecting a good code is definitely positive, and this means that there exists a good code for that n great enough, so we have an upper bound on the length of the shortest possible code.

A code is *not* a signature code, if and only if there are two different subsets U and $V\subseteq[t]$ which contain at most m users, and the sums of the corresponding code vectors are the same:

$$\mathbf{P}\big(\text{event }(*)\big)=\mathbf{P}\left(\bigcup_{\substack{U\ne V\subseteq[t]:\\ |U|\le m,|V|\le m}}\{\mathbf{S}(U)=\mathbf{S}(V)\}\right).$$

If there are two subsets U and V which satisfy $\mathbf{S}(U)=\mathbf{S}(V)$, then there are also two disjoint subsets which satisfy it (e.g. $U\setminus V$ and $V\setminus U$). Moreover, the $(n+1)^{\text{th}}$ component is 1 in all codewords, so the last component of the received vector is the size of the active set. Thus, if $\mathbf{S}(U)=\mathbf{S}(V)$ then $|U|=|V|$, so it is enough to take into account disjoint subsets of equal size:

$$\mathbf{P}\big(\text{event }(*)\big)=\mathbf{P}\left(\bigcup_{\substack{U\ne V\subseteq[t]:\\ |U|=|V|\le m,U\cap V=\emptyset}}\{\mathbf{S}(U)=\mathbf{S}(V)\}\right).$$

Now we can calculate the upper bound with the so called union bounding:

$$\begin{aligned}\mathbf{P}\big(\text{event }(*)\big) =& \sum_{\substack{U,V\subseteq[t]:\\|U|=|V|\leq m, U\cap V=\emptyset}} \mathbf{P}\left(\big\{\mathbf{S}(U)=\mathbf{S}(V)\big\}\right)\\ \leq& \sum_{k=1}^{m} \sum_{\substack{U,V\subseteq[t]:\\|U|=|V|=k, U\cap V=\emptyset}} \mathbf{P}\big(\mathbf{S}(U)=\mathbf{S}(V)\big).\end{aligned}$$

Here $\mathbf{P}\big(\mathbf{S}(U)=S(V)\big)=Q^n(k)$, where $Q(k)$ is defined by (7.5) and bounded by (7.6):

$$\begin{aligned}\mathbf{P}\big(\text{event }(*)\big) \leq& \sum_{k=1}^{m} \sum_{\substack{U,V\subseteq[t]:\\|U|=|V|=k, U\cap V=\emptyset}} Q^n(k)\\ \leq& \sum_{k=1}^{m} \binom{t}{k}\binom{t-k}{k}\left(\frac{1}{\sqrt{\pi k}}\right)^n\\ \leq& \sum_{k=1}^{m} t^{2k}\left(\frac{1}{\sqrt{\pi k}}\right)^n\\ \leq& m \max_{k:\ 1\leq k\leq m} t^{2k}\left(\frac{1}{\sqrt{\pi k}}\right)^n\\ =& m\exp\left(\max_{k:\ 1\leq k\leq m}\left(2k\log t-\frac{n}{2}\log\pi k\right)\right).\end{aligned}$$

The exponent is convex in k, so the maximum is either at $k=1$ or at $k=m$:

$$\mathbf{P}\big(\text{event }(*)\big) \leq m\exp\left(\max\left\{2\log t-\frac{n}{2}\log\pi, 2m\log t-\frac{n}{2}\log\pi m\right\}\right).$$

If we want to ensure that a u.d. code exists, it is enough to show that the probability of randomly selecting a non-u.d. code is less than 1, namely $\mathbf{P}\left(\text{event }(*)\right) < 1$. This is surely satisfied if our upper bound tends to 0 as $t\to\infty$. For this we require

$$\lim_{t\to\infty} 2\log t-\frac{n}{2}\log\pi=-\infty, \tag{7.28}$$

and

$$\lim_{t\to\infty} 2m\log t-\frac{n}{2}\log\pi m=-\infty. \tag{7.29}$$

Let us set

$$n=\left\lceil\frac{4m}{\log m}\log t\right\rceil. \tag{7.30}$$

Then (7.28) holds for all $m \geq 2$:

$$\lim_{t\to\infty} 2\log t - \frac{n}{2}\log\pi \leq 2\left(1 - \frac{m\log\pi}{\log m}\right)\lim_{t\to\infty}\log t = -\infty,$$

and (7.29) also holds:

$$\begin{aligned}\lim_{t\to\infty} 2m\log t - \frac{n}{2}\log\pi m &\leq 2m\left(1 - \frac{\log\pi m}{\log m}\right)\lim_{t\to\infty}\log t\\ &= -2\frac{m\log\pi}{\log m}\lim_{t\to\infty}\log t\\ &= -\infty.\end{aligned}$$

So if we choose n as given in (7.30) then a u.d. signature code of length $n+1$ will exist, so the length of the shortest possible u.d. signature code is bounded upper for large t:

$$N(t,m) \leq \left\lceil \frac{4m}{\log m}\log t \right\rceil + 1.$$

It follows, that

$$\limsup_{t\to\infty} \frac{N(t,m)\log m}{m\log t} \leq 4.$$

□

We show in the next theorem, that asymptotically for $1 \ll m \ll t$ we have that $N(t,m) \gtrsim \frac{2m}{\log m}\log t$. This is new, but closely relates to Theorem 7.10 of D'yachkov and Rykov.

Theorem 7.8. *For $N(t,m)$ we have that*

$$\liminf_{m\to\infty}\liminf_{t\to\infty} \frac{N(t,m)\log m}{m\log t} \geq 2.$$

Proof. Take an arbitrary u.d. signature code of length n for t users out of which at most m are active, and let U (the set of active users) be a discrete random variable with uniform distribution over the $\binom{t}{m}$ m-sized subsets of $[t]$:

$$\mathbf{P}(U = A) = \begin{cases}\binom{t}{m}^{-1} & \text{if } A \subseteq [t] \text{ and } |A| = m;\\ 0 & \text{otherwise.}\end{cases}$$

We will bound the entropy of $\mathbf{S}(U)$ in two different ways, to get an upper and a lower bound. Then by joining these bounds, we will get a lower bound on the code length.

First we set the lower bound on $\mathbf{H}\left(\mathbf{S}(U)\right)$, which is the Shannon–entropy of the random variable $\mathbf{S}(U)$:

$$\mathbf{H}\left(\mathbf{S}(U)\right) = \mathbf{H}\left(U\right),$$

since for all different values of U the corresponding $\mathbf{S}(U)$ values are different for a u.d. signature code. But U has uniform distribution, so for it is entropy

$$\mathbf{H}\left(U\right) = \log\binom{t}{m},$$

and now we are ready to derive the lower bound on $\mathbf{H}\left(\mathbf{S}(U)\right)$:

$$\begin{aligned}\mathbf{h}(\mathbf{S}(U)) &= \mathbf{H}\left(U\right)\\ &= \log\binom{t}{m}\\ &= \log\frac{t(t-1)\cdots(t-(m-1))}{m(m-1)\cdots 1}\\ &\geq m\log\frac{t}{m}.\end{aligned} \tag{7.31}$$

Now we will derive an upper bound, via bounding the entropy of the individual components of $\mathbf{S}(U)$. We can easily get the distribution of the i^{th} component, if we introduce w_i, which is the number of codewords having 1 in their i^{th} component:

$$w_i = \left|\left\{\mathbf{x}^{(u)} \colon x_i^{(u)} = 1\right\}\right|.$$

The number of all possible values of U is $\binom{t}{m}$. Moreover, the number of those U values for which $\left[\mathbf{S}(U)\right]_i = k$ can be enumerated. First we select k users out of the w_i ones those have 1 in the i^{th} component of their codeword. Then we select $m-k$ more out of the $t-w_i$ ones those have 0 there. This is $\binom{w_i}{k}\binom{t-w_i}{m-k}$, if $\max\{0, m-(t-w_i)\} \leq k \leq \min\{w_i, m\}$. So the distribution of $\left[\mathbf{S}(U)\right]_i$ is hypergeometrical, with parameters $(m, w_i, t-w_i)$:

$$\mathbf{P}\big(S(U)_i = k\big) = \begin{cases}\frac{\binom{w_i}{k}\binom{t-w_i}{m-k}}{\binom{t}{m}} & \text{if } \max\{0, m-(t-w_i)\} \leq k \leq \min\{w_i, m\};\\ 0 & \text{otherwise.}\end{cases}$$

If we introduce $\mathbf{H}_{\text{hyp}}(m, a, b)$ which is the entropy of the hypergeometrical distribution with parameters (m, a, b), then we have that

$$\mathbf{H}\big(S(U)_i\big) = \mathbf{H}_{\text{hyp}}(m, w_i, t-w_i).$$

Since the entropy of a vector can be bounded by the sum of the entropies of its components, we get the following upper bound on $\mathbf{H}\left(\mathbf{S}(U)\right)$:

$$\begin{aligned}\mathbf{H}\left(\mathbf{S}(U)\right) \leq &\sum_{i=1}^{n} \mathbf{H}\big(S(U)_i\big)\\ = &\sum_{i=1}^{n} \mathbf{H}_{\mathrm{hyp}}(m, w_i, t-w_i)\\ \leq &n \max_{w} \mathbf{H}_{\mathrm{hyp}}(m, w, t-w),\end{aligned}$$

and using Lemma 7.2 we get

$$\mathbf{H}\left(\mathbf{S}(U)\right) \leq n \max_{w} \frac{1}{2} \log\left(2\pi e\left(\mathrm{Var}_{\mathrm{hyp}}(m, w, t-w) + \frac{1}{12}\right)\right),$$

where $\mathrm{Var}_{\mathrm{hyp}}(m, w, t-w)$ denotes the variance of the hypergeometrical distribution with parameters $(m, w, t-w)$. Therefore

$$\begin{aligned}\mathbf{H}\left(\mathbf{S}(U)\right) \leq &n \max_{w} \frac{1}{2} \log\left(2\pi e\left(m\frac{w}{t}\left(1-\frac{w}{t}\right)\left(1-\frac{m-1}{t-1}\right) + \frac{1}{12}\right)\right)\\ \leq &n\frac{1}{2} \log\left(\frac{1}{2}\pi e\left(m\left(1-\frac{m-1}{t-1}\right) + \frac{1}{12}\right)\right). \qquad (7.32)\end{aligned}$$

Combining (7.31) and (7.32) we get

$$m \log\frac{t}{m} \leq \mathbf{H}\left(\mathbf{S}(U)\right) \leq n\frac{1}{2} \log\left(\frac{1}{2}\pi e\left(m\left(1-\frac{m-1}{t-1}\right) + \frac{1}{12}\right)\right),$$

which holds for all u.d. signature codes, including the shortest possible one. So

$$N(t, m) \geq \frac{m \log\frac{t}{m}}{\frac{1}{2} \log\left(\frac{1}{2}\pi e\left(m\left(1-\frac{m-1}{t-1}\right) + \frac{1}{12}\right)\right)},$$

form which

$$\liminf_{m\to\infty} \liminf_{t\to\infty} \frac{N(t, m) \log m}{m \log t} \geq 2.$$

□

7.13 Lindström's signature code construction

In this section we will show a construction of Lindström (1975) for a signature code for the m-out-of-t multiple access binary adder channel. The construction will have a code length $n(t,m)$ for which

$$\lim_{m\to\infty}\lim_{t\to\infty}\frac{n(t,m)}{m\log t}\leq 1.$$

The construction works only when the number of users is a prime power, so first we take $s \geq t$ which is a prime power, construct a code for s users, and then simply leave $s-t$ codewords in order to obtain a code for t users. Thus we can use the upper bound $s \leq 2^{\lceil \log t\rceil}$.

The construction is the following: Take the elements $\beta_1, \beta_2, \ldots, \beta_s$ of $\mathbf{GF}(s)$ and take the primitive element α of $\mathbf{GF}(s^m)$. Let $x_1, x_2, \ldots, x_s$ be integers such that

$$\alpha^{x_i} = \alpha + \beta_i \qquad \forall i = 1, 2, \ldots, s.$$

Such x_i exists for all i, since α is a primitve element of $\mathbf{GF}(s^m)$ and clearly $\alpha + \beta_i$ is a nonzero element. It also follows, that $0 < x_i < s^m - 1$, where the left inequality follows from the fact, that $\alpha + \beta_i$ cannot be $1 \in \mathbf{GF}(s^m)$.

The numbers $x_1, x_2, \ldots, x_s$ has the important property, that different sums of exactly m of them cannot coincide:

$$\sum_{i=1}^{m} x_{a_i} = \sum_{i=1}^{m} x_{b_i} \iff a_i = b_i \text{ for all } i = 1, \ldots, m.$$

This is because the simple fact, that if the sums are equal, then

$$(\alpha + \beta_{a_1})(\alpha + \beta_{a_2})\cdots(\alpha + \beta_{a_m}) = (\alpha + \beta_{b_1})(\alpha + \beta_{b_2})\cdots(\alpha + \beta_{b_m}),$$

where we can cancel α^m on both sides, and zero one side to get a polynomial of degree at most $m-1$ over $\mathbf{GF}(s)$, which is satisfied by α. Note, that the polynomial cannot be the constant zero, since the root sets of the original left and right hand sides in the equation differ. Moreover, since α is a primitive element of $\mathbf{GF}(s^m)$, it cannot satisfy such a polynomial.

Let the codewords $\mathbf{x}_i$ be the binary representations of the numbers x_i with one additional digit 1 appended. For binary representation of numbers up to ℓ we need $\lceil \log(\ell+1)\rceil$ bits. So based on $x_i < s^m - 1$ and $s \leq 2^{\lceil \log t\rceil}$ we have

$$\begin{aligned} n(t,m) &= \lceil \log(s^m - 1)\rceil + 1 \\ &\leq m\log s + 2 \\ &\leq m\lceil \log t\rceil + 2 \\ &\leq (1 + \log t)m + 2. \end{aligned}$$

If U denotes the active subset of the users, then clearly for the output vector of the channel $\mathbf{S}(U) = \sum_{u \in U} \mathbf{x}_u$. From $\mathbf{S}(U)$ we can simply calculate $|U|$ and $S(U) = \sum_{u \in U} x_u$:

$$|U| = [\mathbf{S}(U)]_{n(t,m)},$$

$$S(U) = \sum_{u \in U} x_u = \sum_{i=1}^{n(t,m)-1} 2^{n(t,m)-i-1} [\mathbf{S}(U)]_i.$$

To finally see, that this code is u.d. we shall see, that for different U belongs different $\mathbf{S}(U)$. First, if U and U' differs in size, then the last element of $\mathbf{S}(U)$ will also differ. If $|U| = |U'|$, then take $m - |U|$ arbitrary codewords. Consider U and the new codewords as one set of numbers $a_1, a_2, \ldots, a_m$ (with possible repetitions) and U' and the new codewords as the other set $b_1, b_2, \ldots, b_m$ (again with possible repetitions). The property of the codewords shown above ensures that we will have different $\mathbf{S}(U)$.

7.14 Signature Coding and Information Transfer

Here we deal with the coding problem of the multiple-access adder channel, considering both the identification of the set of active users and decoding of their messages. We examine the bounds on the minimal length of codes solving these two tasks simultaneously.

The coding problem can be formulated in the following way: there are t users of the channel: $\mathcal{U} = \{1, 2, \ldots, t\}$. Each user u has a component code, which is formed by s binary codewords of length n: $\mathbf{x}^{(u,1)}, \mathbf{x}^{(u,2)}, \ldots, \mathbf{x}^{(u,s)} \in \{0,1\}^n$, each codeword is associated with a specific message of the user. At a given instant, there are some (say r) active users. They are denoted by the set U. Enumerate them as $U = \{u_1, u_2, \ldots, u_r\}$ where $u_1 < u_2 < \ldots < u_r$. We consider, that at any time at most m users are active, so $r \leq m$. For each active user $u_i \in U$, let $m_i \in \{1, 2, \ldots, s\}$ denote the message this user wants to send. Form a vector of length r from the messages as $\mathbf{m} = \big(m_1, m_2, \ldots, m_r\big)$. The pair $(U, \mathbf{m})$, which is the set of active users and the vector of their messages together, is called a message constellation.

The active users send their corresponding codeword to the channel: user u_i sends $\mathbf{x}^{(u_i,m_i)}$. The receiver gets the sum of the codewords sent, which is denoted by $\mathbf{S}(U, \mathbf{m})$:

$$\mathbf{S}(U, \mathbf{m}) = \sum_{i=1}^{r} \mathbf{x}^{(u_i,m_i)}.$$

If the code $\mathcal{C}$ is such that for each different pair $(U, \mathbf{m})$, the channel output is different, we say this code is a uniquely decipherable (u.d.) code. Formally,

$$\mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n}) \iff (U = V \text{ and } \mathbf{m} = \mathbf{n}) \qquad \forall U, V, \mathbf{m}, \mathbf{n} \colon |U| \leq m, |V| \leq m.$$

This definition simply formalizes the fact, that if we want to recover the active users and their messages from the received vector, then we cannot have coincidence for different constellations.

We will show the following theorem:

Theorem 7.9.

$$2 \leq \liminf_{m\to\infty} \liminf_{ts\to\infty} \frac{N(t,m,s)\log m}{m \log ts} \leq \limsup_{m\to\infty} \limsup_{ts\to\infty} \frac{N(t,m,s)\log m}{m \log ts} \leq 4.$$

Proof. For the upper bound, the proof is by random coding. We give each user s codewords, each of length $n+1$. The codewords are composed of n random components independently and uniformly distributed over $\{0,1\}$, and an $(n+1)^{\text{th}}$ component fixed to 1:

$$\mathbf{P}\left(x_j^{(u,i)} = 0\right) = \mathbf{P}\left(x_j^{(u,i)} = 1\right) = \frac{1}{2} \qquad \forall j = 1, 2, \ldots, n,$$

and $x_{n+1}^{(u,i)} = 1$ for all message $i = 1, 2, \ldots, s$ of all user $u = 1, 2, \ldots, t$, where $x_j^{(u,i)}$ denotes the j^{th} component of the codeword $\mathbf{x}^{(u,i)}$.

This is not a u.d. code, if and only if there are two different constellations $(U, \mathbf{m})$ and $(V, \mathbf{n})$, for which the corresponding sum vectors are the same. So if we define "event $*$" as "code $\mathcal{C}$ is *not* a u.d. code", then we can write

$$\mathbf{P}\,(\text{event } *) = \mathbf{P}\big\{\exists (U, \mathbf{m}), (V, \mathbf{n}) \colon |U| \leq m, |V| \leq m, \mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})\big\}.$$

If $\mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})$, so the sum vector is the same for two constellations, then these constellations must have the same number of active users: $|U| = |V|$. (This is because the $(n+1)^{\text{th}}$ component of the codewords is fixed to 1, so this component of the sum vector is the number of active users.) Thus

$$\mathbf{P}\,(\text{event } *) = \mathbf{P}\big\{\exists (U, \mathbf{m}), (V, \mathbf{n}) \colon |U| = |V| \leq m, \mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})\big\}.$$

In the followings, it will be convenient to have only different vectors added in $\mathbf{S}(U, \mathbf{m})$ and $\mathbf{S}(V, \mathbf{n})$, but certainly, the same vector can occur in both terms, if there is a common user in U and V, who is sending the same message in $\mathbf{m}$ and $\mathbf{n}$. We call such a user an invariant user. If $(U, \mathbf{m})$ and $(V, \mathbf{n})$ does not have any invariant users, then we will denote it by $(U, \mathbf{n}) \perp (V, \mathbf{m})$, because it will turn out to be some kind of independence.

Fortunately, to test whether a code is u.d. or not, it is enough to consider only the constellations without invariant users: if there exists a constellation-pair $(U, \mathbf{m}), (V, \mathbf{n})$ with $\mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})$, then there also exists at least one pair $(\hat{U}, \hat{\mathbf{m}}), (\hat{V}, \hat{\mathbf{n}})$ for which $\mathbf{S}(\hat{U}, \hat{\mathbf{m}}) = \mathbf{S}(\hat{V}, \hat{\mathbf{n}})$ also, moreover it contains no invariant users: $(\hat{U}, \hat{\mathbf{n}}) \perp (\hat{V}, \hat{\mathbf{m}})$. To find this constellation-pair, simply leave each invariant user from the original constellations.

Thus we have

$$\begin{aligned}\mathbf{P}\,(\text{event } *) = \mathbf{P}\big(\exists (U, \mathbf{m}), (V, \mathbf{n})\colon\ & |U| = |V| \leq m,\\ & (U, \mathbf{m}) \perp (V, \mathbf{n}), \mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})\big),\end{aligned}$$

and by union bounding, we get

$$\mathbf{P}\,(\text{event } *) \leq \sum_{\substack{\forall (U,\mathbf{m}),(V,\mathbf{n})\colon \\ |U|=|V|\leq m, \\ (U,\mathbf{m})\perp(V,\mathbf{n})}} \mathbf{P}\big(\mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})\big). \tag{7.33}$$

Let us denote the j^{th} component of the sum vector $\mathbf{S}(U, \mathbf{m})$ by $S_j(U, \mathbf{m})$. Since the components of the codewords are independent, it follows, that the components of the sum vector are also independent. Moreover, the last component of the sum vector equals to $|U|$. Then for $|U| = |V|$,

$$\mathbf{P}\big(\mathbf{S}(U, \mathbf{m}) = \mathbf{S}(V, \mathbf{n})\big) = \prod_{j=1}^{n} \mathbf{P}\big(S_j(U, \mathbf{m}) = S_j(V, \mathbf{n})\big),$$

where

$$\mathbf{P}\big(S_j(U, \mathbf{m}) = S_j(V, \mathbf{n})\big)$$

$$
\begin{aligned}
&= \mathbf{P}\left(\sum_{i=1}^{|U|} x_j^{(u_i,m_i)} = \sum_{i=1}^{|V|} x_j^{(v_i,n_i)}\right)\\
&= \mathbf{P}\left(\sum_{i=1}^{|U|} x_j^{(u_i,m_i)} - \sum_{i=1}^{|V|} x_j^{(v_i,n_i)} = 0\right)\\
&= \mathbf{P}\left(\sum_{i=1}^{|U|} x_j^{(u_i,m_i)} + \sum_{i=1}^{|V|} \left(1 - x_j^{(v_i,n_i)}\right) = |V|\right).
\end{aligned}
$$

Because the constellations are without invariant users, we know that no common codewords appear in the sums. The j^{th} component of the codewords are independent random variables uniformly distributed over $\{0,1\}$. Thus $1-x_j^{(v_i,n_i)}$ has the same distribution as $x_j^{(v_i,n_i)}$. Moreover since the constellations are without invariant users, each codeword appears at most once in the sum, so the sum is the sum of $|U|+|V|$ independent uniformly distributed random variables. This gives a binomial distribution with parameters $\left(|U|+|V|, \frac{1}{2}\right)$, so

$$\mathbf{P}\big(S_j(U,\mathbf{m}) = S_j(V,\mathbf{n})\big) = \binom{|U|+|V|}{|V|} 2^{-|U|-|V|}.$$

Gallager ((1968) Problem 5.8 pp. 530) gives bounds on the binomial coefficients. It shows, that

$$\binom{2r}{r} 2^{-2r} \leq \sqrt{\frac{1}{\pi r}}.$$

Using this, and that $|U| = |V|$, we get

$$\mathbf{P}\big(S_j(U,\mathbf{m}) = S_j(V,\mathbf{n})\big) \leq \sqrt{\frac{1}{\pi |U|}},$$

and then

$$\mathbf{P}\big(\mathbf{S}(U,\mathbf{m}) = \mathbf{S}(V,\mathbf{n})\big) \leq \left(\sqrt{\frac{1}{\pi |U|}}\right)^n.$$

Substituting this result into (7.33) yields

$$\mathbf{P}\left(\text{event } *\right) \leq \sum_{\substack{\forall (U,\mathbf{m}),(V,\mathbf{n}):\\ |U|=|V|\leq m,\\ (U,\mathbf{m})\perp(V,\mathbf{n})}} \left(\sqrt{\frac{1}{\pi |U|}}\right)^n$$

$$\leq \sum_{\substack{\forall (U,\mathbf{m}),(V,\mathbf{n}): \\ |U|=|V|\leq m}} \left(\sqrt{\frac{1}{\pi\,|U|}}\right)^n .$$

By simple enumeration we get

$$\mathbf{P}\,(\text{event } *) \leq \sum_{r=1}^{m} \binom{t}{r}\binom{t}{r} s^{2r} \left(\sqrt{\frac{1}{\pi r}}\right)^n ,$$

where we enumerated the suitable constellation pairs based on $r = |U| = |V|$.

We use $\binom{t}{r} \leq t^r$ and take the logarithm of base 2:

$$\mathbf{P}\,(\text{event } *) \leq \sum_{r=1}^{m} 2^{2r\log ts - \frac{n}{2}\log \pi r}.$$

We bound the sum with m times the maximal element. The exponent to be maximized is convex in r, so the maximum is either at $r = 1$ or at $r = m$:

$$\mathbf{P}\,(\text{event } *) \leq m \max\left\{2^{2\log ts - \frac{n}{2}\log \pi}, 2^{2m\log ts - \frac{n}{2}\log(\pi m)}\right\}.$$

In both formulas, we substitute $n = \lceil c(m)\log tk \rceil$, and get the following bounds for $\mathbf{P}\,(\text{event } *)$:

$$m\,(ts)^{2-\frac{c(m)\log\pi}{2}} \qquad \text{(case } r = 1\text{);}$$
$$m\,(ts)^{2m-\frac{c(m)\log\pi m}{2}} \qquad \text{(case } r = m\text{).}$$

Both formulas tend to 0 as $ts \to \infty$ if for some $\varepsilon > 0$

$$c(m) > \frac{4+\varepsilon}{\log\pi} \qquad \text{(case } r = 1\text{);}$$
$$c(m) > \frac{4m+\varepsilon}{\log\pi m} \qquad \text{(case } r = m\text{).}$$

Setting $c(m) = \frac{4m+1}{\log\pi m}$ satisfies both conditions ($m \geq 1$), so for

$$n = \left\lceil \frac{4m+1}{\log\pi m}\log ts \right\rceil ,$$

we have

$$\lim_{ts\to\infty} \mathbf{P}\,(\text{event } *) = 0.$$

This means, that the probability of the complementer event is greater than 0 for some large enough ts. So the random code we select is a u.d. code with positive probability, thus at least one u.d. code exists. This means, that the value we derived for n is an upper bound for the minimal code length $N(t, m, s)$. For any $\varepsilon > 0$ and large enough ts we have

$$N(t, m, s) \leq n = \left\lceil \frac{4m + 1}{\log \pi m} \log ts \right\rceil ,$$

from which

$$\limsup_{m \to \infty} \limsup_{ts \to \infty} \frac{N(t, m, s) \log m}{m \log ts} \leq 4.$$

For the proof of the lower bound, consider an arbitrary u.d. code of length n, and select constellations at random: let $U = \big(u_1, u_2, \ldots, u_m\big)$ be uniformly distributed over the m sized subsets of $\mathcal{U}$, and let $\mathbf{m}$ be uniformly distributed over the vectors $\{1, 2, \ldots, s\}^m$, independently of U. Since the code is u.d., for each different constellation $(U, \mathbf{m})$, we get a different sum vector $\mathbf{S}(U, \mathbf{m})$, thus for the Shannon-entropy,

$$\mathbf{H}\big(\mathbf{S}(U, \mathbf{m})\big) = \mathbf{H}\big(U, \mathbf{m}\big) = \log \left(\binom{t}{m} s^m \right) \geq m \log \frac{ts}{m}, \tag{7.34}$$

where we used that $\binom{t}{m} \geq \left(\frac{t}{m}\right)^m$.

On the other side, using the standard bound on the entropy (c.f. (1991)), we can bound the entropy of the individual components of $\mathbf{S}(U, \mathbf{m})$:

$$\mathbf{H}\big(\mathbf{S}(U, \mathbf{m})\big) \leq \sum_{j=1}^{n} \mathbf{H}\big(S_j(U, \mathbf{m})\big), \tag{7.35}$$

and

$$\mathbf{H}\big(S_j(U, \mathbf{m})\big) \leq \frac{1}{2} \log \left(2\pi e \left(\mathrm{Var}\big(S_j(U, \mathbf{m})\big) + \frac{1}{12} \right) \right). \tag{7.36}$$

Here the variance $\mathrm{Var}\big(S_j(U, \mathbf{m})\big)$ can be bounded in the following way:

$$\begin{aligned}
&\mathrm{Var}\big(S_j(U, \mathbf{m})\big) \\
&\qquad = \mathbf{E}\left(\big(S_j(U, \mathbf{m})\big)^2 \right) - \left(\mathbf{E}\big(S_j(U, \mathbf{m})\big) \right)^2 \\
&\qquad = \mathbf{E}\left(\left(\sum_{i=1}^{m} x_j^{(u_i, m_i)} \right)^2 \right) - \left(\mathbf{E}\left(\sum_{i=1}^{m} x_j^{(u_i, m_i)} \right) \right)^2 \\
&\qquad = \mathbf{E}\left(\sum_{i=1}^{m} \left(x_j^{(u_i, m_i)} \right)^2 \right)
\end{aligned}$$

$$+\mathbf{E}\left(\sum_{k=1}^{m}\sum_{\substack{\ell=1\\ \ell\neq k}}^{m} x_j^{(u_k,m_k)} x_j^{(u_\ell,m_\ell)}\right)$$

$$-\left(\mathbf{E}\left(\sum_{i=1}^{m} x_j^{(u_i,m_i)}\right)\right)^2. \tag{7.37}$$

Here the first and the third expected values are easy to calculate. Let $p_j^{(u)} = \frac{1}{s}\sum_{i=1}^{s} x_j^{(u,i)}$, and $\bar{p}_j = \frac{1}{t}\sum_{u=1}^{t} p_j^{(u)}$. Then since $x_j^{(u_i,m_i)} \in \{0,1\}$,

$$\mathbf{E}\left(\sum_{i=1}^{m}\left(x_j^{(u_i,m_i)}\right)^2\right) = \mathbf{E}\left(\sum_{i=1}^{m} x_j^{(u_i,m_i)}\right) = m\bar{p}_j,$$

and

$$\left(\mathbf{E}\left(\sum_{i=1}^{m} x_j^{(u_i,m_i)}\right)\right)^2 = m^2\bar{p}_j^2.$$

For the second expected value,

$$\mathbf{E}\left(\sum_{k=1}^{m}\sum_{\substack{\ell=1\\ \ell\neq k}}^{m} x_j^{(u_r,m_r)} x_j^{(u_\ell,m_\ell)}\right) =$$

$$\mathbf{E}\left(\sum_{k=1}^{m}\sum_{\substack{\ell=1\\ \ell\neq k}}^{m} \mathbf{E}\Big(x_j^{(u_k,m_k)}\Big|U\Big)\mathbf{E}\Big(x_j^{(u_\ell,m_\ell)}\Big|U\Big)\right),$$

since for a given U, $x_j^{(u_k,m_k)}$ and $x_j^{(u_\ell,m_\ell)}$ are conditionally independent ($k \neq \ell$), and calculating the conditional expected value yields

$$\mathbf{E}\left(\sum_{k=1}^{m}\sum_{\substack{\ell=1\\ \ell\neq k}}^{m} x_j^{(u_k,m_k)} x_j^{(u_\ell,m_\ell)}\right) = \mathbf{E}\left(\sum_{k=1}^{m}\sum_{\substack{\ell=1\\ \ell\neq k}}^{m} p_j^{(u_k)} p_j^{(u_\ell)}\right)$$

$$= \frac{\binom{t-2}{m-2}}{\binom{t}{m}} \sum_{k=1}^{t}\sum_{\substack{\ell=1\\ \ell\neq k}}^{t} p_j^{(k)} p_j^{(\ell)}.$$

Thus for $\operatorname{Var}\bigl(S_j(U,\mathbf{m})\bigr)$ we have obtained (c.f. (7.37))

$$\begin{aligned}
\operatorname{Var}\bigl(S_j(U,\mathbf{m})\bigr) &= m\bar{p}_j + \frac{\binom{t-2}{m-2}}{\binom{t}{m}} \sum_{k=1}^{t} \sum_{\substack{\ell=1\\ \ell\neq k}}^{t} p_j^{(k)} p_j^{(\ell)} - m^2\bar{p}_j^2 \\
&\le m\bar{p}_j + \frac{m(m-1)}{t(t-1)} \left(\sum_{i=1}^{t} p_j^{(i)} \right)^2 - m^2\bar{p}_j^2 \\
&= m\bar{p}_j + \frac{m(m-1)}{t(t-1)} t^2 \bar{p}_j^2 - m^2\bar{p}_j^2 \\
&= m\bar{p}_j + \frac{m}{t-1}\bigl((m-1)t - m(t-1)\bigr)\bar{p}_j^2 \\
&= m\bar{p}_j + \frac{m(m-t)}{t-1}\bar{p}_j^2 \\
&\le m,
\end{aligned}$$

since $m \le t$ and $\bar{p}_j \le 1$.

Returning to (7.35) and (7.36) and using that $m + \frac{1}{12} < 2m$, we get

$$\begin{aligned}
\mathbf{H}\bigl(\mathbf{S}(U,\mathbf{m})\bigr) &\le \sum_{j=1}^{n} \mathbf{H}\bigl(S_j(U,\mathbf{m})\bigr) \\
&< \sum_{j=1}^{n} \frac{1}{2}\log(4\pi e m) \\
&= \frac{n}{2}\log(4\pi e m).
\end{aligned}$$

Comparing this with (7.34) yields

$$n \ge \frac{m \log \frac{ts}{m}}{\frac{1}{2}\log(4\pi e m)}.$$

This holds for any u.d. code, even for the shortest possible one, with length $n = N(t,m,s)$. So

$$\liminf_{m\to\infty} \liminf_{ts\to\infty} \frac{N(t,m,s)\log m}{m \log ts} \ge 2.$$

This finishes the proof of Theorem 7.9. □

7.15 Bounds for B_m Codes

D'yachkov and Rykov (1981) considered a special class of u.d. signature codes, namely the B_m codes.

Definition 7.6. *A* $\boldsymbol{B_m}$ ***code*** *is a set of t binary codewords of length n*

$$\mathcal{C} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}\right\} \subseteq \mathcal{B}^n$$

which has the following property: all sums of exactly m (not definitely different) codewords are different.

It is obvious, that a B_m code $\mathcal{C}$ can be converted into a u.d. signature code $\mathcal{C}_s$ for the adder channel. What we have to do, is just to append a fixed 1 bit to the end of all the codewords in the B_m code:

$$\mathcal{C}_s = \left\{\mathbf{y} \colon \mathbf{y} \in \mathcal{B}^{n+1}, \exists \mathbf{x} \in \mathcal{C} \colon \forall j \in [n] \colon y_j = x_j \text{ and } y_{n+1} = 1\right\}.$$

The length of this signature code is $n+1$. To see that this is really a u.d. signature code, we indirectly put up that there are two different subsets U and V of the users for which the sum vector is the same. The size of U and V cannot differ, since then the $(n+1)^{\text{th}}$ component of the sum vectors would also differ. So we can assume that $|U| = |V|$. Now take the following (exactly) m codewords: all the codewords in U plus $\mathbf{x}^{(1)}$ as many times as needed to get exactly m codewords ($m - |U|$ times). The sum vector of this multiset must be equal to the sum vector of all the codewords in V plus $\mathbf{x}^{(1)}$ $m - |V|$ times. But then we have found two multisets of codewords with m elements in the original B_m code $\mathcal{C}$, for which the sum vector is the same. This is a contradiction with the definition of the B_m codes.

D'yachkov and Rykov (1981) have given upper and lower bounds on $N_B(t, m)$, which is the length of the shortest possible B_m code for t total users out of which at most m are active simultaneously. For the length of the shortest possible signature code, they have proven the following two theorems:

Theorem 7.10. *(Dyachkov–Rykov (1981)) For any $m < t$*

$$N_B(t, m) \geq \frac{\log \dfrac{t^m}{m!}}{\mathrm{H}_{\text{bin}}(m, \frac{1}{2})}.$$

Theorem 7.11. *(Dyachkov–Rykov (1981)) If $t \to \infty$ then for any fixed m*

$$N_B(t, m) \leq \frac{2m}{\log \dfrac{2^{2m}}{\binom{2m}{m}}}(1 + o(1)) \log t.$$

Asymptotically for $1 \ll m \ll t$, these theorems say that

$$\frac{2m}{\log m}\log t \lesssim N_B(t,m) \lesssim \frac{4m}{\log m}\log t. \tag{7.38}$$

Using the construction of signature codes from B_m codes, it is trivial, that $N(t,m) \le N_B(t,m)+1$, thus the asymptotic lower bound follows from Theorem 7.8. For the asymtotic upper bound, we show a proof here using the technique of the Pippenger Theorem (Theorem 7.3).

Proof of the asymptotics of Theorem 7.11. Select a random code $\mathcal{C}$ of length $n+1$ for t users: the first n bits are randomly selected while the $(n+1)^{\text{th}}$ is fixed to one. (See the proof of the Pippenger Theorem 7.3 for the exact distribution.)

We will show, that the probability of the event, that

$$\text{code } \mathcal{C} \text{ is not a } B_m \text{ code} \tag{*}$$

is less than one. We need a code with property

$$\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v}) \iff \mathbf{u} = \mathbf{v}$$
$$\forall \mathbf{u},\mathbf{v} \in \{1,2,\dots,m\}^t\colon \sum_{i=1}^t u_i \le m, \sum_{i=1}^t v_i \le m.$$

(Here we assumed only that the number of sent codewords is less than or equals to m, while the definition of B_m codes says that it is exactly m. So we are now proving a somewhat stronger statement. Via proving in this stronger form, we will have an alternative proof of Theorem 7.7, which then can be seen as a corollary of this.)

$$\mathbf{P}\big(\text{event } (*)\big) = \mathbf{P}\left(\bigcup_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\dots,m\}^t\colon\\ \sum_{i=1}^t u_i \le m, \sum_{i=1}^t v_i \le m}} \big\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\big\}\right)$$
$$= \mathbf{P}\left(\bigcup_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\dots,m\}^t\colon\\ \mathbf{u}^\top\mathbf{v}=0, \sum_{i=1}^t u_i = \sum_{i=1}^t v_i \le m}} \big\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\big\}\right),$$

where we used that because the last fixed 1 bit, the number of codewords must be the same for both constellations, and that if there is a vector pair

$\mathbf{u}, \mathbf{v}$ which satisfy $\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})$, then there is also an orthogonal vector pair satisfying it. (See the proof of the Pippenger Theorem 7.3 for explanation.)

Now applying the union bound one gets

$$\mathbf{P}\big(\text{event } (*)\big) \le \sum_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\ldots,m\}^t:\\ \mathbf{u}^\top\mathbf{v}=0, \sum_{i=1}^t u_i = \sum_{i=1}^t v_i \le m}} \mathbf{P}\left(\big\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\big\}\right). \tag{7.39}$$

For a given $\mathbf{u}$ and $\mathbf{v}$ we can bound $\mathbf{P}\left(\big\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\big\}\right)$ using (7.17) from the proof of the Pippenger Theorem (Theorem 7.3). Let $\ell = |U| + |V|$, where $U = \big\{i \in \{1, 2, \ldots, t\} \colon u_i \ne 0\big\}$ and $V = \big\{i \in \{1, 2, \ldots, t\} \colon v_i \ne 0\big\}$. Then

$$\mathbf{P}\left(\big\{\mathbf{S}(\mathbf{u}) = \mathbf{S}(\mathbf{v})\big\}\right) \le 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}.$$

Returning to (7.39), we get

$$\begin{aligned}\mathbf{P}\big(\text{event } (*)\big) &\le \sum_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\ldots,m\}^t:\\ \mathbf{u}^\top\mathbf{v}=0, \sum_{i=1}^t u_i = \sum_{i=1}^t v_i \le m}} 2^{-\frac{n}{2}\log\frac{\pi|U\cup V|}{2}} \\ &= \sum_{\ell=1}^{m} \sum_{u,v\colon u+v=\ell} \sum_{\substack{U,V\subseteq\{1,2,\ldots,t\}:\\ |U|=u, |V|=v,\\ U\cap V=\emptyset}} \sum_{\substack{\mathbf{u},\mathbf{v}\in\{1,2,\ldots,m\}^t:\\ \sum_{i=1}^t u_i = \sum_{i=1}^t v_i \le m\\ \{i\colon u_i\ne 0\}=U\\ \{i\colon v_i\ne 0\}=V}} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}},\end{aligned}$$

where we enumerated the possible vectors $\mathbf{u}$ and $\mathbf{v}$ based on the size of their base set U and V. Thus

$$\begin{aligned}\mathbf{P}\big(\text{event } (*)\big) &\le \sum_{\ell=1}^{2m} \sum_{u,v\colon u+v=\ell} \binom{t}{u}\binom{t-u}{v}\binom{m}{u}\binom{m}{v} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}} \\ &\le \sum_{\ell=1}^{2m} \binom{t}{\ell} 2^\ell \sum_{u,v\colon u+v=\ell} \binom{m}{u}\binom{m}{v} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}} \\ &= \sum_{\ell=1}^{2m} \binom{t}{\ell} 2^\ell \binom{2m}{\ell} 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}}. \\ &\le \sum_{\ell=1}^{2m} (4tm)^\ell 2^{-\frac{n}{2}\log\frac{\pi\ell}{2}},\end{aligned}$$

where in the last step we bounded $\binom{t}{\ell}$ with t^ℓ and $\binom{2m}{\ell}$ with $(2m)^\ell$. Bounding with the maximal element we get

$$\mathbf{P}\big(\text{event } (*)\big) \le 2m 2^{\max_{\ell\in\{1,2,\ldots,2m\}}\left\{\ell\log 4tm - \frac{n}{2}\log\frac{\pi\ell}{2}\right\}}.$$

Since the exponent to be maximized is convex in ℓ, the maximum is either at $\ell = 1$ or at $\ell = 2m$. For the first case we get

$$\mathbf{P}\big(\text{event } (*)\big) \leq 2m 2^{\log 4tm - \frac{n}{2}\log\frac{\pi}{2}},$$

while for the second

$$\mathbf{P}\big(\text{event } (*)\big) \leq 2m 2^{2m\log 4tm - \frac{n}{2}\log \pi m}.$$

Set $n = \left\lceil \frac{cm\log t}{\log m} \right\rceil$, and examine the bounds as $t \to \infty$:

$$\begin{aligned}\lim_{t\to\infty} 2m2^{\log 4tm - \frac{n}{2}\log\frac{\pi}{2}} &\leq \lim_{t\to\infty} 2m2^{\log 4tm - \frac{cm\log t}{2\log m}\log\frac{\pi}{2}} \\ &= 2m2^{\left(1 - \frac{cm}{2\log m}\log\frac{\pi}{2}\right)\lim_{t\to\infty}\log t}.\end{aligned}$$

This is 0, if $c > \frac{2\log m}{m\log\frac{\pi}{2}}$.

For the second bound,

$$\begin{aligned}\lim_{t\to\infty} 2m2^{2m\log 4tm - \frac{n}{2}\log\pi m} &\leq \lim_{t\to\infty} 2m2^{2m\log 4tm - \frac{cm\log t}{2\log m}\log\pi m} \\ &\leq 2m2^{\left(2m - \frac{cm}{2\log m}\log\pi m\right)\lim_{t\to\infty}\log t}.\end{aligned}$$

This is also 0, if $c > \frac{4\log m}{\log \pi m}$. Let us set $c = 4+\varepsilon$ for some $\varepsilon > 0$. This satisfies both bounds on c for m large enough. Thus the random code of length

$$\left\lceil \frac{(4+\varepsilon)m\log t}{\log m} \right\rceil + 1$$

we select, is not a B_m code with probability less than one, so at least one B_m code of that length must exist. So for the length $N_B(t,m)$ of the shortest possible B_m code we have

$$N_B(t,m) \leq \left\lceil \frac{(4+\varepsilon)m\log t}{\log m} \right\rceil + 1,$$

from which

$$\limsup_{m\to\infty}\limsup_{t\to\infty} \frac{N_B(t,m)\log m}{m\log t} \leq 4.$$

□

Chapter 8

Collision Channel with Known Collision Multiplicity

8.1 Channel Model

Similarly to Chapter 6, in this chapter we consider the slotted multiple-access collision channel with feedback. Again, suppose that there are infinitely many non cooperating users and that the packet arrivals can be modelled as a Poisson process with intensity λ.

Here the feedback at the time slot $[n, n+1)$ is:

- feedback 0 means an idle slot,
- feedback 1 means successful transmission by a single user,
- feedback c with $c \geq 2$ means that collision happened such that c users sent packet. c is called the multiplicity of the collision.

8.2 Pippenger's Random Protocol of Throughput 1

This protocol is based on the Pippenger's Theorem (cf. Theorem 7.3), which says that for any integer t there is a code

$$C = C_t = \{\mathbf{c}_1, \ldots, \mathbf{c}_t\}$$

of code word length

$$n = \frac{at}{\log t}$$

such that the following holds: The n-tuple

$$\mathbf{y} = \sum_{i=1}^{t} u_i \mathbf{c}_i$$

uniquely determines the nonnegative integers $u_1, u_2, \ldots, u_t \in \{0, 1, \ldots, t\}$ satisfying

$$\sum_{i=1}^{t} u_i \leq t.$$

Similarly to Chapter 6, the protocol consists of *collision resolution intervals* (CRI) which contain a certain number of slots as follows. All channel users are assumed to follow what is going on on the channel, so everybody not involved in a certain initial collision has to wait until the end of the epoch before transmitting his information packet (blocked access). All users can find out the end of the CRI, because the protocol is known to everybody. Therefore, it is sufficient to describe the protocol for one epoch.

1. In the initial time slot of a CRI, each active user sends his packet. After this, all users know the number of transmitted packets, i.e., the number say s of active users, because that number is the feedback they receive (the multiplicity of the collisions). If this number is 0 or 1, the CRI ends.

2. Choose an integer $t = t_s$. Suppose there are $s \geq 2$ active users, each of which selects, randomly and uniformly, a number from $\{1, 2, \ldots, t\}$. These numbers $i_1, \ldots, i_s$ are called the identifiers.

3. Having the code C_{t_s} with the property above, each user j $(1 \leq j \leq s)$ resends his packet according to the the code word $\mathbf{c}_{i_j}$. This means that the packet will be resent in slots pointed out by $1's$ in the code word $\mathbf{c}_{i_j}$.

4. Let $u_i := \#\{j : i_j = i\}$ be the number of active users which chose the identifier i and $\mathbf{u} = (u_1, u_2, \ldots, u_t)$. According to step 3 (and since feedback is the multiplicity of the collisions) the feedback sequence will be

$$\mathbf{y} = \sum_{i=1}^{t} u_i \mathbf{c}_i.$$

Now each user determines **u** from **y**, which is possible by the above-mentioned Pippenger's Theorem because clearly

$$\sum_{i=1}^{t} u_i \leq t.$$

5. Now, all users know the multiplicities u_i by which the identifiers i were chosen. In a reserved way, only the users with identifiers i with $u_i = 1$ re-send their packets.

The number of slots needed for this procedure (i.e. the length of a CRI) is

$$\ell = 1 + n + m,$$

where m is the number of successfully transmitted packets during this CRI. (In step 1 of the protocol, 1 slot was used; in step 3 this was

$$n = n_s = \frac{at_s}{\log t_s}$$

slots; and in step 5 we used m slots.)

Theorem 8.1. *(Pippenger (1981)) For the Pippenger protocol, choose $t_s = s(\log s)^b$ with $0 < b < 1$, and assume that the arrivals of the packets are according to a Poisson process $\{Z(t) : t \geq 0\}$ with intensity $\lambda > 0$. If*

$$\lambda < 1,$$

then the sequence of numbers of packets at the end of a CRI forms a stable Markov chain.

Proof. Let X_n be the number of packets at the end of the n-th CRI, Z_n the number of packets arrived in the n-th CRI, V_n the number of the successfully transmitted packets in the n-th CRI. Then

$$X_{n+1} = X_n - V_{n+1} + Z_{n+1}.$$

It is easy to see that $\{X_n\}$ is an irreducible and aperiodic Markov chain, so in order to prove its stability we have to check the Foster criteria (Theorem B.4). For any $s \geq 0$,

$$\begin{aligned} \mathbf{E}\{X_{n+1}|X_n = s\} &= \mathbf{E}\{X_n - V_{n+1} + Z_{n+1}|X_n = s\} \\ &= s - \mathbf{E}\{V_{n+1}|X_n = s\} + \mathbf{E}\{Z_{n+1}|X_n = s\}. \end{aligned}$$

Let Y_n be the length of the n-th CRI, then

$$\begin{aligned}\mathbf{E}\{Z_{n+1}|X_n=s\} = \frac{\mathbf{E}\{Z_{n+1}I_{\{X_n=s\}}\}}{\mathbf{P}\{X_n=s\}} \\ &= \frac{\mathbf{E}\{\sum_{y=1}^{\infty} Z_{n+1}I_{\{X_n=s,Y_{n+1}=y\}}\}}{\mathbf{P}\{X_n=s\}} \\ &= \frac{\sum_{y=1}^{\infty}\mathbf{E}\{Z_{n+1}|X_n=s,Y_{n+1}=y\}\mathbf{P}\{X_n=s,Y_{n+1}=y\}}{\mathbf{P}\{X_n=s\}} \\ &= \sum_{y=1}^{\infty}\mathbf{E}\{Z_{n+1}|X_n=s,Y_{n+1}=y\}\mathbf{P}\{Y_{n+1}=y|X_n=s\} \\ &= \sum_{y=1}^{\infty}\mathbf{E}\{Z_{n+1}|Y_{n+1}=y\}\mathbf{P}\{Y_{n+1}=y|X_n=s\} \\ &= \sum_{y=1}^{\infty}\lambda y\,\mathbf{P}\{Y_{n+1}=y|X_n=s\} \\ &= \lambda\,\mathbf{E}\{Y_{n+1}|X_n=s\}.\end{aligned}$$

Given $X_n = s$,

$$Y_{n+1} = 1 + n_s + V_{n+1},$$

therefore

$$\mathbf{E}\{Z_{n+1}|X_n=s\} = \lambda(1+n_s+\mathbf{E}\{V_{n+1}|X_n=s\}).$$

Thus

$$\begin{aligned}\mathbf{E}\{X_{n+1}|X_n=s\} &= s-\mathbf{E}\{V_{n+1}|X_n=s\}+\lambda(1+n_s+\mathbf{E}\{V_{n+1}|X_n=s\}) \\ &= s-(1-\lambda)\mathbf{E}\{V_{n+1}|X_n=s\}+\lambda(1+n_s).\end{aligned}$$

From the definition and the notations of the protocol we get that

$$\begin{aligned}\mathbf{E}\{V_{n+1}|X_n=s\} &= \mathbf{E}\left\{\sum_{j=1}^{t_s} I_{\{u_j=1\}}|X_n=s\right\} \\ &= \sum_{j=1}^{t_s}\mathbf{P}\{u_j=1|X_n=s\} \\ &= t_s\mathbf{P}\{u_1=1|X_n=s\} \\ &= t_s s\frac{1}{t_s}\left(1-\frac{1}{t_s}\right)^{s-1}\end{aligned}$$

$$
\begin{aligned}
&= s\left(1-\frac{1}{t_s}\right)^{s-1}\\
&\geq s\left(1-\frac{s-1}{t_s}\right).
\end{aligned}
$$

Because of the choice of t_s,

$$n_s = \frac{at_s}{\log t_s} = \frac{as(\log s)^b}{\log s + b\log\log s}, = s\cdot o(1)$$

and so

$$
\begin{aligned}
\mathbf{E}\{X_{n+1}|X_n = s\} &\leq s - s(1-\lambda)\left(1-\frac{s-1}{t_s}\right) + 2n_s\\
&= s - s\left((1-\lambda)\left(1-\frac{s-1}{s(\log s)^b}\right) + o(1)\right)\\
&= s - s(1-\lambda)(1+o(1)).
\end{aligned}
$$

For any fixed $d > 0$, one can choose I such that for $s > I$

$$s(1-\lambda)(1+o(1)) > d,$$

therefore

$$\mathbf{E}\{X_{n+1}|X_n = s\} \leq s - d,$$

and we verified the conditions of Theorem B.4. □

8.3 Ruszinkó-Vanroose Conflict Resolution Protocol of Throughput 1

The only probabilistic argument in the proof of Pippenger's protocol (Theorem 8.1) is the construction of a code C based on Pippenger's Theorem 7.3. Thus, if someone could constructively generate such a code then Pippenger's probabilistic protocol would become a constructive protocol of throughput 1. Unfortunately, it is still an *open problem* how to generate such codes. However, note that code C in the protocol can be seen as a *parallel* or *non-adaptive* strategy for determining $\mathbf{u}$. In this section, due to Ruszinkó and Vanroose (1997), we give an *adaptive* search strategy instead.

Stated in terms of search theory, the code C of Pippenger's Theorem 7.3 is essentially a *parallel* solution to the following combinatorial search problem, in $n = O(t/\log t)$ steps:

Problem 8.1. *Let $u_1, \dots, u_t$ be a sequence of non negative integers with $\sum_{i=1}^{t} u_i \leq t$. We are allowed to ask queries of type How much is*

$$z(A) = \sum_{i \in A} u_i,$$

for any subset A of $\{1, \dots, t\}$. Find the sequence $u_1, \dots, u_t$ with as few queries as possible.

This is because of the following. Since C is binary, the coordinate y_i $(i = 1, \dots, n)$ of the vector $\mathbf{y} = \sum_{j=1}^{t} u_j \mathbf{c}_j$ is the sum of a certain subset of coordinates of $\mathbf{u} = (u_1, \dots, u_t)$ pointed out by the 1s in the i^{th} components of the code words in C. Thus we can consider the vector $\mathbf{y} = (y_1, \dots, y_n)$ as a sequence of answers for queries of the type above. Let A_i be the set

$$A_i = \{j; c_{j,i} = 1\}$$

then

$$y_i = z(A_i).$$

Since in this sense, we ask for each i $(i = 1, \dots, n)$ the i^{th} question according to the i^{th} components of the code words in C and it does not depend on previous answers, it is a *parallel* strategy. The random coding argument in the proof of Pippenger's Theorem thus shows the existence of a *parallel search strategy* solving Problem 8.1 in $n = O(t/\log t)$ steps. But according to the channel model all users get feedback immediately, thus *it suffices to have an adaptive search strategy which solves Problem 8.1 in $o(t)$ steps.*

These observations lead us to the following *constructive* solution of the communication problem.

First we intend to solve the search Problem 8.1 in $o(t)$ *adaptive* steps. For this, we will use an extended version of Lindström's construction (cf. section 7.10).

In 7.10, we have shown the construction of a uniquely decodable signature code for the t-user multiple access adder channel. The t-user binary adder channel has t binary inputs, and one output, which is the real sum of the inputs. In the case of signature coding, each user can send his codeword or the zero vector into the channel. The u. d. code is such that, we can determine the set of those users sending their codeword.

Now, in the extended construction we consider a t-user channel with t integer inputs and one integer output, which is the sum of the inputs. We enable each user to send his codeword multiplied by an arbitrary integer between 0 and k. The task is to recover this multiplier for each user.

The formalization of the extended problem is as follows: A code for t users is a set $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ of codewords. The message constellation is a vector of length t over the nonnegative integers: $\mathbf{m} = (m_1, m_2, \ldots, m_t)^\top$. The i^{th} integer m_i shows that at a given instant, which multiple of $\mathbf{x}_i$ does the i^{th} user send. Let us denote the output in the case of constellation $\mathbf{m}$ with $\mathbf{S}(\mathbf{m})$, which is a vector of length n over the set of nonnegative integers:

$$\mathbf{S}(\mathbf{m}) = \sum_{i=1}^{t} m_i \mathbf{x}_i. \tag{8.1}$$

Definition 8.1. *A code is called u.d. if for all the input constellations where each user sends his codeword with multiplicity of at most $k-1$, the channel output is unique.*

$$\forall \mathbf{m} \in \{0, 1, \ldots, k-1\}^t, \forall \mathbf{n} \in \{0, 1, \ldots, k-1\}^t, \mathbf{m} \neq \mathbf{n} \colon \mathbf{S}(\mathbf{m}) \neq \mathbf{S}(\mathbf{n}). \tag{8.2}$$

Such class of codes will be costructed for a certain subseries of t. In the sequel, we will refer to these codes as *Lindström codes.* Lindström has shown the following theorem:

Theorem 8.2. *(Lindström, (1964)) One can construct a u.d. code of the above type for t users with code length $n(t)$, for which*

$$\limsup_{t \to \infty} \frac{n(t) \log t}{t \log k} \leq 2.$$

Proof. In the followings, to construct the code, we will do almost the same as in section 7.10. We will only give the differences here. We will use the notations (e.g. $a \subseteq b$) on the binary forms given there.

For a given n, for each $r \in \{1, 2, \ldots, n\}$ construct the matrix $\mathbf{C}^{(r)}$ of size $r \times \alpha(r)$ in the following way: for row i where $i \subseteq r$ select j^{th} element $c_{ij}^{(r)} \in \mathcal{B}$ where

$$j \in \left\{1, 2, \ldots, \left\lfloor \frac{\alpha(r) - 1}{\log k} \right\rfloor + 1 \right\}$$

such that

$$\sum_{i \subseteq r} (-1)^{\alpha(i)+1} c_{ij}^{(r)} = k^{j-1}. \tag{8.3}$$

This is possible, for the same reasons as given in section 7.10, because here $2^{\alpha(r)-1}$ is an upper bound for k^{j-1}, since $j - 1 \leq \frac{\alpha(r)-1}{\log k}$. For the other rows, the construction is the same.

Create the code matrix in the same way. Since **C** still have n rows, the codewords are of length n. To calculate the number of columns, which is the number of users in the code, do as the following:

$$t(n) = \sum_{r=1}^{n} t_r = \sum_{r=1}^{n} \left\lfloor \frac{\alpha(r) - 1}{\log k} \right\rfloor + 1 \geq \sum_{r=1}^{n} \frac{\alpha(r) - 1}{\log k} \geq \frac{\sum_{r=1}^{n} \alpha(r) - n}{\log k}. \quad (8.4)$$

We have proven in section 7.10 that

$$\sum_{r=1}^{n} \alpha(r) \sim \frac{n \log n}{2}.$$

Thus continuing (8.4), we get that this construction yields a maximal number of $t(n)$ users for a given n for which

$$t(n) \gtrsim \frac{n \log n}{2 \log k},$$

or precisely

$$\liminf_{n \to \infty} \frac{t(n) \log k}{n \log n} \geq \frac{1}{2}. \quad (8.5)$$

To this point we have a code construction of an arbitrary length n, for $t(n)$ users. Now we will invert the relation, and express the minimal code length $n(t)$ achievable with this construction for a given number t of users. More precisely we will give an upper bound on this $n(t)$.

$$n(t) \lesssim \frac{2t \log k}{\log t},$$

or precisely

$$\limsup_{t \to \infty} \frac{n(t) \log t}{t \log k} \leq 2.$$

To see this, in the exact same way as we did in section 7.10, we can set an arbitrary $\varepsilon > 0$, and show that

$$n'(t) = \frac{2(1+\varepsilon) t \log k}{\log t}$$

is an upper bound for $n(t)$ if t is great enough. For the detailed proof refer to section 7.10.

Since $\varepsilon > 0$ is arbitrary, this yields

$$\limsup_{t \to \infty} \frac{n(t) \log t}{t \log k} \leq 2.$$

What remains to see, is that the above constructed $\mathbf{C}$ is really a u.d. code matrix. Again, refer to section 7.10 with the following modifications:

For $r \supsetneq \ell$, and for $r \not\supseteq \ell$ the sum is zero, as there.

For $r = \ell$,

$$\begin{aligned}\mathbf{y}^\top \mathbf{C}^{(\ell)}(\mathbf{m}^{(\ell)} - \mathbf{n}^{(\ell)}) &= \sum_{j=1}^{t_\ell} \left(\sum_{i \subseteq \ell} (-1)^{\alpha(i)+1} c_{ij}^{(\ell)} \right) (m_j^{(\ell)} - n_j^{(\ell)}) \\ &= \sum_{j=1}^{t_\ell} k^{j-1} (m_j^{(\ell)} - n_j^{(\ell)}),\end{aligned}$$

by formula (8.3).

So we have

$$\begin{aligned}\mathbf{0} &= \mathbf{y}^\top \mathbf{C}(\mathbf{m} - \mathbf{n}) \\ &= \mathbf{y}^\top \mathbf{C}^{(\ell)}(\mathbf{m}^{(\ell)} - \mathbf{n}^{(\ell)}) \\ &= \sum_{j=1}^{t_\ell} k^{j-1} (m_j^{(\ell)} - n_j^{(\ell)}),\end{aligned}$$

which means

$$\sum_{j=1}^{t_\ell} k^{j-1} m_j^{(\ell)} = \sum_{j=1}^{t_\ell} k^{j-1} n_j^{(\ell)}.$$

This is not possible, since

$$\begin{aligned}\mathbf{m}^{(\ell)} &\in \{0, 1, \ldots, k-1\}^{t_\ell}, \\ \mathbf{n}^{(\ell)} &\in \{0, 1, \ldots, k-1\}^{t_\ell}, \\ \mathbf{m}^{(\ell)} &\neq \mathbf{n}^{(\ell)},\end{aligned}$$

and we know, that the representation of integers in radix k is unique. □

Unfortunately this result of Lindström cannot be applied directly to our problem, since in Problem 8.1 a single coordinate can be large, if the other coordinates are zeroes it can be even t. Thus if we replace k by t into the result of Lindström mentioned above, n will not be of magnitude $o(t)$. But we can do the following.

Construction: An adaptive algorithm which solves Problem 8.1 in

$$O\left(t \, \frac{\log \log t}{\log t} \right)$$

steps:

1. First partition the coordinates of $\mathbf{u} = (u_1, \ldots, u_t]$ into $t/\log t$ parts of size $\log t$, for example, let T_i $(1 \leq i \leq t/\log t)$ contain coordinates u_j for which
$$(i-1)\log t \leq j \leq i \log t$$
holds.

2. Ask for the total sum of the coordinates in each part T_i $(1 \leq i \leq t/\log t)$.

3. Now according to the answers we split the partition into two classes. Let class $\mathcal{C}_1$ contain the partition elements where the sum of the coordinates exceeds $(\log t)^2$, and let $\mathcal{C}_2$ contain the other partition elements.

4. Ask one-by-one each coordinate u_i contained in parts $T_j \in \mathcal{C}_1$.

5. Determine coordinates u_i contained in parts $T_j \in \mathcal{C}_2$ using a Lindström code C with $k = (\log t)^2$. (By step 3 all such coordinates should be $\leq (\log t)^2$.)

By this procedure we solve Problem 8.1 in
$$O\left(t \, \frac{\log\log t}{\log t}\right)$$
constructive steps:

We ask questions only in steps 2, 4 and 5. Step 2 obviously needs $t/\log t$ queries, as well as step 4. In the latter case it follows from the assumption $\sum_{i=1}^{t} u_i \leq t$ that the number of T_j-s contained in $\mathcal{C}_1$ is
$$\leq t/(\log t)^2,$$
and each part contains $\log t$ coordinates. In step 5, by Lindström's construction we have to ask
$$O\left(t \, \frac{\log\log t}{\log t}\right)$$
queries. Thus the total number of queries is also of magnitude
$$O\left(t \, \frac{\log\log t}{\log t}\right).$$

Now we are ready to present a conflict resolution protocol of throughput 1 which uses almost the same steps as the non-constructive proof of Pippenger presented in the previous section.

This protocol also consists of CRIs which contain a certain number of slots. All channel users are assumed to follow what is going on on the channel, so everybody not involved in a certain initial collision has to wait until the end of the CRI before transmitting his information packet. All users can find out the end of the CRI, because the protocol is known to everybody. Therefore, it is sufficient to describe the protocol for one CRI. We do this in five consecutive steps, where for each step j we calculate the number ℓ_j of time slots needed:

1. In the initial time slot, each active user sends his packet. Thus,

   $$\ell_1 = 1,$$

   and after this, all users know the number of transmitted packets, i.e., the number say s of active users, because that number is the feedback they receive (the multiplicity of the collisions). If this number is 0 or 1, the CRI ends.

2. Choose an integer $t = t_s$. Suppose there are $s \geq 2$ active users, each of which selects, randomly and uniformly, a number from $\{1, 2, \ldots, t\}$. These numbers $i_1, \ldots, i_s$ are called the identifiers. Group the identifiers in intervals of length $\log t$. I.e., for $1 \leq d \leq t/\log t$ let

   $$L_d := \{i_j : (d-1)\log t < i_j \leq d \log t\}.$$

   In the d^{th} time slot, users with an identifier in L_d re-send their packet. Thus,

   $$\ell_2 = \frac{t}{\log t}.$$

   (In this section, we avoid the use of "$\lceil . \rceil$" and "$\lfloor . \rfloor$" notations because only magnitudes of the expressions are important.)

3. Consider first the groups L_d for which the feedback in the d-th slot exceeds $(\log t)^2$. There are certainly less than $t/(\log t)^2$ such groups. In a reserved way all users with an identifier belonging to these groups re-send their packet. After that, these users leave the system. So,

   $$\ell_3 \leq \frac{t}{(\log t)^2} \log t = \frac{t}{\log t}.$$

4. Consider now the groups L_d for which the feedback in the d-th slot in step 2 does not exceed $(\log t)^2$. Let $u_i := \#\{j : i_j = i\}$ be the number

of still active users which chose the identifier i. Then $u_i \leq (\log t)^2$, so construct a Lindström code C with $k = (\log t)^2$, and determine $\mathbf{u}$ from $\mathbf{y} = \sum_{i=1}^{t} u_i \mathbf{c}_i$, where $\mathbf{y}$ is the feedback sequence of length

$$\ell_4 \leq O\left(\frac{t}{\log_k t}\right) = O\left(t\,\frac{\log\log t}{\log t}\right).$$

5. Now, all users know the multiplicities u_i by which the identifiers i were chosen. In a reserved way, only the users with identifiers i with $u_i = 1$ re-send their packets:

   $$\ell_5 \leq m,$$

   where m is the number of successfully transmitted packets during this epoch. The users which left the system in step 3 with a collision, and the users in step 5 with $u_i > 1$ must try again in the next epoch.

Thus exactly m packets are successfully transmitted in $\ell = \ell_1 + \cdots + \ell_5 \leq 1 + n + m$ time slots, where

$$n = O\left(t\,\frac{\log\log t}{\log t}\right). \tag{8.6}$$

Now we are ready to prove that our algorithm is of throughput 1:

Theorem 8.3. *(Ruszinkó and Vanroose (1997)) For the Pippenger protocol, assume the conditions of Theorem 8.1. If*

$$\lambda < 1,$$

then the sequence of numbers of packets at the end of a CRI forms a stable Markov chain.

Proof. The proof is almost the same as that of Theorem 8.1. The only slight difference is the end, where because of (8.6)

$$n = n_s = O\left(t_s\,\frac{\log\log t_s}{\log t_s}\right) = s \cdot O\left(\frac{(\log s)^b \log\log s}{\log s}\right) = s \cdot o(1).$$

□

Chapter 9

Euclidean Channel

9.1 Channel Model

The Euclidean channel is a special case of the multiple-access channel. This channel is an adder channel for real numbers. The channel input and output alphabets are the set $\mathbb{R}$ of real numbers, and the output is simply the sum of the inputs:

$$Y = \sum_{i=1}^{t} X_i.$$

This channel is much like the binary adder channel, the difference is the channel input and output alphabet, which is the set $\mathbb{R}$ of real numbers in the case of this Euclidean channel instead of the set $\{0, 1\}$ and the set $\mathbb{N}$ which was the input and output alphabet of the binary adder channel, respectively.

We will discuss multiple-access codes for this channel. For simplicity, we use signature codes, where as usual, each user (each component code) has only two elements, and one of these is the all zero. The non-zero one is denoted by $\mathbf{x}^{(i)}$ for the ith user. Let us call users sending their all zero codeword inactive, and users sending their nonzero one as active.

Since the channel is synchronized and deterministic, the channel output is simply the sum of the codewords of the active users. If we denote the set of active users by U, then the channel output is

$$\mathbf{y}_U = \sum_{i \in U} \mathbf{x}^{(i)}.$$

Certainly, if we do not have any noise, then the channel capacity is infinite. To better model real transmissions, we introduce minimal distance and maximal energy constraints. This yields the definition of Euclidean signature codes:

Definition 9.1. $\mathcal{C} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(t)}\right\}$ *is an Euclidean signature code with length n for t total and m maximally active users if*

$$\mathbf{x}^{(i)} \in \mathbb{R}^n, \ \left\|\mathbf{x}^{(i)}\right\| \leq 1 \qquad \forall i \in [t],$$

and

$$d_{min}(\mathcal{C}) \geq d,$$

where

$$d_{min}(\mathcal{C}) = \min_{\substack{U \subseteq [t], V \subseteq [t]: \\ |U| \leq m, |V| \leq m, U \neq V}} \|\mathbf{y}_U - \mathbf{y}_V\|,$$

and $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^n$.

The reason for these constraints is straightforward. Consider some disturbing noise in the communication. The minimum distance criteria makes it possible to recover the messages of the users from the noisy output with a certain fidelity. Certainly, if the codewords are from $\mathbb{R}^n$, the minimum-distance criteria makes no sense without a maximal energy constraint.

Furthermore, we assume, that only a small subset of the users are communicating simultaneously. We will use the m-out-of-t model, where there are t total users out of which at most m are active at any given instant.

For given values of t, m and d, we define the minimal Euclidean signature code length $N_E(t, m, d)$ as the length of the shortest possible Euclidean signature code with this given parameters:

$$N_E(t, m, d) = \min\left\{n \in \mathbb{N} \colon \exists \mathcal{C}(n, t, m) \text{ Euclidean code with } d_{\min}(\mathcal{C}) \geq d\right\}. \tag{9.1}$$

Since for $U = \{1\}$ and $V = \emptyset$ the distance $\|\mathbf{y}_U - \mathbf{y}_V\| = \left\|\mathbf{x}^{(1)}\right\| \leq 1$, so there are no codes with minimum distance $d_{\min} > 1$, therefore we only consider $N_E(t, m, d)$ for $0 < d \leq 1$.

9.2 Bounds for Euclidean Signature Codes

It is easy to see, that for the minimal Euclidean signature code length defined by (9.1)

$$\liminf_{m \to \infty} \liminf_{t \to \infty} \frac{N_E(t, m, d) \log m}{m \log t} \geq 1.$$

To see this, there is a rather simple sphere packing bound. Consider the space $\mathbb{R}^n$, and put a sphere with radius $\frac{d}{2}$ centered at each possible received vector. These spheres must be disjoint, since each two received vector has at least distance d.

The number of spheres is

$$\sum_{i=0}^{m} \binom{t}{i},$$

while the volume of one sphere is

$$\left(\frac{d}{2}\right)^n c_n,$$

where c_n is the volume of the sphere with unit radius. So the total occupied volume $\mathcal{V}$ is

$$\mathcal{V} = \sum_{i=0}^{m} \binom{t}{i} \left(\frac{d}{2}\right)^n c_n. \tag{9.2}$$

On the other hand, for any signature code for at most m simultaneously active users the length of the received vector $\mathbf{v}$ is bounded by m, so the previous spheres are located inside a sphere of radius $m + \frac{d}{2}$. So for the total occupied volume $\mathcal{V}$,

$$\mathcal{V} \leq \left(m + \frac{d}{2}\right)^n c_n. \tag{9.3}$$

Putting (9.2) and (9.3) together we get

$$\sum_{i=0}^{m} \binom{t}{i} \left(\frac{d}{2}\right)^n c_n \leq \left(m + \frac{d}{2}\right)^n c_n,$$

and then

$$\binom{t}{m} \left(\frac{d}{2}\right)^n \leq \left(m + \frac{d}{2}\right)^n.$$

It implies that

$$\left(\frac{t}{m}\right)^m \left(\frac{d}{2}\right)^n \leq \left(m + \frac{d}{2}\right)^n,$$

therefore

$$m(\log t - \log m) + n(\log d - \log 2) \leq n \log\left(m + \frac{d}{2}\right),$$

and

$$\frac{n \log m}{m \log t} \geq \frac{\log m \left(1 - \frac{\log m}{\log t}\right)}{\log(2m+d) - \log d}.$$

This also holds for the shortest possible code with length $N_E(t,m,d)$:

$$\frac{N_E(t,m,d) \log m}{m \log t} \geq \frac{\log m \left(1 - \frac{\log m}{\log t}\right)}{\log(2m+d) - \log d},$$

therefore

$$\liminf_{t\to\infty} \frac{N_E(t,m,d) \log m}{m \log t} \geq \frac{\log m}{\log(2m+d) - \log d},$$

which implies that

$$\liminf_{m\to\infty} \liminf_{t\to\infty} \frac{N_E(t,m,d) \log m}{m \log t} \geq 1.$$

This was a simple sphere packing argument: if $\mathcal{C}$ is a Euclidean signature code, then all vectors $\mathbf{y}_U$ $(U \subseteq [t], |U| \leq m)$ are within a ball of radius m. The main idea of the next improved lower bound is to show that for any Euclidean signature code, half of the vectors $\mathbf{y}_U$ $(U \subseteq [t], |U| = m)$ are within a ball of radius of $2\sqrt{m}$. From this, the second lower bound on the code length immediately follows by the same sphere packing argument applied to the sphere with radius $2\sqrt{m}$.

Theorem 9.1. *(Füredi–Ruszinkó, (1999))*

$$\liminf_{m\to\infty} \liminf_{t\to\infty} \frac{N_E(t,m,d) \log m}{m \log t} \geq 2.$$

To proove this theorem, we need the following lemma:

Lemma 9.1. *For any Euclidean signature code $\mathcal{C} = \mathcal{C}(n,t,m)$ the inequality*

$$\sum_{U\subseteq[t]:\, |U|=m} \|\mathbf{y}_U - m\mathbf{c}\|^2 \leq \binom{t}{m} m$$

holds, where $\mathbf{c} = \frac{1}{t}\sum_{i=1}^{t} \mathbf{x}^{(i)}$ is the average vector.

Proof.

$$\sum_{U\subseteq[t]\colon |U|=m} \|\mathbf{y}_U - m\mathbf{c}\|^2 = \sum_{U\subseteq[t]\colon |U|=m} \left(\|\mathbf{y}_U\|^2 - 2m\langle \mathbf{y}_U, \mathbf{c}\rangle + m^2 \|\mathbf{c}\|^2\right) \quad (9.4)$$

We can do the summation by terms. For the second term,

$$\begin{aligned}\sum_{U\subseteq[t]\colon |U|=m} -2m\langle \mathbf{y}_U, \mathbf{c}\rangle &= -2m \left\langle \sum_{U\subseteq[t]\colon |U|=m} \mathbf{y}_U, \mathbf{c} \right\rangle \\ &= -2m\binom{t-1}{m-1} t\langle \mathbf{c}, \mathbf{c}\rangle \\ &= -2\binom{t}{m} m^2 \|\mathbf{c}\|^2,\end{aligned}$$

since in the sum $\sum_{U\subseteq[t]\colon |U|=m} \mathbf{y}_U$ every vector of code $\mathcal{C}$ is summed up with multiplicity $\binom{t-1}{m-1}$. For the third term,

$$\sum_{U\subseteq[t]\colon |U|=m} m^2 \|\mathbf{c}\|^2 = \binom{t}{m} m^2 \|\mathbf{c}\|^2.$$

For the first term,

$$\begin{aligned}\sum_{U\subseteq[t]\colon |U|=m} \|\mathbf{y}_U\|^2 &= \sum_{U\subseteq[t]\colon |U|=m} \left\| \sum_{i\in U} \mathbf{x}^{(i)} \right\|^2 \\ &= \sum_{U\subseteq[t]\colon |U|=m} \left(\sum_{i\in U} \left\|\mathbf{x}^{(i)}\right\|^2 + \sum_{i,j\in U\colon i\neq j} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle \right),\end{aligned}$$

and since $||\mathbf{x}^{(i)}|| \leq 1$,

$$\sum_{U\subseteq[t]\colon |U|=m} \|\mathbf{y}_U\|^2 \leq \sum_{U\subseteq[t]\colon |U|=m} \left(m + \sum_{i,j\in U\colon i\neq j} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle \right).$$

From the fact that a pair of vectors is contained in exactly $\binom{t-2}{m-2}$ m-tuples, it follows that

$$\begin{aligned}
&\sum_{U\subseteq[t]\colon |U|=m} \left\|\mathbf{y}_U\right\|^2 \\
&\quad= \binom{t}{m}m + \binom{t-2}{m-2}\sum_{i,j\in[t]\colon i\neq j}\left\langle \mathbf{x}^{(i)},\mathbf{x}^{(j)}\right\rangle \\
&\quad\leq \binom{t}{m}m + \binom{t-2}{m-2}\sum_{i,j\in[t]\colon i\neq j}\left\langle \mathbf{x}^{(i)},\mathbf{x}^{(j)}\right\rangle + \binom{t-2}{m-2}\sum_{i\in[t]}\left\|\mathbf{x}^{(i)}\right\|^2 \\
&\quad= \binom{t}{m}m + \binom{t-2}{m-2}\left\|\sum_{i\in[t]}\mathbf{x}^{(i)}\right\|^2 \\
&\quad= \binom{t}{m}m + \binom{t-2}{m-2}t^2\left\|\mathbf{c}\right\|^2 \\
&\quad\leq \binom{t}{m}m + \binom{t}{m}m^2\left\|\mathbf{c}\right\|^2.
\end{aligned}$$

And putting the three terms in (9.4) together we get

$$\sum_{U\subseteq[t]\colon |U|=m}\left\|\mathbf{y}_U - m\mathbf{c}\right\|^2 \leq \binom{t}{m}m.$$

□

Now we are ready to prove the new upper bound on the rate of Euclidean superimposed codes.

Proof of Theorem 9.1. Take an arbitrary Euclidean superimposed code $\mathcal{C}$ for t total users out of which at most m are active. Let n denote the length of the code, and – similarly to the above lemma – let $\mathbf{c} = \frac{1}{t}\sum_{i\in[t]}\mathbf{x}^{(i)}$. Let U be a random variable with uniform distribution over the m sized subsets of $[t]$.

$$\mathbf{P}(U=V) = \frac{1}{\binom{t}{m}} \qquad \forall V\subseteq[t]\colon |V|=m.$$

By the definition of expected value,

$$\mathbf{E}||\mathbf{y}_U - m\mathbf{c}||^2 = \frac{1}{\binom{t}{m}}\sum_{V\subseteq[t]\colon |V|=m}\left\|\mathbf{y}_V - m\mathbf{c}\right\|^2,$$

and using Lemma 9.1:

$$\mathbf{E}||\mathbf{y}_U - m\mathbf{c}||^2 \leq m.$$

Jensen's inequality for the random variable $||\mathbf{y}_U - m\mathbf{c}||$ says

$$\left(\mathbf{E}\left(\|\mathbf{y}_U - m\mathbf{c}\|\right)\right)^2 \leq \mathbf{E}\left(\|\mathbf{y}_U - m\mathbf{c}\|^2\right),$$

so

$$\mathbf{E}\|\mathbf{y}_U - m\mathbf{c}\| \leq \sqrt{m}.$$

Thus by Markov's inequality,

$$\mathbf{P}\left(\|\mathbf{y}_U - m\mathbf{c}\| > 2\sqrt{m}\right) \leq \frac{\mathbf{E}\left(\|\mathbf{y}_U - m\mathbf{c}\|\right)}{2\sqrt{m}} = \frac{1}{2}.$$

This means that at least half of the m active user's sum vectors lies within an n-dimensional sphere of radius $2\sqrt{m}$.

But $\mathcal{C}$ is a Euclidean code, which means that even those received vectors within the sphere of radius $2\sqrt{m}$ must have distance at least d from each other. Applying the sphere packing argument to these vectors we get that

$$\frac{1}{2}\binom{t}{m} \leq \left(\frac{2\sqrt{m} + \frac{d}{2}}{\frac{d}{2}}\right)^n,$$

thus

$$\frac{1}{2}\left(\frac{t}{m}\right)^m \leq \left(1 + \frac{4\sqrt{m}}{d}\right)^n,$$

and by taking the logarithm,

$$n \geq \frac{\log\frac{1}{2} + m(\log t - \log m)}{\log\left(1 + \frac{4\sqrt{m}}{d}\right)},$$

and this also holds for the shortest possible Euclidean code with given parameters:

$$N_E(t,m,d) \geq \frac{\log\frac{1}{2} + m(\log t - \log m)}{\log\left(1 + \frac{4\sqrt{m}}{d}\right)},$$

thus

$$\liminf_{m\to\infty}\liminf_{t\to\infty}\frac{N_E(t,m,d)\log m}{m\log t} \geq 2.$$

□

Theorem 9.2. *(Ericson–Györfi, (1988))*

$$\limsup_{m\to\infty}\limsup_{t\to\infty}\frac{N_E(t,m,d)\log m}{m\log t}\le 4.$$

Proof. The proof is based on random coding. Choose a random code for t total and m active users with length n with the following distribution:

$$\mathbf{P}\left(\left\{\mathbf{X}_j^{(i)}=\frac{1}{\sqrt{n}}\right\}\right)=\mathbf{P}\left(\left\{\mathbf{X}_j^{(i)}=-\frac{1}{\sqrt{n}}\right\}\right)=\frac{1}{2}\qquad \forall i\in[t]\ \forall j\in[n].$$

For the probability of that this code does not have minimal distance d, we have

$$\begin{aligned}\mathbf{P}\big(d_{\min}(\mathcal{C})<d\big)&=\mathbf{P}\left(d^2_{\min}(\mathcal{C})<d^2\right)\\&=\mathbf{P}\left(\min_{(U,V)\in A_{t,m}}\|\mathbf{y}_U-\mathbf{y}_V\|^2<d^2\right),\end{aligned}$$

where

$$A_{t,m}=\big\{(U,V)\colon U\subseteq[t],V\subseteq[t],|U|\le m,|V|\le m,U\ne V\big\}.$$

Since for each pair (U,V) setting the minimum, the disjoint pair $(U\setminus U\cap V,V\setminus U\cap V)$ also sets the minimum, it is enough to take into account the disjoint sets only:

$$\mathbf{P}\big(d_{\min}(\mathcal{C})<d\big)=\mathbf{P}\left(\min_{\substack{(U,V)\in A_{t,m}\\U\cap V=\emptyset}}\|\mathbf{y}_U-\mathbf{y}_V\|^2<d^2\right),$$

and applying the union bound, we get

$$\mathbf{P}\big(d_{\min}(\mathcal{C})<d\big)\le\sum_{\substack{(U,V)\in A_{t,m}\\U\cap V=\emptyset}}\mathbf{P}\left(\|\mathbf{y}_U-\mathbf{y}_V\|^2<d^2\right).$$

Since the codewords are composed of components $\pm\frac{1}{\sqrt{n}}$, if $|U|+|V|$ is odd, then

$$\big|[\mathbf{y}_U-\mathbf{y}_V]_j\big|\ge\frac{1}{\sqrt{n}}\qquad\forall j\in[n],$$

so $\|\mathbf{y}_U-\mathbf{y}_V\|\ge 1$. Thus for $|U|+|V|$ odd for $d\le 1$, the probability

$$\mathbf{P}\left(\|\mathbf{y}_U-\mathbf{y}_V\|^2<d^2\right)=0,$$

so we do not have to sum these cases:

$$\mathbf{P}\big(d_{\min}(\mathcal{C}) < d\big) \le \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} \mathbf{P}\left(\|\mathbf{y}_U - \mathbf{y}_V\|^2 < d^2\right).$$

Moreover, for $U \cap V = \emptyset$ the distribution of $\mathbf{y}_U - \mathbf{y}_V$ and $\mathbf{y}_U + \mathbf{y}_V$ are the same, so we can use this latter in the formula:

$$\begin{aligned}\mathbf{P}\big(d_{\min}(\mathcal{C}) < d\big) &\le \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} \mathbf{P}\left(\|\mathbf{y}_U + \mathbf{y}_V\|^2 < d^2\right)\\ &= \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} \mathbf{P}\left(\left\|\sum_{i\in U\cup V} \mathbf{X}^{(i)}\right\|^2 < d^2\right)\\ &= \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} \mathbf{P}\left(n\sum_{j=1}^{n}\left(\sum_{i\in U\cup V} \mathbf{X}_j^{(i)}\right)^2 < nd^2\right),\end{aligned}$$

and by the Chernoff bounding technique,

$$\begin{aligned}&\mathbf{P}\big(d_{\min}(\mathcal{C}) < d\big)\\ &\quad= \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} \mathbf{P}\left(\exp\left(-\frac{n}{2}\sum_{j=1}^{n}\left(\sum_{i\in U\cup V} \mathbf{X}_j^{(i)}\right)^2\right) > e^{-\frac{nd^2}{2}}\right)\\ &\quad\le \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} e^{\frac{nd^2}{2}}\mathbf{E}\exp\left(-\frac{n}{2}\sum_{j=1}^{n}\left(\sum_{i\in U\cup V} \mathbf{X}_j^{(i)}\right)^2\right)\\ &\quad= \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} e^{\frac{nd^2}{2}}\prod_{j=1}^{n}\mathbf{E}\exp\left(-\frac{1}{2}\left(\sqrt{n}\sum_{i\in U\cup V} \mathbf{X}_j^{(i)}\right)^2\right)\\ &\quad= \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V| \text{ is even}}} e^{\frac{nd^2}{2}}\left(\mathbf{E}\exp\left(-\frac{1}{2}\left(\sqrt{n}\sum_{i\in U\cup V} \mathbf{X}_1^{(i)}\right)^2\right)\right)^n.\end{aligned}$$

If $|U| + |V|$ is even, then $\sqrt{n}\sum_{i\in U\cup V}\mathbf{X}_j^{(i)}$ is also even, with distribution

$$\mathbf{P}\left(\sqrt{n}\sum_{i\in U\cup V}\mathbf{X}_j^{(i)} = 2z\right) = \binom{2k}{k+z}\frac{1}{2^{2k}}$$

over $z \in \{-k, \dots, k\}$, where $2k = |U| + |V|$. Thus

$$\mathbf{E}\exp\left(-\frac{1}{2}\left(\sqrt{n}\sum_{i\in U\cup V}\mathbf{X}_j^{(i)}\right)^2\right) = \sum_{z=-k}^{k} e^{-2z^2}\binom{2k}{k+z}\frac{1}{2^{2k}}.$$

So the Chernoff-bound is

$$\mathbf{P}(d_{\min}(\mathcal{C}) < d) \le \sum_{\substack{(U,V)\in A_{t,m}\\ U\cap V=\emptyset\\ |U|+|V|=2k}} e^{\frac{nd^2}{2}}\left(\sum_{z=-k}^{k} e^{-2z^2}\binom{2k}{k+z}\frac{1}{2^{2k}}\right)^n,$$

where we enumerate the appropriate pairs (U, V) with respect to $2k = |U| + |V|$:

$$\mathbf{P}(d_{\min}(\mathcal{C}) < d) \le \sum_{k=1}^{m}\binom{t}{2k}2^{2k}e^{\frac{nd^2}{2}}\left(\sum_{z=-k}^{k} e^{-2z^2}\binom{2k}{k+z}2^{-2k}\right)^n,$$

and since $\binom{t}{2k}2^{2k} \le 2t^{2k}$ and $d \le 1$,

$$\mathbf{P}(d_{\min}(\mathcal{C}) < d) \le 2\sum_{k=1}^{m} t^{2k}e^{\frac{n}{2}}\left(\sum_{z=-k}^{k} e^{-2z^2}\binom{2k}{k+z}2^{-2k}\right)^n = 2(A+B),$$

where

$$A = t^2e^{\frac{n}{2}}\left(\sum_{z=-1}^{1} e^{-2z^2}\binom{2}{1+z}2^{-2}\right)^n,$$

and

$$B = \sum_{k=2}^{m} t^{2k}e^{\frac{n}{2}}\left(\sum_{z=-k}^{k} e^{-2z^2}\binom{2k}{k+z}2^{-2k}\right)^n.$$

We will derive upper bounds on A and B:

$$\begin{aligned} A &= \exp\left(2\log t + n\left(\frac{1}{2} + \log(1+e^{-2}) - \log 2\right)\right) \\ &\le \exp\left(2\log t - 0.066n\right). \end{aligned}$$

For B, we will use $\binom{2k}{k+z} \leq \binom{2k}{k}$ and $\binom{2k}{k}2^{-2k} \leq \frac{1}{\sqrt{\pi k}}$ (c.f.: Gallager):

$$\sum_{z=-k}^{k} e^{-2z^2} \binom{2k}{k+z} 2^{-2k} \leq \frac{1}{\sqrt{\pi k}} \sum_{z=-k}^{k} e^{-2z^2},$$

and using $\exp(-2z^2) \leq \exp(-2|z|)$, we get

$$\begin{aligned}
\sum_{z=-k}^{k} e^{-2z^2} \binom{2k}{k+z} 2^{-2k} &\leq \frac{1}{\sqrt{\pi k}} \sum_{z=-k}^{k} e^{-2|z|} \\
&= \frac{1}{\sqrt{\pi k}} \left(1 + 2\sum_{z=1}^{k} e^{-2z}\right) \\
&\leq \frac{1}{\sqrt{\pi k}} \left(1 + 2\frac{e^{-2}}{1-e^{-2}}\right) \\
&\leq \frac{0.741}{\sqrt{k}}.
\end{aligned}$$

So for B, we have

$$\begin{aligned}
B &\leq \sum_{k=2}^{m} t^{2k} e^{\frac{n}{2}} \left(\frac{0.741}{\sqrt{k}}\right)^n \\
&\leq m \max_{k=2\ldots m} \exp\left(2k\log t + n\left(0.201 - \frac{\log k}{2}\right)\right).
\end{aligned}$$

It can be easily seen, that the exponent is convex in k. So the maximum is either at $k = 2$ or at $k = m$:

$$B \leq m \max\{C, D\},$$

where

$$\begin{aligned}
C &= \exp\left(4\log t + n\left(0.201 - \frac{\log 2}{2}\right)\right) \\
&\leq \exp\left(4\log t - 0.145n\right),
\end{aligned}$$

and

$$D = \exp\left(2m\log t + n\left(0.201 - \frac{\log m}{2}\right)\right).$$

We want to show that for large values of t, the probability of that this random code does not have a certain minimal distance is less than one. We have $\mathbf{P}\left(d_{\min}(\mathcal{C}) < d\right) \leq 2(A + m\min\{B, C\})$, so it is enough to show that

$$\lim_{t\to\infty} A = 0, \lim_{t\to\infty} C = 0 \text{ and } \lim_{t\to\infty} D = 0.$$

Set $n = \lceil c(m)\log t\rceil$, then

$$\begin{aligned} A &\leq \exp\left(\left(2 - 0.066c(m)\right)\log t\right), \\ C &\leq \exp\left(\left(4 - 0.145c(m)\right)\log t\right), \end{aligned}$$

and

$$D \leq \exp\left(\left(2m - \left(\frac{\log m}{2} - 0.201\right)c(m)\right)\log t\right).$$

All these quantities A, B and C tends to 0 as $t \to \infty$ if in the exponents $\log t$ has a negative factor. We have this for A if $c(m) > 30.304$, for C if $c(m) > 27.587$, and for D if

$$2m - \left(\frac{\log m}{2} - 0.201\right)c(m) < 0.$$

All of these conditions are satisfied by

$$c(m) = \frac{4(1+\varepsilon)m}{\log m}$$

for $m \geq 25$ and $m \geq \exp\left(\frac{0.402(1+\varepsilon)}{\varepsilon}\right)$, where $\varepsilon > 0$ arbitrary.

Summarizing, we have shown that for any $\varepsilon > 0$, if

$$m > \max\left\{25, \exp\left(\frac{0.4005(1+\varepsilon)}{\varepsilon}\right)\right\},$$

the probability of a randomly selected code with length

$$n = \left\lceil \frac{4(1+\varepsilon)m}{\log m}\log t\right\rceil$$

not having minimal distance d tends to 0:

$$\lim_{t\to\infty} \mathbf{P}(d_{\min}(\mathcal{C}) < d) = 0.$$

This means that for t large enough, a good code with certain parameters exists, so for any $\varepsilon > 0$, m large enough and t large enough

$$N_E(t, m, d) < \frac{4(1+\varepsilon)m}{\log m} \log t + 1,$$

which implies that

$$\limsup_{m\to\infty} \limsup_{t\to\infty} \frac{N_E(t, m, d) \log m}{m \log t} \le 4.$$

□

9.3 Signature Coding and Information Transfer for the Euclidean Channel

Introduction

There are t users of the channel: $\mathcal{U} = \{1, 2, \ldots, t\}$. Each user u has a component code, which is formed by s real valued codewords of length n:

$$C_u = \{\mathbf{x}^{(u,1)}, \mathbf{x}^{(u,2)}, \ldots, \mathbf{x}^{(u,s)}\},$$

each codeword is associated with a specific message of the user. We have an energy constraint: $\left\|\mathbf{x}^{(u,j)}\right\| \le 1$, where $\|.\|$ denotes the Euclidean norm. At a given instant, there are some (say r) active users. They are denoted by the set U. Enumerate them as $U = \{u_1, u_2, \ldots, u_r\}$, where $u_1 < u_2 < \ldots < u_r$. We consider, that at any time at most m users are active, so $r \le m$. For each active user $u_i \in U$, let $m_i \in \{1, 2, \ldots, s\}$ denote the message this user wants to send. Form a vector of length r from the messages as $\mathbf{m} = \big(m_1, m_2, \ldots, m_r\big)$. The pair $(U, \mathbf{m})$, which is the set of active users and the vector of their messages together, is called a message constellation.

The active users send their corresponding codewords to the channel: user u_i with message m_i sends $\mathbf{x}^{(u_i,m_i)}$. The receiver gets the sum of the codewords sent, which is denoted by $\mathbf{S}(U, \mathbf{m})$:

$$\mathbf{S}(U, \mathbf{m}) = \sum_{i=1}^{r} \mathbf{x}^{(u_i,m_i)}.$$

If the code $\mathcal{C}$ is such that for each different pair $(U, \mathbf{m})$, the channel output is different at least by d in Euclidean norm, then say this code has distance d. Formally,

$$\|\mathbf{S}(U, \mathbf{m}) - \mathbf{S}(V, \mathbf{n})\| < d \iff (U = V \text{ and } \mathbf{m} = \mathbf{n})$$
$$\forall U, V, \mathbf{m}, \mathbf{n} \colon |U| \le m, |V| \le m.$$

Bounds for the code length

Given t, m, s and d, the smallest codeword length for which a d-distance s-message Euclidean code $\mathcal{C}$ for t total users out of which at most m are active exists is noted by $N(t,m,s,d)$.

Theorem 9.3.

$$\liminf_{m\to\infty}\liminf_{ts\to\infty}\frac{N(t,m,s,d)\log m}{m\log ts}\geq 2.$$

To proove this theorem, we need the following lemma:

Lemma 9.2. *(Füredi–Ruszinkó, (1999)) For any code $\mathcal{C}$ defined above, the inequality*

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}}\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|^2\leq\binom{t}{m}s^m m$$

holds, where

$$\mathbf{c}=\frac{1}{ts}\sum_{\substack{i\in\{1,2,\ldots,t\}\\ k\in\{1,2,\ldots,s\}}}\mathbf{x}^{(i,k)}$$

is the average vector.

Proof. We will denote the Euclidean inner product with $\langle .,.\rangle$.

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}}\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|^2$$

$$=\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}}\left(\|\mathbf{S}(U,\mathbf{m})\|^2-2m\left\langle\mathbf{S}(U,\mathbf{m}),\mathbf{c}\right\rangle+m^2\|\mathbf{c}\|^2\right)\quad(9.5)$$

We can do the summation by terms. For the second term,

$$\begin{aligned}\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}}-2m\left\langle\mathbf{S}(U,\mathbf{m}),\mathbf{c}\right\rangle&=-2m\left\langle\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}}\mathbf{S}(U,\mathbf{m}),\mathbf{c}\right\rangle\\&=-2m\left\langle\binom{t-1}{m-1}s^{m-1}ts\mathbf{c},\mathbf{c}\right\rangle\\&=-2\binom{t}{m}s^m m^2\|\mathbf{c}\|^2,\end{aligned}$$

since in the sum

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \mathbf{S}(U,\mathbf{m})$$

every vector of code $\mathcal{C}$ is summed up with multiplicity $\binom{t-1}{m-1}s^{m-1}$.

For the third term,

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} m^2\left\|\mathbf{c}\right\|^2=\binom{t}{m}s^m m^2\left\|\mathbf{c}\right\|^2.$$

For the first term,

$$\begin{aligned}\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\mathbf{S}(U,\mathbf{m})\right\|^2 &= \sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\sum_{i=1}^m \mathbf{x}^{(u_i,m_i)}\right\|^2\\ &= \sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left(\sum_{i=1}^m \left\|\mathbf{x}^{(u_i,m_i)}\right\|^2+\sum_{i=1}^m\sum_{\substack{j=1\\ j\neq i}}^m \left\langle \mathbf{x}^{(u_i,m_i)},\mathbf{x}^{(u_j,m_j)}\right\rangle\right),\end{aligned}$$

and since $\left\|\mathbf{x}^{(u_i,m_i)}\right\|\leq 1$,

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\mathbf{S}(U,\mathbf{m})\right\|^2 \leq \sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left(m+\sum_{i=1}^m\sum_{\substack{j=1\\ j\neq i}}^m \left\langle \mathbf{x}^{(u_i,m_i)},\mathbf{x}^{(u_j,m_j)}\right\rangle\right).$$

From the fact that a pair of vectors is contained in exactly $\binom{t-2}{m-2}s^{m-2}$ constellations, it follows that

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\mathbf{S}(U,\mathbf{m})\right\|^2$$

$$\begin{aligned}
&= \binom{t}{m} s^m m + \binom{t-2}{m-2} s^{m-2} \sum_{\substack{i\in\{1,2,\ldots,m\}\\ k\in\{1,2,\ldots,s\}}} \sum_{\substack{j\in\{1,2,\ldots,t\},j\neq i\\ \ell\in\{1,2,\ldots,s\}}} \left\langle \mathbf{x}^{(i,k)}, \mathbf{x}^{(j,\ell)} \right\rangle \\
&\le \binom{t}{m} s^m m + \binom{t-2}{m-2} s^{m-2} \sum_{\substack{i\in\{1,2,\ldots,t\}\\ k\in\{1,2,\ldots,s\}}} \sum_{\substack{j\in\{1,2,\ldots,t\}\\ \ell\in\{1,2,\ldots,s\}}} \left\langle \mathbf{x}^{(i,k)}, \mathbf{x}^{(j,\ell)} \right\rangle \\
&= \binom{t}{m} s^m m + \binom{t-2}{m-2} s^{m-2} \left\| \sum_{\substack{i\in\{1,2,\ldots,t\}\\ k\in\{1,2,\ldots,s\}}} \mathbf{x}^{(i,k)} \right\|^2 \\
&= \binom{t}{m} s^m m + \binom{t-2}{m-2} s^{m-2} t^2 s^2 \left\|\mathbf{c}\right\|^2 \\
&\le \binom{t}{m} s^m m + \binom{t}{m} s^m m^2 \left\|\mathbf{c}\right\|^2 .
\end{aligned}$$

And putting the three terms in (9.5) together we get

$$\sum_{\substack{U\subseteq[t]\colon |U|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\mathbf{S}(U,\mathbf{m}) - m\mathbf{c}\right\|^2 \le \binom{t}{m} s^m m.$$

□

Now we are ready to prove the new upper bound on the rate of Euclidean s-message codes.

Proof of Theorem 9.3. Take an arbitrary s-message Euclidean code $\mathcal{C}$ for the m-out-of-t case. Let n denote the length of the code, and—similarly to the above lemma—let $\mathbf{c} = \frac{1}{ts}\sum_{i=1}^{t}\sum_{k=1}^{s}\mathbf{x}^{(i,k)}$. Let U be a random variable with uniform distribution over the m sized subsets of $\{1,2,\ldots,t\}$, and let $\mathbf{m}$ be an independent random vector with uniform distribution over $\{1,2,\ldots,s\}^m$. By the definition of expected value,

$$\mathbf{E}\left(\left\|\mathbf{S}(U,\mathbf{m}) - m\mathbf{c}\right\|^2\right) = \frac{1}{\binom{t}{m}s^m} \sum_{\substack{U\subseteq[t]\colon |V|=m\\ \mathbf{m}\in\{1,2,\ldots,s\}^m}} \left\|\mathbf{S}(U,\mathbf{m}) - m\mathbf{c}\right\|^2 ,$$

and using Lemma 9.2:

$$\mathbf{E}\left(\left\|\mathbf{S}(U,\mathbf{m}) - m\mathbf{c}\right\|^2\right) \le m.$$

Jensen's inequality for the random variable $\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|$ says

$$\left(\mathbf{E}\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|\right)^2 \le \mathbf{E}\left(\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|^2\right),$$

so

$$\mathbf{E}\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\| \le \sqrt{m}.$$

Thus by Markov's inequality,

$$\mathbf{P}\left(\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\| > 2\sqrt{m}\right) \le \frac{\mathbf{E}\left(\|\mathbf{S}(U,\mathbf{m})-m\mathbf{c}\|\right)}{2\sqrt{m}} = \frac{1}{2}.$$

This means that at least half of the sum vectors $S(U,\mathbf{m})$ with m active user's lies within an n-dimensional sphere of radius $2\sqrt{m}$.

But $\mathcal{C}$ is an s-message Euclidean code, which means that even those received vectors within the sphere of radius $2\sqrt{m}$ must have distance at least d from each other. Apply the sphere packing argument to these vectors: the sum vectors must differ by Euclidean distance at least d, thus we can draw disjoint n-dimensional spheres of radius $\frac{d}{2}$ around them. This way we get $\binom{t}{m}s^m$ spheres, and at least the half of these spheres are contained within the sphere of radius $2\sqrt{m}+\frac{d}{2}$. For the volumes, we get that

$$\frac{1}{2}\binom{t}{m}s^m\left(\frac{d}{2}\right)^n \le \left(2\sqrt{m}+\frac{d}{2}\right)^n,$$

thus

$$\frac{1}{2}\left(\frac{ts}{m}\right)^m \le \left(1+\frac{4\sqrt{m}}{d}\right)^n,$$

and by taking the logarithm,

$$n \ge \frac{\log\frac{1}{2}+m(\log ts-\log m)}{\log\left(1+\frac{4\sqrt{m}}{d}\right)},$$

and this also holds for the shortest possible s-message Euclidean code with given parameters:

$$N(t,m,s,d) \ge \frac{\log\frac{1}{2}+m(\log ts-\log m)}{\log\left(1+\frac{4\sqrt{m}}{d}\right)},$$

thus

$$\liminf_{m\to\infty}\liminf_{ts\to\infty}\frac{N(t,m,s,d)\log m}{m\log ts} \ge 2.$$

□

For the upper bound, consider that an Euclidean signature code with $t' = ts$ users (and so with $t' = ts$ codewords) is also an s-message Euclidean code for t users. This is because for a signature code with $t' = ts$ users we required that all sum of at most m codewords should be distinct by distance d. For an s message code for t users we require that only those at most m sums must be distinct, which has at most one codeword from all component code. Thus the theorem of Ericson and Györfi provides an upper bound also for the minimal codeword length of s-message Euclidean codes:

Theorem 9.4. *(Ericson–Györfi, (1988))*

$$\limsup_{m\to\infty} \limsup_{ts\to\infty} \frac{N(t,m,s,d)\log m}{m\log ts} \le 4.$$

Appendix A

Linear codes

A.1 Error detection, error correction, erasure error correction

Let $\mathbf{u}$ denote the message vector, and $\hat{\mathbf{u}}$ its reconstruction. The coordinates of vectors $\mathbf{u}$ and $\hat{\mathbf{u}}$ are in set S. This set S is called source alphabet. The message vector $\mathbf{u}$ is encoded into a code word $\mathbf{c} = (c_1, c_2, \dots, c_n)$ which is sent through the channel, and the output vector is denoted by $\mathbf{r}$. The coordinates of vectors $\mathbf{c}$ and $\mathbf{r}$ are in set F. This set F is called code alphabet. Here $\mathbf{r} = (r_1, r_2, \dots, r_n)$ is an n-tuple, a possibly corrupted version of $\mathbf{c}$, which the decoder gets.

If $|S| = |F| = q$ there are exactly q^k q-ary k-tuples $\mathbf{u} = (u_1, u_2, \dots, u_k)$. We can assign one codeword to each such k-tuple and therefore say that we are encoding k "information digits" into n "encoded digits".

We can define an *encoder*, for the length n, q-ary block code $\mathcal{C}$ the set of code words, as a one-to-one mapping from the set of q^k q-ary message vectors to $\mathcal{C}$. We say in this case that the code has parameter (n, k).

It is the task of the decoder to operate on $\mathbf{r}$ to obtain an estimate $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_k)$ of the message vector $\mathbf{u}$. Equivalently, we can view the task of the decoder as that of forming an estimate $\hat{\mathbf{c}} = (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n)$ of the codeword, since the process of recovering $\hat{\mathbf{u}}$ from $\hat{\mathbf{c}}$ is quite trivial compared to that of forming an estimate $\hat{\mathbf{c}}$ of $\mathbf{c}$. It is obvious that the ability to recover correctly $\mathbf{c}$ from $\mathbf{r}$ highly depends on the number of positions where $\mathbf{c}$ and $\mathbf{r}$ differs, i.e., where the channel made an error.

As customary, F^n will denote the set of all n-tuples which has coordinates from F.

Definition A.1. *For* $\mathbf{x} = (x_1, \dots, x_n)$ *and* $\mathbf{y} = (y_1, \dots, y_n)$*, the Hamming distance between* $\mathbf{x}$ *and* $\mathbf{y}$*, denoted* $d(\mathbf{x}, \mathbf{y})$*, is the number of positions in which*

$\mathbf{x}$ *and* $\mathbf{y}$ *differ.*

We can readily see that $d(\mathbf{x}, \mathbf{y})$ satisfies the following three properties:

(1) $d(\mathbf{x}, \mathbf{y}) \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{y}$.
(2) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$. (Symmetry)
(3) $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$. (Triangle inequality)

The three properties above are defining axioms for a *metric*.

Definition A.2. *The minimum distance,* $d_{\min}$*, of the code* $\mathcal{C}$ *is defined as the minimum value of* $d(\mathbf{c}, \mathbf{c}')$ *over all* $\mathbf{c}$ *and* $\mathbf{c}'$ *in* $\mathcal{C}$ *with* $\mathbf{c} \neq \mathbf{c}'$.

We now show the importance of $d_{\min}$ in determining the error detecting or correcting power of the code.

Error detection. Suppose that the task of the decoder is only to detect whether the received vector is erroneous or not. Obviously we can detect the errors by checking the received vector is a codeword or not. If the channel had errors in less than $d_{\min}$ positions then the received vector is surely not a new codeword. Thus the decoder can detect up to $d_{\min} - 1$ errors.

Error correction. A more difficult task is to correct errors. The task of the decoder is to find the "closest" codeword to the received vector measured in Hamming distance. If the channel had errors in less than $d_{\min}/2$ positions then there is a unique codeword which is closest to the received vector. Thus the decoder can correct up to $\lfloor \frac{d_{\min}-1}{2} \rfloor$ errors where $\lfloor . \rfloor$ denotes integer part.

Erasure correction. Another kind of error when we have information of the positions of the errors. This kind of error is called erasure. If we erase less than $d_{\min}$ positions of a codeword, the remainder part cannot correspond to an other codeword otherwise these two codewords would have less than $d_{\min}$ position distinct. Thus the decoder can correct up to $d_{\min} - 1$ erasures.

We saw that the quality of the "closest neighbor" decision in the decoder, which is in some case the maximum likelihood decision, is better if the minimum distance of the code is greater. However it is obvious that if the encoder maps q-ary k-tuples into q-ary n-tuples, then it cannot have arbitrarily high minimum distance. In other words an n-length code with minimum distance $d_{\min}$ cannot have arbitrarily many codewords. One of the simplest upper bound is the so called Singleton bound on the size of a code.

Theorem A.1. *A* q*-ary code of length* n *and minimum distance* $d_{\min}$ *cannot have more than* $q^{n-d_{\min}+1}$ *codewords. In other words, for an* (n, k) *parameter code*

$$d_{\min} \leq n - k + 1.$$

Proof. Since the number of distinct q-ary $k-1$-tuples are q^{k-1} and there are q^k codewords, there must be at least two codewords $\mathbf{c}$ and $\mathbf{c}'$ such that they coincide in the first $k-1$ positions. For these

$$d(\mathbf{c}, \mathbf{c}') \leq n - k + 1,$$

which implies

$$d_{\min} \leq n - k + 1,$$

therefore

$$q^k \leq q^{n-d_{\min}+1}.$$

□

An important class of the codes which reach the Singleton bound is the following.

Definition A.3. *An (n, k) parameter code with*

$$d_{\min} = n - k + 1$$

is called maximum distance separable (or MDS) code.

A.2 Finite fields

Structured code alphabet F is necessary to construct efficient error-control codes. We introduce algebraic operations on F for this purpose.

A finite group is a finite set of elements together with a binary operation that satisfy the four fundamental properties of closure, associativity, the identity property, and the inverse property. A group with a commutative operation is called commutative (Abelian) group.

Definition A.4. *A field is an algebraic system $(F, +, \cdot)$ consisting of a set F and operations $+$ (addition) and $\cdot$ (multiplication) such that:*

1. *$(F, +)$ is a commutative (Abelian) group having the neutral element 0.*

2. *$(F \setminus \{0\}, \cdot)$ is commutative (Abelian) group, and define $a \cdot 0 = 0$ for all $a \in F$.*

3. *The addition is distributive in respect of the multiplication: for every a, b and c in F, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.*

A finite field, or Galois field, is a field $(F, +, \cdot)$ in which F is a finite set. A Galois field of size p is denoted by $\mathrm{GF}(p)$. One can prove that the size of a Galois field is either a prime, or a power of prime.

First we discuss finite fields of prime number of elements. We shall write $a \bmod b$ to denote the unique remainder when the integer a is divided by the non-zero integer b.

For a prime p the algebraic system $\mathrm{GF}(p) = (F, \oplus, \odot)$, where $F = \{0, 1, \ldots p-1\}$ and where the field operations are defined for a and b in F by

$$a \oplus b = a + b \mod p \tag{A.1}$$

$$a \odot b = a \cdot b \mod p \tag{A.2}$$

(where the operations on the right sides are integer addition and multiplication, respectively) is a field.

Next we shall see how to construct a bigger field $(F^m, +, \cdot)$ that contains a given field $(F, +, \cdot)$. But, in order to do this, we need to exploit some properties of polynomials.

Let $(F, +, \cdot)$ be an arbitrary field. Then a *polynomial* over $(F, +, \cdot)$ in the indeterminate x is an expression of the form $a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots$ in which $a_i \in F$ for all i, but in which at most a finite number of the "coefficients" a_i are non-zero. If $A(x) = a_0 + a_1 \cdot x + \cdots + a_n \cdot x^n$ is a polynomial with $a_n \neq 0$, then n is the *degree* of $A(x)$, denoted $\deg(A(x))$, and a_n is called the *leading coefficient*. When the leading coefficient is 1, the polynomial is called *monic*.

Elements of $(F^m, +, \cdot)$ are vectors $(c_0, c_1, \ldots, c_{m-1})$ of length m over the field $(F, +, \cdot)$ which are represented by polynomials $c_0 + c_1 \cdot x + \cdots + c_{m-1} \cdot x^{m-1}$ of degree at most $m-1$ and with coefficients in F. Whether adding or multiplying polynomials, the arithmetic for the coefficients is carried out in the field F. The powers of the indeterminate are, however, always ordinary integers. We shall write $F[x]$ to denote the set of all polynomials over the field F.

The polynomial property analogous to the integer property of "primeness" is "irreducibility". A polynomial $P(x)$ in $F[x]$ is *irreducible in* $F[x]$ if $\deg(P(x)) \geq 1$ and $P(x)$ cannot be written as a product of polynomials in $F[x]$, each having degree smaller than $\deg(P(x))$. Notice that all first degree polynomials are trivially irreducible.

Let p be a prime, then we introduce the arithmetic of $\mathrm{GF}(p^m)$. The elements of $\mathrm{GF}(p^m)$ are considered as vectors from $\mathrm{GF}(p)^m$, and are represented by the corresponding polynomials. So for $\mathbf{a}, \mathbf{b} \in \mathrm{GF}(p^m)$

$$\mathbf{a} = (a_0, a_1, \ldots, a_{m-1})$$

and

$$\mathbf{b} = (b_0, b_1, \ldots, b_{m-1})$$

and the corresponding polynomials

$$a(x) = a_0 + a_1 x \cdots + a_{m-1} x^{m-1}$$

and

$$b(x) = b_0 + b_1 x \cdots + b_{m-1} x^{m-1}.$$

The addition in $\mathrm{GF}(p^m)$ is defined by

$$c(x) = a(x) + b(x).$$

Concerning the multiplication, choose an irreducible polynomial $P(x)$ of degree m. One can prove that such a polynomial always exists. The multiplication in $\mathrm{GF}(p^m)$ is defined by

$$d(x) = a(x) \cdot b(x) \mod P(x).$$

A.3 Linear codes

Brief summary of linear algebra: linear space, linear independence, basis.

Definition A.5. *A block code $\mathcal{C}$ of length n is linear if its codewords form a vector space over the field F, i.e., if $\mathcal{C}$ is a subspace of F^n. To check to see if $\mathcal{C}$ is a subspace of F^n, we need only check to see that*
(1) if $\mathbf{x}$ and $\mathbf{y}$ are in $\mathcal{C}$, then so is $\mathbf{x} + \mathbf{y}$,
(2) if $\mathbf{x}$ is in $\mathcal{C}$, then so is $c \cdot \mathbf{x}$ for all $c \in F$.

Suppose $\mathcal{C}$ is linear and $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$ are a basis for $\mathcal{C}$, i.e., $\mathcal{C}$ is a k-dimensional subspace of F^n. Then the q^k vectors

$$\mathbf{c} = u_1 \cdot \mathbf{g}_1 + u_2 \cdot \mathbf{g}_2 + \cdots + u_k \cdot \mathbf{g}_k, \qquad \forall u_1, \ldots, u_k \in F \tag{A.3}$$

are all and only the codewords of $\mathcal{C}$. Writing $\mathbf{g}_i = (g_{i1}, g_{i2}, \ldots, g_{in})$, we can then rewrite (A.3) in matrix form as

$$(c_1, c_2, \ldots, c_n) = (u_1, u_2, \ldots, u_k) \cdot \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix} \tag{A.4}$$

or, more compactly, as $\mathbf{c} = \mathbf{u}\cdot\mathbf{G}$ where

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}.$$

We can consider (A.4) as defining a linear code $\mathcal{C}$ and, thus we shall call $\mathbf{G}$ an *encoding matrix* or *generator matrix.* The encoding matrix of code $\mathcal{C}$ is any matrix whose rows are a basis for $\mathcal{C}$. In particular for $\mathcal{C}(n, k)$ the encoding matrix must be a $k \times n$ matrix.

A linear encoder $\mathbf{G}$ is said to be *systematic* whenever its use results in

$$(c_1, c_2, \dots, c_k) = (u_1, u_2, \dots, u_k),$$

i.e., whenever the information digits appear unchanged in the first k components of the codeword. We see from this definition that $\mathbf{G}$ is systematic if and only if it has the form

$$\mathbf{G} = (\mathbf{I}_k|\mathbf{P}) \tag{A.5}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix and where $\mathbf{P}$ is some $k \times (n-k)$ matrix. Clearly if $\mathcal{C}$ has a systematic encoder $\mathbf{G}$, then this systematic encoder is unique. It can be proved with Gauss-elimination that either a linear code has a systematic encoder, or one can rearrange the order of the digits in the codeword so that the new code does. Hence, for most purposes, it suffices to consider *systematic linear codes*, i.e., linear codes which has a systematic encoder as in (A.5).

Definition A.6. *The Hamming weight of a vector* $\mathbf{x}$*, denoted* $w(\mathbf{x})$*, is defined as the number of non-zero components in* $\mathbf{x}$*.*

Clearly then,

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} - \mathbf{y}).$$

Definition A.7. *The minimum weight,* $w_{\min}$*, of a linear code* $\mathcal{C}$ *is the smallest value of* $w(\mathbf{c})$ *for* $\mathbf{c} \in \mathcal{C}$ *and* $\mathbf{c} \neq \mathbf{0}$*.*

One of the most important facts about linear codes is

Theorem A.2. *For a linear code,* $w_{\min} = d_{\min}$*.*

Proof. Let $\mathbf{c}$ be a non-zero codeword such that $w(\mathbf{c}) = w_{\min}$. Then, since $\mathbf{0}$ is also a codeword, $d_{\min} \le d(\mathbf{c}, \mathbf{0}) = w(\mathbf{c}) = w_{\min}$. Conversely, let $\mathbf{c}_1$ and $\mathbf{c}_2$ ($\mathbf{c}_1 \neq \mathbf{c}_2$) be codewords such that $d(\mathbf{c}_1, \mathbf{c}_2) = d_{\min}$. But $\mathbf{c} = \mathbf{c}_1 - \mathbf{c}_2 \neq \mathbf{0}$ is also a codeword and $w_{\min} \le w(\mathbf{c}) = w(\mathbf{c}_1 - \mathbf{c}_2) = d(\mathbf{c}_1, \mathbf{c}_2) = d_{\min}$. □

The importance of Theorem A.2 is that one need not look at distances between all *pairs* of distinct codewords to find $d_{\min}$ for a linear code. One can just as well look only at the weights of single codewords.

A.4 Shortened Reed–Solomon codes

We need a few more algebraic concepts to construct some of the most interesting and powerful error-correcting codes yet discovered.

In any field GF(q), we say that a number α is a *primitive element* when $m = q-1$ is the smallest positive integer that $\alpha^m = 1$. When α is a primitive element, then $\alpha, \alpha^2, \ldots, \alpha^m = 1$ must off be non-zero and distinct because $\alpha^j = \alpha^i$ for $1 \leq i < j \leq m$ would imply $\alpha^{j-i} = 1$ and $1 \leq j-i < m$. One can prove that each Galois field has a primitive element.

In this section we introduce the shortened Reed–Solomon codes, and then we give a proof of its MDS property.

Definition A.8. *Let $\alpha \in GF(q)$ be a primitive element and*

$$u(x) = u_0 + u_1 x + \cdots + u_{k-1} x^{k-1}$$

be the message polynomial in $GF(q)[x]$. Then the codewords of the (n, k) shortened Reed–Solomon code $(n \leq q-1)$ determined by α are

$$\begin{aligned} c_0 &= u(1) \\ c_1 &= u(\alpha) \\ c_2 &= u(\alpha^2) \\ \vdots &= \vdots \\ c_{n-1} &= u(\alpha^{n-1}). \end{aligned}$$

If $n = q-1$ then this code is called Reed–Solomon code.

It is easy to see that the shortened Reed–Solomon code is a q-ary linear code for which

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \alpha^{(k-1)2} & \ldots & \alpha^{(k-1)(n-1)} \end{pmatrix} \tag{A.6}$$

is a generator matrix.

Theorem A.3. *The minimum distance of* (n,k) *shortened Reed–Solomon codes is*

$$d_{\min} = n - k + 1,$$

i.e., the shortened Reed–Solomon codes are MDS codes.

Proof. Because of the Singleton bound $d_{\min} \leq n - k + 1$ for any (n,k) code, it is sufficient to prove $d_{\min} \geq n - k + 1$. The shortened Reed–Solomon code is linear, so we should prove that $w(\mathbf{c}) \geq n - k + 1$ for any non-zero codeword in the code. Thus we should lower bound the weights of non-zero codewords, which is equivalent to give an upper bound on the number of zero components of non-zero codewords. But the zero components of the codewords correspond to distinct roots of the message polynomial, and by the fundamental theorem a non-zero polynomial cannot have more roots than its degree. Formally,

$$\begin{aligned} w(\mathbf{c}) &= |\{\text{non-zero coordinates of } \mathbf{c}\}| \\ &= n - |\{\text{zero coordinates of } \mathbf{c}\}| \\ &\geq n - |\{\text{roots of } u(x)\}| \\ &\geq n - (k-1), \end{aligned}$$

so the theorem follows. □

A.5 Shortened Bose-Chaudhuri-Hocquenghem codes

A linear code over $GF(q^m)$ will have some codewords all of whose components are in the smaller field $GF(q)$; $(0,0,\ldots,0)$ is always one such codeword. The entire set of these codewords with components in $GF(q)$ is called the $GF(q)$ *subcode* of the original $GF(q^m)$ code. This $GF(q)$ subcode is a vector space over the smaller field $GF(q)$ and hence is a linear code over $GF(q)$. Its minimum weight, and hence also its minimum distance, cannot be less than that of the original code since its non-zero codewords are subset of those in the original code. But in general the new code will have fewer information digits since, when we use a systematic encoder for the original code, we cannot get a codeword all of whose components are in $GF(q)$ unless the k information digits are all in $GF(q)$; but some of the q^k choices of the k information digits as elements of $GF(q)$ in the original code might yield codewords whose $n-k$ parity digits are not all in $GF(q)$. Thus, the $GF(q)$ subcode will in general have fewer than q^k codewords. We summarize these facts in the following theorem.

Theorem A.4. *If $\mathcal{C}$ is an (n,k) linear code over $GF(q^m)$ with minimum distance $d_{\min}$, then its $GF(q)$ subcode is an (n,k') linear code over $GF(q)$ with minimum distance $d'_{\min}$, where*

$$k' \leq k$$

and

$$d'_{\min} \geq d_{\min}.$$

With this background, we can now define the *shortened Bose-Chaudhuri-Hocquenghem codes* (or shortened BCH codes).

Consider a shortened Reed–Solomon code over $\mathrm{GF}(q^m)$ with parameters (n,k). The (n,k') shortened BCH code over $\mathrm{GF}(q)$ is the $\mathrm{GF}(q)$ subcode of the shortened Reed–Solomon code. It follows immediately from Theorem A.4 that, for the shortened BCH code,

$$d'_{\min} \geq n-k+1 \tag{A.7}$$

and

$$k' \leq k. \tag{A.8}$$

(Note that (A.8) also could have been obtained from the Singleton-bound (Theorem A.1).) We shall see in later examples that (A.7) often holds with equality, but that the bound (A.8) is usually very loose for BCH codes.

This definition of shortened BCH codes does not give much insight into their nature. This will be remedied in a later section where we will show how one can determine k' exactly.

A.6 Cyclic codes

By *cyclic shift* of an n-tuple $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, denoted $\mathbf{c}\cdot\mathbf{T}$, we mean the n-tuple $(c_{n-1}, c_0, \ldots, c_{n-2})$. Notice that when the digits c_i are in a field we can consider the cyclic shift operator $\mathbf{T}$ to be the matrix

$$\mathbf{T} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix},$$

which emphasizes that taking the cyclic shifts is a linear operation, i.e., $(a\cdot\mathbf{c} + b\cdot\mathbf{c}')\cdot\mathbf{T} = a\cdot\mathbf{c}\cdot\mathbf{T} + b\cdot\mathbf{c}'\cdot\mathbf{T}$.

Definition A.9. *A block code (whether linear or not) is said to be cyclic if the cyclic shift of every codeword is also a codeword.*

We shall now examine the general structure of cyclic codes, after which we shall show that RS codes and BCH codes are cyclic.

It is convenient to identify the n-tuple $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ with components in the field $F = GF(q)$ with the polynomial

$$c(x) = c_0 + c_1 \cdot x + \cdots + c_{n-1} \cdot x^{n-1} \tag{A.9}$$

of degree less than n in $F[x]$. We shall speak interchangeably of $\mathbf{c}$ or $c(x)$ since either uniquely specifies the other. Notice that the cyclic shift, $\mathbf{c} \cdot \mathbf{T}$, has the polynomial representation

$$c_{n-1} + c_0 \cdot x + c_1 \cdot x^2 + \cdots + c_{n-2} \cdot x^{n-1} = x \cdot c(x) - c_{n-1} \cdot (x^n - 1)$$

which we can also write as

$$x \cdot c(x) - c_{n-1} \cdot (x^n - 1) = x \cdot c(x) \mod (x^n - 1). \tag{A.10}$$

Equation (A.10) shows that cyclic shifting of $\mathbf{c}$ corresponds to multiplication of $\mathbf{c}$ by x, modulo the polynomial $x^n - 1$.

Suppose that $\mathcal{C}$ is a linear cyclic code, and that the minimum degree among the polynomials corresponding to its non-zero codewords is r. We first claim that there is a unique monic polynomial of degree r in the code; since if there were two their difference would be a non-zero polynomial of degree less than r and also in the code (because $\mathcal{C}$ is linear) and this would contradict the definition of r. We shall hereafter write

$$g(x) = g_0 + g_1 \cdot x + \cdots + g_{r-1} \cdot x^{r-1} + x^r$$

to denote this unique monic polynomial of minimum degree in the code. This polynomial corresponds to the codeword $(g_0, g_1, \ldots, g_{r-1}, 1, 0, \ldots, 0)$ and has $n-r-1$ ending zeros. Because $\mathcal{C}$ is cyclic, the n-tuples $(0, g_0, g_1, \ldots, g_{r-1}, 1, 0, \ldots, 0)$, $(0, 0, g_0, g_1, \ldots, g_{r-1}, 1, 0, \ldots, 0)$, $\ldots$, $(0, \ldots, 0, g_0, g_1, \ldots, g_{r-1}, 1)$ must all be in the code and these n-tuples correspond to the polynomials $x \cdot g(x)$, $x^2 \cdot g(x)$,$\ldots$, $x^{n-r-1} \cdot g(x)$. Again because $\mathcal{C}$ is linear,

$$u_0 \cdot g(x) + u_1 \cdot x \cdot g(x) + \cdots + u_{n-r-1} \cdot x^{n-r-1} \cdot g(x)$$

must also be in the code for all choices of $u_0, u_1, \ldots, u_{n-r-1}$ in $GF(q)$. Letting

$$u(x) = u_0 + u_1 \cdot x + \cdots + u_{n-r-1} x^{n-r-1},$$

we can equivalently say that $u(x){\cdot}g(x)$ must be also in the code for every choice of $u(x)$, a polynomial of degree less than $n-r$. One can show that there are no other codewords, which will further imply that

$$k = n - r,$$

because we can take $u_0, u_1, \ldots, u_{k-1}$ as the information digits if we wish.

We next show that $g(x)$ divides $x^n - 1$. We have already seen that $x^{k-1}{\cdot}g(x)$ is in the code. But $\mathcal{C}$ is cyclic so that the cyclic shift of this codeword is also a codeword, i.e., $x^k{\cdot}g(x) - (x^n - 1)$ or, equivalently, that $g(x)$ must divide $x^n - 1$ as we wished to show.

The code generated by a monic polynomial $g(x)$ of degree r that divides $x^n - 1$ is cyclic and has $k = n - r$ information digits. If $c(x) = u(x){\cdot}g(x)$ has degree less than n, then $u(x)$ has degree less than k thus we can choose the k coefficients of the polynomial $u(x)$ as the information digits. The cyclic shift of $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ is represented by

$$x{\cdot}c(x) \operatorname{mod}(x^n - 1) = x{\cdot}c(x) - c_{n-1}{\cdot}(x^n - 1)$$

where both term on the right side is divisible by $g(x)$ therefore the cyclic shift of $\mathbf{c}$ is in the code.

We have thus proved the following theorem that states the principal structural features of linear cyclic codes.

Theorem A.5. *There is a unique monic code polynomial, $g(x)$, of degree $n-k$ in every q-ary linear cyclic (n,k) code. This generating polynomial, $g(x)$, specifies the code in the sense that the q-ary n-tuple $\mathbf{c}$ is a codeword if and only if $g(x)$ divides $c(x)$; moreover, $g(x)$ divides $x^n - 1$. Conversely, every q-ary monic polynomial of degree r that divides $x^n - 1$ generates such a q-ary linear cyclic (n,k) code with $k = n - r$.*

Our proof of Theorem A.5 establishes that the $k \times n$ matrix

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & \ldots & g_{n-k-1} & 1 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & g_{n-k-1} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ldots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & g_0 & g_1 & \ldots & g_{n-k-1} & 1 \end{pmatrix} \tag{A.11}$$

is an encoding matrix for the cyclic code generated by $g(x)$, because it showed that the rows of this matrix are a basis for the vector space of codewords. Since the first k columns form a non-singular matrix, we can by appropriate linear operations on the rows of $\mathbf{G}$ obtain an encoding matrix of the form $\mathbf{G}' = (\mathbf{I}_k|\mathbf{P})$. Thus *every linear cyclic code is a systematic code.*

$g(x)$ divides $x^n - 1$, therefore we can introduce

$$h(x) = \frac{x^n - 1}{g(x)}$$

which is called *parity check polynomial.* Since the generating polynomial uniquely determines the linear cyclic code, so the parity check polynomial does it, too.

It is easy to prove that a polynomial $c(x)$ with degree at most $n-1$ is a code word polynomial if and only if

$$c(x)h(x) = 0 \mod(x^n - 1).$$

A.7 Reed–Solomon codes

We now show that the Reed–Solomon codes over GF(q) (shortened Reed–Solomon codes with $n = q-1$) are cyclic.

Theorem A.6. *The (n,k) Reed–Solomon code determined by a primitive element α is cyclic if $n = q-1$.*

Proof. We must show that for any $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ in the code $\mathbf{cT} = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathcal{C}$ also holds. From Definition A.8 it follows that

$$\begin{aligned}
(\mathbf{cT})_i &= c_{i-1 \bmod n} \\
&= u(\alpha^{i-1 \bmod n}) \\
&= \sum_{j=0}^{k-1} u_j \alpha^{(i-1 \bmod n)j} \\
&= \sum_{j=0}^{k-1} u_j \alpha^{(i-1)j} \\
&= \sum_{j=0}^{k-1} u_j \alpha^{-1} \alpha^{ij} \\
&= \sum_{j=0}^{k-1} \hat{u}_j \alpha^{ij} \\
&= \hat{u}(\alpha^i) \\
&= \hat{c}_i,
\end{aligned}$$

therefore the cyclic shift of $\mathbf{c}$ is also in the code. □

A.8 BCH codes

Here we introduce just a special case of the primitive BCH code over GF(p) with parameters (n, k'), where p is a prime. Concerning the concept of general BCH code and the details we refer to Blahut (1984). Choose an integer $r \geq 1$ and let the length n of the code be the so called primitive length:

$$n = p^r - 1.$$

We have the unique factorization

$$x^n - 1 = M_0(x) \cdot \dots \cdot M_s(x)$$

over GF(p), where $M_j(x)$ are irreducible monic polynomials over GF(p). Let α be a primitive element of GF(p^r). Over GF(p^r),

$$x^n - 1 = (x - \alpha^0) \cdot (x - \alpha^1) \cdot \dots \cdot (x - \alpha^{p^r-2}),$$

where α^i are the non-zero elements of GF(p^r). It implies that for each α^i there is an $M_j(x)$ such that $(x - \alpha^i)$ divides $M_j(x)$. Then we say that $M_j(x)$ is the minimal polynomial of α^i.

Both $g(x)$ and $h(x)$ should be a product of some $M_j(x)$'s. Concerning $g(x)$, the minimum distance of the code can be increased by increasing the number of factors of $g(x)$, while the size of the code ($p^{k'}$) is large if the degree of $g(x)$ is small.

Choose $1 \leq k \leq p$. Let $M_i(x)$ denote the minimal polynomial of α^i, $i = 0, 1, \dots, k-1$. Choose the parity check polynomial of the primitive BCH code as

$$h(x) = \text{l.c.m}\{M_0(x), \dots, M_{k-1}(x)\}.$$

We have that

$$M_0(x) = x - 1.$$

If M(x) is the primitive polynomial of a $\beta \in GF(p^r)$ then

$$M(x) = (x - \beta)(x - \beta^p) \dots (x - \beta^{p^{m-1}}),$$

where m is such that $\beta^{p^m} = \beta$ (cf. Theorem 5.3.6 in Blahut (1984)). Thus, because of $k \leq p$, for each $1 \leq j \leq k-1$

$$M_j(x) = \prod_{i=1}^{r} (x - \alpha^{jp^{i-1}}),$$

and therefore $\deg M_j(x) = r$ for $j \geq 1$, and $M_1(x), \ldots, M_{k-1}(x)$ are different, and

$$h(x) = \prod_{j=0}^{k-1} M_j(x),$$

and

$$k' = \deg h(x) = (k-1)r + 1. \tag{A.12}$$

One can calculate a lower bound on the minimum distance of the code, called designed distance d:

$$d_{\min} \geq d = p^r - 1 - (k-1)p^{r-1}. \tag{A.13}$$

Appendix B

Probability

B.1 Inequalities

Lemma B.1. (Chernoff (1952)). *Let B be a binomial random variable with parameters n and p. Then, for $1 > \epsilon > p > 0$,*

$$\mathbf{P}\{B > n\epsilon\} \le e^{-n\left[\epsilon \log \frac{\epsilon}{p} + (1-\epsilon)\log\frac{1-\epsilon}{1-p}\right]} \le e^{-n[p-\epsilon+\epsilon\log(\epsilon/p)]}$$

and, for $0 < \epsilon < p < 1$,

$$\mathbf{P}\{B < n\epsilon\} \le e^{-n\left[\epsilon \log \frac{\epsilon}{p} + (1-\epsilon)\log\frac{1-\epsilon}{1-p}\right]} \le e^{-n[p-\epsilon+\epsilon\log(\epsilon/p)]}.$$

Proof. We proceed by Chernoff's exponential bounding method. In particular, for arbitrary $s > 0$,

$$\begin{aligned}
\mathbf{P}\{B > n\epsilon\} &= \mathbf{P}\{sB > sn\epsilon\} \\
&= \mathbf{P}\{e^{sB} > e^{sn\epsilon}\} \\
&\le e^{-sn\epsilon}\mathbf{E}\{e^{sB}\} \\
&\quad \text{(by the Markov inequality)} \\
&= e^{-sn\epsilon}\sum_{k=0}^{n} e^{sk}\binom{n}{k}p^k(1-p)^{n-k} \\
&= e^{-sn\epsilon}(e^s p + 1 - p)^n \\
&= [e^{-s\epsilon}(e^s p + 1 - p)]^n.
\end{aligned}$$

Next choose s such that

$$e^s = \frac{\epsilon}{1-\epsilon}\frac{1-p}{p}.$$

With this value we get

$$
\begin{aligned}
e^{-s\epsilon}(e^s p+1-p) &= e^{-\epsilon\cdot\log\left(\frac{\epsilon}{1-\epsilon}\frac{1-p}{p}\right)}\cdot\left(\frac{\epsilon}{1-\epsilon}\frac{1-p}{p}\cdot p+1-p\right)\\
&= e^{-\epsilon\cdot\log\left(\frac{\epsilon}{p}\frac{1-p}{1-\epsilon}\right)}\cdot\left(\epsilon\cdot\frac{1-p}{1-\epsilon}+1-p\right)\\
&= e^{-\epsilon\cdot\log\frac{\epsilon}{p}-\epsilon\cdot\log\frac{1-p}{1-\epsilon}+\log\frac{1-p}{1-\epsilon}}\\
&= e^{-\epsilon\cdot\log\frac{\epsilon}{p}+(1-\epsilon)\cdot\log\frac{1-p}{1-\epsilon}},
\end{aligned}
$$

which implies the first inequality.

The second inequality follows from

$$
\begin{aligned}
(1-\epsilon)\log\frac{1-\epsilon}{1-p} &= -(1-\epsilon)\log\frac{1-p}{1-\epsilon}\\
&= -(1-\epsilon)\log\left(1+\frac{\epsilon-p}{1-\epsilon}\right)\\
&\geq -(1-\epsilon)\cdot\frac{\epsilon-p}{1-\epsilon}\\
&\quad\text{(by }\log(1+x)\leq x)\\
&= p-\epsilon.
\end{aligned}
$$

To prove the second half of the lemma, observe that $n-B$ is a binomial random variable with parameters n and $1-p$. Hence for $\epsilon<p$ the results of the first step imply that

$$
\begin{aligned}
\mathbf{P}\{B<n\epsilon\} &= \mathbf{P}\{n-B>n(1-\epsilon)\}\\
&\leq e^{-n\left[(1-\epsilon)\log\frac{1-\epsilon}{1-p}+\epsilon\log\frac{\epsilon}{p}\right]}\\
&= e^{-n\left[\epsilon\log\frac{\epsilon}{p}+(1-\epsilon)\log\frac{1-\epsilon}{1-p}\right]}\\
&\leq e^{-n[p-\epsilon+\epsilon\log(\epsilon/p)]}.
\end{aligned}
$$

□

Lemma B.2 (Bernstein (1946)). *Let $X_1,\ldots,X_n$ be independent real-valued random variables, let $a,b\in\mathcal{R}$ with $a<b$, and assume that $X_i\in[a,b]$ with probability one $(i=1,\ldots,n)$. Let*

$$
\sigma^2=\frac{1}{n}\sum_{i=1}^{n}\mathbf{Var}\{X_i\}>0.
$$

Then, for all $\epsilon > 0$*,*

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}\{X_i\})\right| > \epsilon\right\} \leq 2e^{-\frac{n\epsilon^2}{2\sigma^2+2\epsilon(b-a)/3}}.$$

Proof. Set $Y_i = X_i - \mathbf{E}\{X_i\}$ $(i = 1, \ldots, n)$. Then we have, with probability one,

$$|Y_i| \leq b - a \quad \text{and} \quad \mathbf{E}\{Y_i^2\} = \mathbf{Var}\{X_i\} \quad (i = 1, \ldots, n).$$

By Chernoff's exponential bounding method we get, for arbitrary $s > 0$,

$$\begin{aligned}
\mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}\{X_i\}) > \epsilon\right\} &= \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n} Y_i > \epsilon\right\} \\
&= \mathbf{P}\left\{s\sum_{i=1}^{n} Y_i - sn\epsilon > 0\right\} \\
&\leq \mathbf{E}\left\{e^{s\sum_{i=1}^{n} Y_i - sn\epsilon}\right\} \\
&= e^{-sn\epsilon}\prod_{i=1}^{n}\mathbf{E}\{e^{sY_i}\},
\end{aligned}$$

by the independence of Y_i's. Because of $|Y_i| \leq b - a$ a.s.

$$\begin{aligned}
e^{sY_i} &= 1 + sY_i + \sum_{j=2}^{\infty}\frac{(sY_i)^j}{j!} \\
&\leq 1 + sY_i + \sum_{j=2}^{\infty}\frac{s^j Y_i^2 (b-a)^{j-2}}{2 \cdot 3^{j-2}} \\
&= 1 + sY_i + \frac{s^2 Y_i^2}{2}\sum_{j=2}^{\infty}\left(\frac{s\,(b-a)}{3}\right)^{j-2} \\
&= 1 + sY_i + \frac{s^2 Y_i^2}{2}\frac{1}{1 - s(b-a)/3}
\end{aligned}$$

if $|s(b-a)/3| < 1$. This, together with $\mathbf{E}\{Y_i\} = 0$ $(i = 1, \ldots, n)$ and $1+x \leq e^x$ $(x \in \mathcal{R})$, implies

$$\begin{aligned}
&\mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}\{X_i\}) > \epsilon\right\} \\
&\leq e^{-sn\epsilon}\prod_{i=1}^{n}\left(1 + \frac{s^2\,\mathbf{Var}\{X_i\}}{2}\frac{1}{1 - s(b-a)/3}\right)
\end{aligned}$$

$$\begin{aligned}
&\leq \quad e^{-sn\epsilon}\prod_{i=1}^{n}\exp\left(\frac{s^2\,\mathbf{Var}\{X_i\}}{2}\frac{1}{1-s(b-a)/3}\right)\\
&= \quad \exp\left(-sn\epsilon+\frac{s^2n\sigma^2}{2(1-s(b-a)/3)}\right).
\end{aligned}$$

Set

$$s=\frac{\epsilon}{\epsilon(b-a)/3+\sigma^2}.$$

Then

$$\left|\frac{s(b-a)}{3}\right|<1$$

and

$$\begin{aligned}
&-sn\epsilon+\frac{s^2n\sigma^2}{2(1-s(b-a)/3)}\\
&=\frac{-n\epsilon^2}{\epsilon(b-a)/3+\sigma^2}+\frac{\epsilon^2}{(\epsilon(b-a)/3+\sigma^2)^2}\cdot\frac{n\sigma^2}{2\left(1-\frac{\epsilon(b-a)/3}{\epsilon(b-a)/3+\sigma^2}\right)}\\
&=\frac{-n\epsilon^2}{\epsilon(b-a)/3+\sigma^2}+\frac{\epsilon^2}{\epsilon(b-a)/3+\sigma^2}\cdot\frac{n\sigma^2}{2\left(\epsilon(b-a)/3+\sigma^2-\epsilon(b-a)/3\right)}\\
&=\frac{-n\epsilon^2}{2\epsilon(b-a)/3+2\sigma^2},
\end{aligned}$$

hence

$$\mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mathbf{E}X_i)>\epsilon\right\}\leq\exp\left(\frac{-n\epsilon^2}{2\epsilon(b-a)/3+2\sigma^2}\right).$$

Similarly,

$$\begin{aligned}
\mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i-\mathbf{E}X_i)<-\epsilon\right\} &= \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(-X_i-\mathbf{E}\{-X_i\})>\epsilon\right\}\\
&\leq \exp\left(\frac{-n\epsilon^2}{2\epsilon(b-a)/3+2\sigma^2}\right),
\end{aligned}$$

which implies the assertion. □

Lemma B.3 (Hoeffding (1963)). *Let $X_1,\dots,X_n$ be independent real-valued random variables, let $a_1,b_1,\dots,a_n,b_n\in\mathcal{R}$, and assume that $X_i\in[a_i,b_i]$ with probability one ($i=1,\dots,n$). Then, for all $\epsilon>0$,*

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}(X_i-\mathbf{E}\{X_i\})\right|>\epsilon\right\}\leq 2e^{-\frac{2n\epsilon^2}{\frac{1}{n}\sum_{i=1}^{n}|b_i-a_i|^2}}.$$

Proof. Let $s > 0$ be arbitrary. Similarly to the proof of Lemma B.2 we get

$$\begin{aligned} & \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i) > \epsilon\right\} \\ \leq \; & \exp(-sn\epsilon)\cdot\prod_{i=1}^{n}\mathbf{E}\left\{\exp\left(s\cdot(X_i - \mathbf{E}X_i)\right)\right\}. \end{aligned}$$

We will show momentarily

$$\mathbf{E}\left\{\exp\left(s\cdot(X_i - \mathbf{E}X_i)\right)\right\} \leq \exp\left(\frac{s^2(b_i - a_i)^2}{8}\right) \quad (i = 1, \ldots, n), \tag{B.1}$$

from which we can conclude

$$\mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i) > \epsilon\right\} \leq \exp\left(-sn\epsilon + \frac{s^2}{8}\sum_{i=1}^{n}(b_i - a_i)^2\right).$$

The right-hand side is minimal for

$$s = \frac{4\,n\,\epsilon}{\sum_{i=1}^{n}(b_i - a_i)^2}.$$

With this value we get

$$\begin{aligned} & \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i) > \epsilon\right\} \\ \leq \; & \exp\left(-\frac{4n\epsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i - a_i)^2} + \frac{2n\epsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i - a_i)^2}\right) \\ = \; & \exp\left(-\frac{2n\epsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i - a_i)^2}\right). \end{aligned}$$

This implies that

$$\begin{aligned} & \mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i)\right| > \epsilon\right\} \\ & = \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbf{E}X_i) > \epsilon\right\} + \mathbf{P}\left\{\frac{1}{n}\sum_{i=1}^{n}(-X_i - \mathbf{E}\{-X_i\}) > \epsilon\right\} \\ & \leq 2\exp\left(-\frac{2n\epsilon^2}{\frac{1}{n}\sum_{i=1}^{n}(b_i - a_i)^2}\right). \end{aligned}$$

So it remains to show (B.1). Fix $i \in \{1, \ldots, n\}$ and set

$$Y = X_i - \mathbf{E}X_i.$$

Then $Y \in [a_i - \mathbf{E}X_i, b_i - \mathbf{E}X_i] =: [a, b]$ with probability one, $a - b = a_i - b_i$, and $\mathbf{E}Y = 0$. We have to show

$$\mathbf{E}\left\{\exp(sY)\right\} \leq \exp\left(\frac{s^2(b-a)^2}{8}\right). \tag{B.2}$$

Because of e^{sx} convex we have

$$e^{sx} \leq \frac{x-a}{b-a}e^{sb} + \frac{b-x}{b-a}e^{sa} \quad \text{for all } a \leq x \leq b,$$

thus

$$\begin{aligned} \mathbf{E}\{\exp(sY)\} &\leq \frac{\mathbf{E}\{Y\} - a}{b-a}e^{sb} + \frac{b - \mathbf{E}\{Y\}}{b-a}e^{sa} \\ &= e^{sa}\left(1 + \frac{a}{b-a} - \frac{a}{b-a}e^{s(b-a)}\right) \\ &\qquad \text{(because of } \mathbf{E}\{Y\} = 0\text{).} \end{aligned}$$

Setting

$$p = -\frac{a}{b-a}$$

we get

$$\mathbf{E}\{\exp(sY)\} \leq (1 - p + p \cdot e^{s(b-a)})e^{-s\,p\,(b-a)} = e^{\Phi(s(b-a))},$$

where

$$\Phi(u) = \ln\left((1 - p + pe^u)e^{-pu}\right) = \ln\left(1 - p + pe^u\right) - pu.$$

Next we make a Taylor expansion of Φ. Because of

$$\Phi(0) = 0,$$

$$\Phi'(u) = \frac{pe^u}{1 - p + pe^u} - p, \text{ hence } \Phi'(0) = 0$$

and

$$\begin{aligned} \Phi''(u) &= \frac{(1 - p + pe^u)pe^u - pe^u pe^u}{(1 - p + pe^u)^2} = \frac{(1-p)pe^u}{(1 - p + pe^u)^2} \\ &\leq \frac{(1-p)pe^u}{4(1-p)pe^u} = \frac{1}{4} \end{aligned}$$

we get, for any $u > 0$,

$$\Phi(u) = \Phi(0) + \Phi'(0)u + \frac{1}{2}\Phi''(\eta)u^2 \leq \frac{1}{8}u^2$$

for some $\eta \in [0, u]$. We conclude

$$\mathbf{E}\{\exp(sY)\} \leq e^{\Phi(s(b-a))} \leq \exp\left(\frac{1}{8}s^2(b-a)^2\right),$$

which proves (B.2). □

Theorem B.1 (Binomial bound, c.f. Problem 5.8 in Gallager (1968)). *For $1 \leq k \leq n-1, n \geq 2$,*

$$\sqrt{\frac{n}{8k(n-k)}} \leq \binom{n}{k} 2^{-n\mathrm{h}\frac{k}{n}} < \sqrt{\frac{n}{2\pi k(n-k)}},$$

where

$$h(x) = -x\log x - (1-x)\log(1-x)$$

is the binary entropy function.

Proof. For the proof, we will use the Stirling formula, (c.f. Feller (1968) pp. 53.) which is

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \exp\left(\varepsilon_n\right),$$

where ε_n is a decreasing sequence, which satisfies $0 < \varepsilon_n < \frac{1}{12n}$. Using this,

$$\begin{aligned}\binom{n}{k} 2^{-n\mathrm{h}\frac{k}{n}} &= \frac{n!}{k!(n-k)!} 2^{k\log\frac{k}{n} + (n-k)\log\frac{n-k}{n}} \\ &= \sqrt{\frac{n}{2\pi k(n-k)}} \frac{\exp\left(\varepsilon_n\right)}{\exp\left(\varepsilon_k + \varepsilon_{n-k}\right)}.\end{aligned}$$

For the upper bound, use that ε_n is decreasing, so $\varepsilon_k + \varepsilon_{n-k} > \varepsilon_n > 0$, thus

$$\binom{n}{k} 2^{-n\mathrm{h}\frac{k}{n}} < \sqrt{\frac{n}{2\pi k(n-k)}}.$$

For the lower bound, use $0 < \varepsilon_n < \frac{1}{12n}$:

$$\frac{\exp\left(\varepsilon_n\right)}{\exp\left(\varepsilon_k + \varepsilon_{n-k}\right)} \geq \frac{1}{\exp\left(\frac{1}{12k} + \frac{1}{12(n-k)}\right)} = \exp\left(-\frac{n}{12k(n-k)}\right).$$

Since $1 \leq k \leq n-1$, we have $k(n-k) \geq n-1$. Thus for $n \geq 4$,

$$\frac{n}{k(n-k)} \leq \frac{n}{n-1} \leq \frac{4}{3},$$

and then

$$\exp\left(-\frac{n}{12k(n-k)}\right) \geq \exp\left(-\frac{1}{9}\right) \approx 0.8948.$$

Since $\frac{\sqrt{\pi}}{2} \approx .8862$, we have that for $n \geq 4$,

$$\binom{n}{k} 2^{-n\mathrm{h}\frac{k}{n}} = \sqrt{\frac{n}{2\pi k(n-k)}} \frac{\exp\left(\varepsilon_n\right)}{\exp\left(\varepsilon_k + \varepsilon_{n-k}\right)} \geq \sqrt{\frac{n}{8k(n-k)}}.$$

For $(n,k) \in \big\{(2,1),(3,1),(3,2),(4,1),(4,2),(4,3)\big\}$, we can verify the same bound by direct calculations:

n	k	$\binom{n}{k}2^{-n\mathrm{h}\frac{k}{n}}$	$\sqrt{\frac{n}{8k(n-k)}}$
2	1	0.5	0.5
3	1 or 2	0.4444	0.4330
4	1 or 3	0.4219	0.4082
4	2	0.3750	0.3536

$\square$

Theorem B.2. *For the probability density functions $f(x)$ and $g(x)$, we have that for the I-divergence,*

$$D(f,g) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)}\, dx \geq 0.$$

Proof. Let X be a random variable with probability density function $f(x)$, and let $Y = \frac{g(X)}{f(X)}$. Then from the Jensen inequality,

$$\mathbf{E}\left(-\log Y\right) \geq -\log \mathbf{E}\left(Y\right),$$

thus

$$\begin{aligned}
D(f,g) &= \int_{-\infty}^{\infty} f(x) \log \frac{f(X)}{g(x)} \,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} f(x) \left(-\log \frac{g(x)}{f(x)} \right) \,\mathrm{d}x \\
&= \mathbf{E}\left(-\log Y \right) \\
&\geq -\log \mathbf{E}\left(Y \right) \\
&= -\log \int_{-\infty}^{\infty} f(x) \frac{g(x)}{f(x)} \,\mathrm{d}x \\
&= -\log \int_{-\infty}^{\infty} g(x) \,\mathrm{d}x \\
&= 0.
\end{aligned}$$

□

Theorem B.3. *(Discrete Entropy Bound) If X is an integer valued random variable, then*

$$\mathbf{H}\left(X\right) \leq \frac{1}{2} \log \left(2\pi e \left(\mathit{Var}\left(X\right) + \frac{1}{12} \right) \right).$$

This theorem was developed independently by Massey (unpublished), Willems (unpublished) and D'yachkov (1977). Cover and Thomas (1991) published it in their book, and later Mow (1998) extended it by eliminating the factor $\frac{1}{12}$ for some cases.

Proof. Put $p_i = \mathbf{P}\left(X = i\right)$, and let U be a continuous random variable uniformly distributed over $(0,1)$ independent of X. Define $Y = X + U$, which has the density

$$f(x) = p_i \text{ if } x \in (i, i+1).$$

Then for the entropy of X,

$$\begin{aligned}
\mathbf{H}\left(X\right) &= -\sum_{i=-\infty}^{\infty} \mathbf{P}\left(X = i\right) \log \mathbf{P}\left(X = i\right) \\
&= -\sum_{i=-\infty}^{\infty} \left(\int_{i}^{i+1} f(y) \,\mathrm{d}y \right) \log \mathbf{P}\left(X = i\right) \\
&= -\sum_{i=-\infty}^{\infty} \int_{i}^{i+1} f(y) \log \mathbf{P}\left(X = i\right) \,\mathrm{d}y
\end{aligned}$$

$$\begin{aligned}
&= -\sum_{i=-\infty}^{\infty} \int_i^{i+1} f(y) \log f(y) \, \mathrm{d}y \\
&= -\int_{-\infty}^{\infty} f(y) \log f(y) \, \mathrm{d}y.
\end{aligned}$$

This formula is the differential entropy of Y, which is maximized over all distributions with the same variance, by the Gaussian distribution. To see this, let $g(x)$ be the distribution of the normal distribution with zero mean and variance $\mathrm{Var}Y$:

$$\begin{gathered}
g(x) = \frac{1}{\sqrt{2\pi \mathrm{Var}Y}} \exp\left(-\frac{x^2}{2\mathrm{Var}Y}\right), \\
\int_{-\infty}^{\infty} f(x) \, \mathrm{d}x = \int_{-\infty}^{\infty} g(x) \, \mathrm{d}x = 1, \\
\int_{-\infty}^{\infty} f(x) x^2 \, \mathrm{d}x = \int_{-\infty}^{\infty} g(x) x^2 \, \mathrm{d}x = \mathrm{Var}Y.
\end{gathered}$$

Thus we have

$$\begin{aligned}
&-\int_{-\infty}^{\infty} g(x) \log g(x) \, \mathrm{d}x \\
&\qquad = -\int_{-\infty}^{\infty} g(x) \left(\log \frac{1}{\sqrt{2\pi \mathrm{Var}Y}} - \frac{x^2}{2\mathrm{Var}Y} \log e\right) \mathrm{d}x \\
&\qquad = -\log \frac{1}{\sqrt{2\pi \mathrm{Var}Y}} \int_{-\infty}^{\infty} g(x) \, \mathrm{d}x + \frac{\log e}{2\mathrm{Var}Y} \int_{-\infty}^{\infty} g(x) x^2 \, \mathrm{d}x \\
&\qquad = -\log \frac{1}{\sqrt{2\pi \mathrm{Var}Y}} \int_{-\infty}^{\infty} f(x) \, \mathrm{d}x + \frac{\log e}{2\mathrm{Var}Y} \int_{-\infty}^{\infty} f(x) x^2 \, \mathrm{d}x \\
&\qquad = -\int_{-\infty}^{\infty} f(x) \left(\log \frac{1}{\sqrt{2\pi \mathrm{Var}Y}} - \frac{x^2}{2\mathrm{Var}Y} \log e\right) \mathrm{d}x \\
&\qquad = -\int_{-\infty}^{\infty} f(x) \log g(x) \, \mathrm{d}x.
\end{aligned}$$

Subtracting $-\int_{-\infty}^{\infty} f(x) \log f(x) \, \mathrm{d}x$ yields

$$-\int_{-\infty}^{\infty} g(x) \log g(x) \, \mathrm{d}x + \int_{-\infty}^{\infty} f(x) \log f(x) \, \mathrm{d}x$$

$$\begin{aligned}
&= -\int_{-\infty}^{\infty} f(x)\log g(x)\,\mathrm{d}x + \int_{-\infty}^{\infty} f(x)\log f(x)\,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} f(x)\log\frac{f(x)}{g(x)}\,\mathrm{d}x \\
&\geq 0,
\end{aligned}$$

by Theorem B.2. This means, that

$$-\int_{-\infty}^{\infty} g(x)\log g(x)\,\mathrm{d}x \geq -\int_{-\infty}^{\infty} f(x)\log f(x)\,\mathrm{d}x.$$

The differential entropy of the normal distribution $g(x)$ can be calculated:

$$\begin{aligned}
-\int_{-\infty}^{\infty} g(x)\log g(x)\,\mathrm{d}x &= -\int_{-\infty}^{\infty} g(x)\left(\log\frac{1}{\sqrt{2\pi\mathrm{Var}Y}} - \frac{x^2}{2\mathrm{Var}Y}\log e\right)\mathrm{d}x \\
&= -\log\frac{1}{\sqrt{2\pi\mathrm{Var}Y}} + \frac{\log e}{2\mathrm{Var}Y}\int_{-\infty}^{\infty} g(x)x^2\,\mathrm{d}x \\
&= \frac{1}{2}\log 2\pi\mathrm{Var}Y + \frac{1}{2}\log e \\
&= \frac{1}{2}\log 2\pi e\mathrm{Var}Y.
\end{aligned}$$

Here we substitute $\mathrm{Var}(Y) = \mathrm{Var}(X) + \mathrm{Var}(U) = \mathrm{Var}(X) + \frac{1}{12}$, and conclude that

$$\mathbf{H}(X) = -\int_{-\infty}^{\infty} f(x)\log f(x)\,\mathrm{d}x \leq \frac{1}{2}\log\left(2\pi e\left(\mathrm{Var}(X) + \frac{1}{12}\right)\right).$$

□

B.2 Markov Chains

Let $\{X_i\} = \{X_0, X_1, \ldots, X_n, \ldots\}$ be a stochastic process with discrete time parameter and discrete state space $S = \{0, 1, 2, \ldots\}$.

Definition B.1. *The process $\{X_i\}$ is called a* **Markov chain** *if it holds the Markov property, i.e., if for all $n \geq 1$ and $x_0, x_1, \ldots, x_{n-1}, x_n \in S$ we have*

$$\mathbf{P}\{X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0\} = \mathbf{P}\{X_n = x_n | X_{n-1} = x_{n-1}\},$$

whenever the conditional probabilities exist.

In general, the transition probability $\mathbf{P}\{X_n = j|X_{n-1} = i\}$ depends not only on i and j, but is a function of the time parameter n, too.

Definition B.2. *If* $\mathbf{P}\{X_n = j|X_{n-1} = i\}$ *doesn't depend on* n*, then the Markov chain* $\{X_i\}$ *is called* **homogeneous**.

From this point we will deal only with homogeneous Markov chains.

$$p_{ij} = \mathbf{P}\{X_1 = j|X_0 = i\} \quad (i, j \in S)$$

denote the one-step transition probability from state i to state j and let

$$\Pi = \{p_{ij}\}.$$

Π is called the one-step transition probability matrix or transition matrix of the chain. Let the initial probability of state i be denoted by $p_i^{(0)}$, i. e.,

$$p_i^{(0)} = \mathbf{P}\{X_0 = i\} \quad (i \in S).$$

We can represent the initial distribution of $\{X_i\}$ as a row vector

$$P^{(0)} = \left\{p_i^{(0)}\right\} = \left\{p_0^{(0)}\ p_1^{(0)}\ p_2^{(0)} \ldots\right\}.$$

If

$$p_{ij}^{(n)} = \mathbf{P}\{X_n = j|X_0 = i\}$$

$(n \geq 1)$ denotes the n-step transition probability from i to j and

$$\Pi^{(n)} = \left\{p_{ij}^{(n)}\right\}$$

is the n-step transition probability matrix, moreover the marginal distribution of X_n is the row vector

$$P^{(n)} = \left\{p_0^{(n)}\ p_1^{(n)}\ \ldots\right\},$$

where

$$p_i^{(n)} = \mathbf{P}\{X_n = i\}$$

then it's easy to see by induction that

$$\Pi^{(n)} = \Pi^n \tag{B.3}$$

and

$$P^{(n)} = P^{(0)}\Pi^{(n)} = P^{(0)}\Pi^n. \tag{B.4}$$

The most important problem concerning a Markov chain is the long term behavior of the process. Henceforth we would like to characterize those Markov chains for which the distribution $P^{(n)}$ has a limit $P^{(\infty)}$ (i. e., for all $j \in S$ the sequence $\{p_j^{(n)}\}$ is convergent), $P^{(\infty)}$ is a probability distribution and is independent from the initial distribution.

Definition B.3. *If the limit*

$$\lim_{n\to\infty} P^{(n)} = P^{(\infty)}$$

exists, $P^{(\infty)}$ is a probability distribution and is independent from the initial distribution $P^{(0)}$ then the Markov chain $\{X_i\}$ is called **stable** *with limit distribution $P^{(\infty)}$.*

In the literature stable chains are named as ergodic ones.

In case $P^{(\infty)}$ exists and is independent from $P^{(0)}$ let $p_j \quad (j = 0, 1, 2, \ldots)$ denote the $(j+1)$th element of $P^{(\infty)}$, i. e., let

$$p_j = \lim_{n\to\infty} p_j^{(n)}.$$

Definition B.4. *The Markov chain $\{X_i\}$ is called* **irreducible** *if any of its states can be reached from any other state through some transitions, i. e., for any pair $i, j \in S$ there exists $n_{ij} > 0$ with $p_{ij}^{(n_{ij})} > 0$.*

Definition B.5. A state $i \in S$ of a Markov chain is **aperiodic** if there exists $n_i \geq 1$ such that for all $n \geq n_i$ we have $p_{ii}^{(n)} > 0$.

We remark that an equivalent definition of aperiodicity of $i \in S$ is

$$\gcd\{n \geq 1 : p_{ii}^{(n)} > 0\} = 1,$$

where gcd stands for the greatest common divisor. It's trivial that our definition implies the latter one, the converse requires some number theoretic considerations. Observe that if $p_{ii} > 0$ then $p_{ii}^{(n)} > 0$ for all $n \geq 1$, thus i is aperiodic.

Definition B.6. A Markov chain $\{X_i\}$ is called **aperiodic** if all of its states are aperiodic.

It is easy to see that if an irreducible Markov chain $\{X_i\}$ has an aperiodic state then $\{X_i\}$ is aperiodic.

In general the irreducibility or aperiodicity of a chain can be decided without much effort. One can show, for example, that if for all $i, j \in S$ with $|i - j| \leq 1$ we have $p_{ij} > 0$, then the chain is irreducible and aperiodic. The condition says that the central three diagonals of the transition matrix Π contain positive elements. Moreover, if for every pair $i, j \in S$ with $|i - j| = 1$ there is an $n_{ij} > 0$ such that $p_{ij}^{(n_{ij})} > 0$, then the chain is irreducible.

Theorem B.4 (Foster's criterion). *Let $\{X_i\}$ be an irreducible and aperiodic Markov chain. If there exist $I \geq 0$, $C > 0$ and $d > 0$ such that for $k \leq I$*

$$\mathbf{E}\{X_{n+1}|X_n = k\} \leq C \tag{B.5}$$

and for $k > I$

$$\mathbf{E}\{X_{n+1}|X_n = k\} \leq k - d, \tag{B.6}$$

then $\{X_i\}$ is stable.

As an application of Theorem B.6 consider a discrete time queueing with random service rate V_n, and denote by Y_n the number of arrivals in time slot n. Let the initial length of the queue Q_0 be arbitrary non-negative integer valued random variable independent of $\{V_n\}$ and $\{Y_n\}$. Then the queue length $\{Q_n\}$ is according to the following evolution:

$$Q_{n+1} = (Q_n - V_{n+1})^+ + Y_{n+1}$$

for $n \geq 0$.

Theorem B.5. *(Loynes (1962), Györfi and Morvai (2002)) If the sequences $\{V_n\}$ and $\{Y_n\}$ are independent and identically distributed, and they are independent of each other, and $\mathbf{E}\{Y_n\} < \mathbf{E}\{V_n\}$, then $\{Q_n\}$ is a stable Markov chain.*

The waiting time follows a similar evolution. According to Lindley (1952) consider the extension of the G/G/1 model. Let W_n be the waiting time of the n-th arrival, S_n be the service time of the n-th arrival, and T_{n+1} be the inter arrival time between the $(n+1)$-th and n-th arrivals. Let W_0 be an arbitrary random variable independent of the arrivals and services. Then the waiting time W_n of the n-th arrival can be calculated by

$$W_{n+1} = (W_n - T_{n+1} + S_n)^+$$

for $n \geq 0$.

Theorem B.6. *(Lindley (1952), Loynes (1962), Feller (1968), Györfi and Morvai (2002)) If $\{S_{i-1} - T_i\}$ is independent and identically distributed, $\mathbf{E}\{S_{i-1}\} < \mathbf{E}\{T_i\}$, then $\{W_i\}$ is a stable Markov process.*

Proof. Put

$$Z_n = S_{n-1} - T_n,$$

then the recursion of the waiting times is of the form

$$W_{n+1} = (W_n + Z_{n+1})^+$$

for $n \geq 0$, where $\{Z_i\}$ is a sequence of independent and identically distributed (i.i.d.) random variables with $\mathbf{E}\{Z_i\} < 0$. Introduce the notations

$$\begin{aligned} V_0 &= 0, \\ V_n &= \sum_{i=0}^{n-1} Z_{-i}, \ (n \geq 1). \end{aligned}$$

We prove that there is a stationary and Markov $\{W'_i\}$ and an almost surely finite random variable τ such that

$$W'_0 = \sup_{n \geq 0} V_n,$$

and

$$W'_n = W_n$$

for $n > \tau$.

Step 1. Fix an integer $N > 0$. Let $W_{-N,-N} = 0$ and define $W_{-N,n}$ for $n > -N$ by the following recursion,

$$W_{-N,n+1} = (W_{-N,n} + Z_{n+1})^+ \ \text{ for } n \geq -N.$$

We show that $W_{-N,0}$ is increasing in N, and almost surely,

$$\lim_{N \to \infty} W_{-N,0} = W',$$

where

$$W' = \sup_{n \geq 0} V_n,$$

and W' is finite a.s.

Notice that $W_{-N,n+1} = (W_{-N,n} + Z_{n+1})^+$ for $n \geq -N$. First we prove that for $n > -N$,

$$W_{-N,n} = \max\{0, Z_n, Z_n + Z_{n-1}, \ldots, Z_n + \cdots + Z_{-N+1}\}. \tag{B.7}$$

For $n = -N + 1$,

$$\begin{aligned} W_{-N,-N+1} &= (W_{-N,-N} + Z_{-N+1})^+ \\ &= (Z_{-N+1})^+ \\ &= \max\{0, Z_{-N+1}\}. \end{aligned}$$

For $n = -N+2$,

$$\begin{aligned} W_{-N,-N+2} &= (W_{-N,-N+1} + Z_{-N+2})^+ \\ &= \max\{0, W_{-N,-N+1} + Z_{-N+2}\} \\ &= \max\{0, \max(0, Z_{-N+1}) + Z_{-N+2}\} \\ &= \max\{0, Z_{-N+2}, Z_{-N+2} + Z_{-N+1}\}. \end{aligned}$$

Now we proceed by induction from n to $n+1$.

$$\begin{aligned} W_{-N,n+1} &= (W_{-N,n} + Z_{n+1})^+ \\ &= \max\{0, \max\{0, Z_n, Z_n + Z_{n-1}, \dots, Z_n + \cdots + Z_{-N+1}\} + Z_{n+1}\} \\ &= \max\{0, Z_{n+1}, Z_{n+1} + Z_n, \dots, Z_{n+1} + \cdots + Z_{-N+1}\}. \end{aligned}$$

We have completed the proof of (B.7). Thus

$$W_{-N,0} = \max\{0, Z_0, Z_0 + Z_{-1}, \dots, Z_0 + \cdots + Z_{-N+1}\},$$

which implies that $W_{-N,0}$ is increasing, since the maximum is taken over larger and larger set. It remains to prove that $W_{-N,0}$ converges to a random variable W' which is finite a.s.. Now by the strong law of large numbers for i.i.d. sequences, a.s.

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=-N+1}^{0} Z_i = \mathbf{E} Z_1 < 0,$$

hence a.s.

$$\lim_{N\to\infty} \sum_{i=-N+1}^{0} Z_i = -\infty.$$

We got that there is a random variable τ such that for all $i > 0$

$$\infty > W_{-\tau,0} = W_{-\tau-i,0},$$

and therefore

$$W' = \sup_{n\geq 0} V_n.$$

Step 2. Put

$$W'_0 = W'$$

and for $n \geq 0$,

$$W'_{n+1} = (W'_n + Z_{n+1})^+.$$

$\{W'_i\}$ is Markov, we show that $\{W'_i\}$ is stationary. For any sequence $z_{-\infty}^{\infty} = (\dots, z_{-1}, z_0, z_1, \dots)$ put

$$F(z_{-\infty}^{\infty}) = \lim_{N\to\infty} \max\{0, z_0, z_0 + z_{-1}, \dots, z_0 + \dots + z_{-N}\}.$$

Then by Step 1

$$W'_0 = W' = F(Z_{-\infty}^{\infty}).$$

We proceed by induction that for $n \geq 0$,

$$W'_n = F(T^n Z_{-\infty}^{\infty}),$$

where T is the left shift. For $n = 1$,

$$\begin{aligned}
F(TZ_{-\infty}^{\infty}) &= \lim_{N\to\infty} \max\{0, Z_1, Z_1 + Z_0, \dots, Z_1 + Z_0 + \dots, Z_{-N+1}\} \\
&= (\lim_{N\to\infty} \max\{0, \max\{0, Z_0, \dots, Z_0 + \dots, Z_{-N+2}\} + Z_1\}) \\
&= (\max\{0, [\lim_{N\to\infty} \max\{0, Z_0, \dots, Z_0 + \dots, Z_{-N+2}\}] + Z_1\}) \\
&= (W'_0 + Z_1)^+ \\
&= W'_1.
\end{aligned}$$

Now we prove from n to $n+1$.

$$\begin{aligned}
W'_{n+1} &= (W'_n + Z_{n+1})^+ \\
&= (F(T^n Z_{-\infty}^{\infty}) + Z_{n+1})^+ \\
&= (\lim_{N\to\infty} \max\{0, Z_n, Z_n + Z_{n-1}, \dots, Z_n + Z_{n-1} + \dots, Z_{n-N}\} + Z_{n+1})^+ \\
&= \lim_{N\to\infty} \max\{0, Z_{n+1}, Z_{n+1} + Z_n, \dots, Z_{n+1} + Z_n + \dots, Z_{n+1-N}\} \\
&= F(T^{n+1} Z_{-\infty}^{\infty}).
\end{aligned}$$

$\{Z_i\}$ is stationary, therefore $\{W'_i\}$ is stationary, too.
Step 3. Similarly to the proof of Step 1,

$$W_n = \max\{0, Z_n, Z_n + Z_{n-1}, \dots, Z_n + \dots + Z_1, Z_n + \dots + Z_1 + W_0\},$$

and

$$W'_n = \max\{0, Z_n, Z_n + Z_{n-1}, \dots, Z_n + \dots + Z_1, Z_n + \dots + Z_1 + W'_0\}.$$

But for large n, both

$$Z_n + \dots + Z_1 + W_0 < 0$$

and

$$Z_n + \cdots + Z_1 + W_0' < 0,$$

and so

$$W_n = W_n' = \max\{0, Z_n, Z_n + Z_{n-1}, \ldots, Z_n + \cdots + Z_1\}.$$

□

Bibliography

A, N. Q. (1986). *Some coding problems of multiple-access communication systems.* DSc dissertation, Hungarian Academy of Sciences.

A, N. Q., Györfi, L., and Massey, J. L. (1992). Constructions of binary constant-weight cyclic codes and cyclically permutable codes. *IEEE Trans. on Information Theory*, 38:940–949.

A, N. Q. and Zeisel, T. (1988). Bounds on constant weight binary superimposed codes. *Problems of Control and Information Theory*, 17(4):223–230.

Abramson, N. (1970). The aloha system – another alternative for computer communication. In *Proc. 1970 Fall Joint Computer Conference*, pages 281–285. AFIPS Press.

Abramson, N. (1985). Development of the alohanet. *IEEE Trans. on Information Theory*, IT–31:119–123.

Ahlswede, R. (1971). Multi–way communication channels. In *Proceedings of the 2nd International Symposium on Information Theory*, pages 23–52. Hungarian Academy of Sciences.

Bassalygo, L. A. and Pinsker, M. S. (1983). Limited multiple-access of a nonsynchronous channel. *Problems of Information Transmission*, 19(4):92–96.

Berlekamp, E. R. and Justesen, J. (1974). Some long cyclic linear binary codes are not so bad. *IEEE Transactions on Information Theory*, 20(3):351–356.

Bernstein, S. N. (1946). *The Theory of Probabilities.* Gastehizdat Publishing House, Moscow.

Blahut, R. E. (1984). *Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley.

Bose, R. C. and Chowla, S. (1962). Theorems in the additive theory of numbers. *Commentarii Mathematici Helvetici*, 37:141–147.

Brooks, R. (1941). On coloring the nodes of a network. *Proceedings of the Cambridge Philosophical Society*, 37:194–198.

Burton and Weldon (1965). Cyclically permutable error-correcting codes. *IEEE Trans. Information Theory*, 11:433–439.

Cantor, D. and Mills, W. (1966). Determination of a subset from certain combinatorial properties. *Canadian Journal of Mathematics*, 18:42–48.

Capetanakis, J. I. (1979). Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, IT-25(5):505–515.

Chang, S. C. and Weldon, E. J. (1979). Coding for T-user multiple-access channels. *IEEE Transactions on Information Theory*, IT-25(6):684–691.

Chernoff, H. (1952). A measure of asymtotic efficiency of tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507.

Chien, R. T. and Frazer, W. D. (1966). An application of coding theory to document retrieval. *IEEE Transactions on Information Theory*, 12(2):92–96.

Cohen, A. R., Heller, J. A., and Viterbi, A. J. (1971). A new coding technique for asynchronous multiple access communication. *IEEE Transactions on Communication Technology*, 19:849–855.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

da Rocha Jr., V. C. (1984). Maximum distance separable multilevel codes. *IEEE Transactions on Information Theory*, 30(3):547–548.

Dorfman, R. (1943). The detection of defective members of large populations. *Annals of Mathematical Statistics*, 14:436–440.

Dostert, K. (2001). *Powerline Communications*. Prentice Hall PTR.

Du, D.-Z. and Hwang, F. K. (1993). *Combinatorial Group Testing and Its Applications*. Singapore, NJ/London/Hong Kong: World Scientific.

Dyachkov, A. G. (1977). On a seach model of false coins. In *Topics in Information Theory*, Csiszr, I. and Elias, P., editors, pages 163–170. North Holland, Amsterdam.

Dyachkov, A. G., Macula, A. J., and Rykov, V. V. (2000). New constructions of superimposed codes. *IEEE Transactions on Information Theory*, 46(1):284–290.

Dyachkov, A. G. and Rykov, V. V. (1981). A coding model for a multiple-access adder channel. *Problems of Information Transmission*, 17(2):26–38.

Dyachkov, A. G. and Rykov, V. V. (1982). Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3):7–13.

Dyachkov, A. G. and Rykov, V. V. (1983). A survey of superimposed code theory. *Problems of Control and Information Theory*, 12(4):3–12.

Einarsson, G. (1980). Address assignment for a time-frequency-coded, spread-spectrum system. *Bell System Technical Journal*, 59(7):1241–1255.

Einarsson, G. and Vajda, I. (1987). Code acquisition for a frequency-hopping system. *IEEE Transactions on Communications*, 35(5):566–568.

Erdős, P., Frankl, P., and Füredi, Z. (1985). Families of finite sets in which no set is covered by the union of r others. *Israel Journal of Mathematics*, 51(1-2):79–89.

Erdős, P. and Rényi, A. (1963). On two problems of information theory. *Publications of the Mathematical Institute of the Hungarian Academy of Science*, 8:229–243.

Ericson, T. and Györfi, L. (1988). Superimposed codes in R^n. *IEEE Transactions on Information Theory*, IT-34(4):877–880.

Ericson, T. and Zinoviev, V. A. (1987). An improvement of the gilbert bound for constant weight codes. *IEEE Transactions on Information Theory*, 33(5):721–723.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, New York, 3rd edition.

Flajolet, P., Gourdon, X., and Dumas, P. (1995). Mellin Transforms and Asymptotics: Harmonic sums. *Theoretical Computer Science*, 144:3–58.

Frankl, P. (1976). On Sperner families satisfying an additional condition. *Journal of Combinatorial Theory Series A*, 20:1–11.

Füredi, Z. (1996). On r-cover-free families. *Journal of Combinatorial Theory Series A*, 73:172–173.

Füredi, Z. and Ruszinkó, M. (1999). An improved upper bound of the rate of euclidean superimposed codes. *IEEE Transactions on Information Theory*, IT-45(2):799–802.

Gallager, R. G. (1968). *Information Theory and Reliable Communication.* John Wiley and Sons. Page 530, Problem 5.8.

Gallager, R. G. (1978). Conflict resolution in random access broadcast networks. In *Proc. AFOSR Workshop Communication Theory and applications*, pages 74–76. Provincetown.

Gallager, R. G. (1985). A perspective on multiaccess channels. *IEEE Transactions on Information Theory*, IT-31(2):124–142.

Gilbert, E. N. (1963). Cyclically permutable error-correcting codes. *IEEE Trans. Information Theory*, 9:175–182.

Gulko, E. and Kaplan, M. (1985). Analytic properties of multiple-access trees. *IEEE Transactions on Information Theory*, 31(2):255–263.

Győri, S. (2003). Signature coding over multiple access OR channel. In *Proceedings of the IEEE Information Theory Workshop*, pages 115–118. La Sorbonne, Paris, France.

Győri, S. (2004). Performance evaluation of the Kautz–Singleton code for random activity. In *Proceedings of the IASTED International Conference on Communication Systems and Applications*, pages 285–289. Banff, AB, Canada.

Györfi, L. and Győri, S. (2004). Bounds for multiple-access collision channel. In *Proceedings of the International Symposium on Information Theory and its Applications*, pages 1381–1386. Parma, Italy.

Györfi, L. and Győri, S. (2005). Analysis of collision channel with asynchronous access. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E88(9).

Györfi, L., Jordán, A., and Vajda, I. (2000). Exact error probability for slow frequency hopping. *European Transactions on Telecommunications*, 11(2):183–190.

Györfi, L. and Morvai, G. (2002). Queueing for ergodic arrivals and services. In *Limit Theorems in Probability and Statistics*, Berkes, I., Csáki, E., and Csörgő, M., editors, pages 127–141. J. Bolyai Math. Society, Budapest.

Györfi, L. and Vajda, I. (1993). Constructions of protocol sequences for multiple access collision channel without feedback. *IEEE Transactions on Information Theory*, 39(5):1762–1765.

Hajek, B. (1980). *Expected number of slots needed for the Capetanakis collision-resolution algorithm.* unpublished manuscript, Coordinated Sci. Lab., Univ. of Illinois, Urbana IL.

Hajek, B. and van Loon, T. (1982). Decetralized dynamic control of a multiaccess broadcast channel. *IEEE Trans. on Automatic Control*, AC–27:559–569.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30.

Hwang, F. K. (1972). A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67:605–608.

Hwang, F. K. and Sós, V. T. (1987). Non-adaptive hypergeometric group testing. *Studia Scientiarum Mathematicarum Hungarica*, 22:257–263.

Jacquet, P. and Regnier, M. (1986). Trie partitioning process: limiting distributions. In *Proceedings of the 11th colloquium on trees in algebra and programming*, pages 196–210. Springer-Verlag, New York.

Janssen, A. J. E. M. and de Jong, M. J. M. (2000). Analysis of contention tree algorithms. *IEEE Trans. Information Theory*, 46:2163–2172.

Kaplan, M. (1979). A sufficient condition for non-ergodicity of a markov chain. *IEEE Trans. on Information Theory*, IT–25:470–471.

Kautz, W. H. and Singleton, R. C. (1964). Nonrandom binary superimposed codes. *IEEE Trans. Information Theory*, IT-10:363–377.

Khachatrian, G. H. (2000). A survey of coding methods for the adder channel. In *Numbers, Information and Complexity*, Althöfer, I., editor, pages 181–196. Kluwer Academic Publishers.

Knill, E., Bruno, W. J., and Torney, D. C. (1998). Non-adaptive group testing in the presense of error. *Discrete Applied Mathematics*, 88:261–290.

Knuth, D. E. (1973). *The Art of Computer Programming*, volume 3. Addison-Wesley, Reading MA.

Liao, H. (1972). A coding theorem for multiple access communications. In *Proceedings of the International Symposium on Information Theory.*

Lidl, R. and Niederreiter, H. (1986). *Introduction to Finite Fields and their Applications.* Cambridge University Press.

Lindley, D. V. (1952). The theory of queues with a single server. *Proc. Cambridge Philos. Soc.*, 48:277–289.

Lindström, B. (1964). On a combinatory detection problem I. *Publications of the Mathematical Institute of the Hungarian Academy of Science*, 9:195–207.

Lindström, B. (1975). Determining subsets by unramified experiments. In *A Survey of Statistical Designs and Linear Models*, Srivastava, J., editor, pages 407–418. North Holland Publishing Company.

Loynes, R. M. (1962). The stability of a queue with non-independent inter-arrival and service times. *Proc. Cambridge Philos. Soc.*, 58:497–520.

Maracle, D. E. and Wolverton, C. T. (1974). Generating cyclically permutable codes. *IEEE Trans. Information Theory*, 20:554–555.

Massey, J. L. (1981). Collision-resolution algorithms and random-access communications. In *Multi-User Communications Systems*, Longo, G., editor, pages 73–137. Springer.

Massey, J. L. and Mathys, P. (1985). The collision channel without feedback. *IEEE Trans. on Information Theory*, IT-31:192–204.

Mathys, P. (1984). *Analysis of random-access algorithms.* PhD dissertation, Dept. of Elec. Engr., Eidgenössische Technische Hochschule, Zürich.

Mathys, P. and Flajolet, P. (1985). q-ary collision resolution algorithms in random-access systems with free or blocked channel access. *IEEE Trans. on Information Theory*, IT–31:119–123.

Mow, W. H. (1998). A tight upper bound on discrete entropy. *IEEE Transactions on Information Theory*, IT-44(2):775–778.

Neumann, P. G. (1964). On a class of cyclically permutable error-correcting codes. *IEEE Trans. Information Theory*, 10:75–78.

Pippenger, N. (1981). Bounds on the performance of protocols for a multiple-access broadcast channel. *IEEE Transactions on Information Theory*, IT-27(2):145–151.

Pursley, M. B. (1987). Spread spectrum in packet radio networks. *Proc. IEEE*, 75:116–134.

Roberts, L. G. (1975). Aloha packet system with and without slots and capture. *Computer Communications Review*, 5:28–42.

Rödl, V. (1985). On a packing and covering problem. *European Journal of Combinatorics*, 5:69–78.

Ruszinkó, M. (1994). Note on the upper bound of the size of the r-cover-free families. *Journal of Combinatorial Theory Series A*, 66(2):302–310.

Ruszinkó, M. and Vanroose, P. (1997). How an Erdős-Rényi-type search approach gives an explicit code construction of rate 1 for random access with multiplicity feedback. *IEEE Trans. on Information Theory*, 43:368–373.

Shannon, C. E. (1961). Two–way communication channels. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 611–644. University of California Press.

Sobel, M. and Groll, P. A. (1959). Group testing to eliminate efficiently all defectives in a binomial sample. *Bell System Technical Journal*, 38:1178–1252.

Sperner, E. (1928). Ein satz über untermengen einer endligen menge. *Mathematische Zeitschrift*, 27:544–548.

Sterrett, A. (1957). On the detection of defective members of large populations. *Annals of Mathematical Statistics*, 28:1033–1036.

Szpankowski, W. (2001). *Average Case Analysis of Algorithms on Sequences.* Wiley, New York.

Tsybakov, B. S. and Likhanov, N. B. (1983). Packet communication on a channel without feedback. *Problems of Information Transmission*, 19(2):69–84.

Tsybakov, B. S. and Mihailov, V. A. (1978). Slotted multiaccess packet-broadcasting feedback channel. *Problemy Peredachi Informatsii*, 14:32–59.

Tsybakov, B. S. and Mihailov, V. A. (1980). Random multiple access of packets. part-and-try algorithm. *Problemy Peredachi Informatsii*, 16:65–79.

Tsybakov, B. S. and Vvedenskaya, V. A. (1980). Stack-algorithm for random multiple access. *Problemy Peredachi Informatsii*, 16:80–94.

Vajda, I. (1995). Code sequences for a frequency-hopping multiple-access systems. *IEEE Trans. on Communications*, 43:2553–2554.

Vajda, I. and Einarsson, G. (1987). Code acquisition for a frequency-hopping system. *IEEE Trans. on Communications*, COM-35:566–568.

van der Meulen, E. C. (1971). The discrete memoryless channel with two senders and one receiver. In *Proceedings of the 2nd International Symposium on Information Theory*, pages 103–135. Hungarian Academy of Sciences.

van Lint, J. H. and Springer, T. A. (1987). Generalized Reed–Solomon codes from algebraic geometry. *IEEE Transactions on Information Theory*, 33:305–309.

Wolf, J. K. (1985). Born again group testing: Multiaccess communications. *IEEE Transactions on Information Theory*, 31(2):185–191.

Zierler, N. (1959). Linear recurring sequences. *J. SIAM*, 7:31–48.

Zinoviev, V. A. (1983). Cascade equal-weight codes and maximal packings. *Problems of Control and Information Theory*, 12(1):3–10.