Budapest University of Technology and Economics
Department of Computer Science and Information Theory

Multiple Access Channels

PhD dissertation

Sándor Győri

Supervisor: Prof. László Györfi

Budapest, 2005
# Contents

1 Introduction 4

2 OR channel 11
   2.1 Channel model ........................................ 11
   2.2 Performance evaluation of the Kautz–Singleton code .......... 13
   2.3 Fast frequency hopping ................................... 22
   2.4 Asynchronous OR channel ................................... 32

3 Collision channel 40
   3.1 Channel model ........................................ 40
   3.2 Bounds for non-binary packets, synchronous access ............ 42
   3.3 Bounds for non-binary packets, asynchronous access ............ 49

4 Collision Channel with Ternary Feedback 55
   4.1 Channel model ........................................ 55
   4.2 Tree Algorithm for Collision Resolution .................... 56
   4.3 Oscillation of $L_N - L(N)$ ................................ 59

Bibliography 66
Acknowledgements

I would like to express my deepest appreciation and gratitude to my supervisor Prof. László Györfi. Without his guidance, unconditional support, scientific and ethical encouragement throughout my studies and research, this dissertation would not have been possible.

I wish to thank Bálint Laczay and Dr. András György for commenting on an earlier version of the manuscript and for the many valuable suggestions. I would also like to thank Vera Vértesi for fruitful discussions from which the dissertation has benefited greatly.

This work was sponsored by the Office of Naval Research International Field Office and the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-05-1-3017. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.
Alulírott Győri Sándor kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

A disszertáció bőrlatai és a védésről készült jegyzőkönyv megtekinthető a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karának Dékáni Hivatalában.


Győri Sándor
Chapter 1

Introduction

The usual model for multiple access communications incorporates many independent users, some small fraction of which have information to be transmitted at any given time. The problem is how to serve them, if one shared and usually expensive communications medium (e.g., a limited, common frequency band) is given. This shared medium could take any of several forms. Here are some examples of multiple access systems:

- In a mobile satellite system with a single earth-coverage beam, ground terminals transmit to the satellite using one frequency band; the satellite rebroadcasts the signal using a different frequency band. All terminals (including the sender) hear the rebroadcast after a delay which includes the two-way propagation time and any processing delays.

- In a Local Area Network (LAN), two or more computers are connected to a common bus (coaxial cable, UTP cable or fiber optic cable). If one computer transmits a message, all other computers hear the transmission after a delay.

- In a packet radio network for mobile users, the transmission of a user can be received by other users within some range which depends on the transmit power level and other factors.

Although the shared medium is quite different in the above examples, a common feature of all three systems is that when any single user transmits, many other users can receive the transmission. In a mobile satellite system, users receive all transmissions from other users. A full-duplex radio can also monitor the rebroadcast of its own transmission. In the packet radio network, in general, each user can receive the transmissions of some subset of the user population.

There are some possible scenarios for that one or more users want to communicate with one or more users, so we distinguish one-to-one, one-to-many, many-to-one and many-to-many scenarios. In mobile communications one-to-many situation is considered for the downlink channel (when the base station want to send information to the handies), and many-to-one for the uplink channel (when the handies send information to the base station). It is easy to see that the problem of many-to-many can be decomposed to the set of many-to-one, so in fact we consider just the many-to-one problem (the uplink channel in terms of mobile communications).
We must somehow split the capacity of the common channel among the users. The classical solution is a kind of multiplexing, i.e., either time-division multiplexing or frequency-division multiplexing. For partially active users, always there are a large number of users which have nothing to send most of the time. In this communications situation the multiplexing is inefficient. Moreover, these traditional methods require guarding time or frequency between bands, and a channel allocation phase. This is an overhead which wastes time or frequency, and gets larger as more users are active simultaneously or as the activity of the users becomes more random. Even if we neglect this overhead, the previously fixed, not-scalable division of resources among users makes a bottleneck in these systems.

By code division multiple access (CDMA) techniques this frontier might be broken. Time and frequency resources can be utilized in one continuous domain throughout the entire communications system. The number of users and percentage of the common medium a user can occupy is widely scalable, and there is a trade-off between these quantities.

Nowadays the most popular CDMA systems, e.g., phase modulated and direct-sequence coded channel access (which led to the standards of the 3G mobile telephony), are extremely poor in frequency utilization. Their data rates are just a small fraction of the information-theoretic capacity. This is because they have to deal with channel errors, synchronization, fading, Gaussian noise, multi-user detection, and most importantly, because we do not have a fundamental method yet to achieve full capacity. Even the mathematical bases are still quite unexplored. For the current practical solutions in mobile communications the active users initiate a login procedure during which they 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:

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

From an information theoretic viewpoint the multiple access channel is a black-box operating in discrete time with a fixed number of inputs and one output (cf. Figure 1.1). There are also extended models, with multiple outputs, the so called interference channels, but we do not deal with them now (cf. Shannon (1961) and Ahlswede (1971)). We consider that one user corresponds to each input, so instead of inputs we usually refer to users. Let us denote
the number of users by $T$. The input and output alphabets of the channel are denoted by $I$ and $O$, respectively.

In information theory one deals usually with memoryless channels. In this case to fully describe the channel, it is enough to give the channel transition probabilities:

$$p(y \mid x_1, x_2, \ldots, x_T) = P\{Y = y \mid X_1 = x_1, X_2 = x_2, \ldots, X_T = x_T\}$$

$$\forall (x_1, x_2, \ldots, x_T) \in I^T, \forall y \in O.$$

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 code words, one for each possible message of the user. We assume, that all these code words of all users have a common length $n$. So the component code of the $i$th user can be written as

$$C_i = \{x_1^i, x_2^i, \ldots, x_{|C_i|}^i\} \subseteq I^n.$$

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

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

The message user $i$ wants to send is denoted by the random variable $K_i \in \{1, 2, \ldots, |C_i|\}$. To send this message, the user transmits $x_{K_i}^i$ through the channel. We will use a further simplification, that the code words sent by the users are bit (and sometimes block) synchronized. So we can treat the channel output as a vector.

In the case of synchronous access the users are block synchronized, i.e., every user begin to send his code word at the same time. At asynchronous access the users are just bit synchronized, so their code words can be shifted relative to each other.

In multi-user information theory the users are always active, the access is block synchronous, and the channel model is an arbitrary memoryless channel. 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 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 $T$ users can be active. Moreover, usually the models allow asynchronous access.

The main problem is to find a code of the smallest possible length $n(T, M)$ such that if from the $T$ total users at most $M$ active users send their code words then from the output vector of the multiple access channel the tasks of identification, synchronization and decoding would be possible.

There is an important special case of the 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.

Next, we give some examples for application of signature coding.

▶ 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.

- **Collection of measurement 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.

- **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.

- **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 out-breaks 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.

- **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 \( j \)th individual has the binary code word (test sequence) \( x_i \) 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

\[
y = \bigvee_{i \in \{i_1, \ldots, i_m\}} x_i,
\]

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

**Structure of the dissertation.**

In this dissertation we deal with three deterministic multiple access channels:

- OR channel (Chapter 2),
- collision channel without feedback (Chapter 3),
- collision channel with ternary feedback (Chapter 4).

In Chapter 2 the problem of signature coding for multiple access OR channel is considered. In Section 2.2 performance of the most popular coding method, the Kautz–Singleton code construction is evaluated. Exact detection error probabilities are derived using a Markov-chain technique. It turns out that in practical applications detection error is much smaller than resulted by the conventional design applied so far (which is a rather simple binomial upper bound). So, it is possible to use codes of much smaller length while keeping detection error probability at small value. Both the case of fixed and binomial activity of users are studied. Signature coding for fast frequency hopping channel is investigated in Section 2.3, where the bandwidth is partitioned into frequency subbands. A frequency hopping pattern is assigned to each user that specifies the sequence of frequency subbands in which the user is allowed to transmit a sine waveform during a time slot. Partial activity is considered where only a small fraction of the potential users may be active simultaneously. We prove that in frame asynchronous case the upper bound on the minimum code length via random coding is asymptotically the same as in the case of synchronous access. Section 2.4 is devoted for signature coding via a multiple access OR channel. We prove that in block asynchronous case the upper bound on the minimum code length is asymptotically the same as in the case of synchronous access.

In Chapter 3 a multiple access collision channel without feedback is considered. The traffic is in the form of packets taking values from the input alphabet \( I \). Each user can send an arbitrary symbol (packet) from \( I \) into the channel or can be silent, too. The output of the channel can be silence, an element of \( I \), or collision. We are looking for codes and protocol sequences of users of minimum length such that from the output of the channel it can be determined which users were active and what they sent. For binary packets (\(|I| = 2\)) the problem had been already solved. Using non-binary packets commonly arises in practical
communications. Section 3.2 contains our new results on using non-binary packets ($|I| > 2$) on a collision channel. With non-binary packets the design of codes and decoding can be systematic and efficient (actually it uses a Reed–Solomon decoding which can be done by fast algorithms), while the same utilization can be reached as with binary packets. Section 3.3 deals with our new results on the block asynchronous case where the synchronization of users’ code words is also among the tasks.

In Chapter 4 a multiple access collision channel with ternary feedback is studied. For the tree algorithm, let $L_N$ denote the expected collision resolution time given the collision multiplicity $N$. $L(z)$ stands for the Poisson transform of $L_N$. Section 4.2 gives an algorithm for easy calculation of $L(z)$ for large $z$. In Section 4.3 we show that the asymptotic difference $L_N - L(N)$ does not tend to 0, but it oscillates with a small amplitude.

**Probabilistic method.**

Throughout the dissertation the so-called probabilistic method will be often applied. This is a non-constructive method pioneered by Pál Erdős, for proving the existence of a prescribed kind of mathematical object, and primarily used in combinatorics, but has acquired ever wider applications. It works by showing that if one randomly chooses objects from a specified class, the probability that the result is of the prescribed kind is more than zero. Although the proof uses probability, the final conclusion is determined for certain, without any possible error.

One way of doing this is by considering a randomly selected thing from a finite-sized universe. In our case this is a code. If the probability that the random code satisfies certain properties is greater than zero, then this proves the existence of a code that satisfies the properties. It does not matter if the probability is astronomically small; any probability strictly greater than zero will do. Also, showing that the probability is (strictly) less than 1 can be used to prove the existence of an object that does not satisfy the prescribed properties.

We would like to prove the existence of a code $C$ which is “good” in some sense, so we need

$$P\{ C \text{ is good} \} > 0.$$  

For this we prove that the probability of a randomly chosen code being bad is below 1. Generally, a code is bad if it fails any of the tasks of identification, synchronization and decoding.

- False identification means that based on the output of the channel the receiver declares a non-active user as active.
- False synchronization means that based on the output of the channel the receiver wrongly recognizes the starting slot of an active user.
- False decoding means that based on the output of the channel the receiver erroneously decodes the message sent by an active user.

$$P\{ C \text{ is bad} \} = P\{ \{ \text{false identification} \} \cup \{ \text{false synchronization} \} \cup \{ \text{false decoding} \} \} \leq P\{ \text{false identification} \} + P\{ \text{false synchronization} \} + P\{ \text{false decoding} \},$$

and we need this sum of probabilities to be less than 1, because then there exists a good code:

$$P\{ \text{false identification} \} + P\{ \text{false synchronization} \} + P\{ \text{false decoding} \} < 1 \quad (1.1)$$
It is enough if the following probabilities tend to 0:

\[
\begin{align*}
P\{\text{false identification}\} & \to 0, \\
P\{\text{false synchronization}\} & \to 0, \\
P\{\text{false decoding}\} & \to 0.
\end{align*}
\] (1.2)

If synchronous access is applied, the second probability, and if signature coding is used, the last probability should be ignored.

**Notation.**
Throughout the text \(\log(\cdot)\) means the logarithm function of base 2, while \(\ln(\cdot)\) stands for the natural logarithm function. \(h(x)\) is the binary entropy function defined as

\[
h(x) = -x \log x - (1-x) \log(1-x).
\]

We denote by \(\simeq, \gtrsim\) and \(\lesssim\) approximate equalities, lower and upper bounds, respectively, which hold asymptotically in case of some given conditions. For example

\[
f(x_1, x_2, \ldots, x_n) \lesssim g(x_1, x_2, \ldots, x_n)
\]

with conditions \(x_1 \to \infty, x_2 \to \infty, \ldots, x_n \to \infty\), if

\[
\lim_{x_1 \to \infty} \lim_{x_2 \to \infty} \cdots \lim_{x_n \to \infty} \frac{f(x_1, \ldots, x_n)}{g(x_1, \ldots, x_n)} \leq 1.
\]

The order of the variables is important. It can happen that \(g(\cdot)\) actually does not depend on some of the variables of \(f(\cdot)\), because these variables “vanish” when we take the limes.

**Remark.** Throughout the dissertation the code lengths must be integers. Putting the integer part signs would be necessary only at the end of the proofs, but would make the formulae hard to read, so they are omitted. As the results are asymptotical statements, with not too much effort it could be shown that taking the integer part of the code length would not harm the validity of lower and upper bounds.
Chapter 2

OR channel

2.1 Channel model

Cohen, Heller and Viterbi (1971) introduced the model of the noiseless OR channel for multiple access communications (cf. Figure 2.1). If there are \( T \) users in the system and \( x_i \) denotes the binary message of the \( i^{th} \) user, then the output of the channel is defined by

\[
y = \bigvee_{i=1}^{T} x_i,
\]

i.e., the output is 0 iff all inputs are 0.

A possible example of communication scheme where this simple model is suitable, is on/off keying (OOK) modulation (Sommer (1968)). The bit 1 corresponds 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.

The reader can find a survey on the multiple access OR channel in Győri (2005b), Győri (2004a), where the results in the last two decades of Eastern (Dyachkov, Erdős, Frankl, Füredi, Győrő, Rödl, Ruszinkó, Rykov, T. Sós, Vajda, Zeisel, Zinoviev) and Western researchers (Berlekamp, Dorfman, Du, Ericson, Hwang, Justesen, Linial, Sterrett, Wolf) are summarized.

For permanent activity of the users this channel is trivial, with time sharing the maximum utilization 1 can be achieved. For partial activity, however, the problem is hard and is far from being solved (Győrő and Kerekes (1981)).

In this chapter we are investigating the detection and synchronization problem which is called signature coding on multiple access OR channel, when each user has only one own code word. Moreover, in the next section frame synchronization is assumed among users (all active users begin transmitting their code words at the same time). The coding problem is to find a code 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 (detected).

**Definition 2.1 (UD code).** A code is Uniquely Decipherable of order \( M \) (UD(M)), if every Boolean sum of up to \( M \) different code words is distinct from every other sum of up to \( M \) different code words.
Formally, let $\mathcal{C} = \{x_1, x_2, \ldots, x_T\}$ be a binary code of length $n$. The UD($M$) property means that for any subsets $A, B \subset \{1, 2, \ldots, T\}$, $|A| \leq M, |B| \leq M, A \neq B$ we have
\[
\bigvee_{i \in A} x_i \neq \bigvee_{i \in B} x_i.
\]

For a given code $\mathcal{C}$, it is hard to verify the UD property, and even if we can ensure this property, the detection needs circuitous search among the Boolean sums of at most $M$ code words, therefore Kautz and Singleton (1964) introduced a less demanding concept which is a special case of the UD code.

We say that a sequence $z = (z_1, \ldots, z_n)$ covers a sequence $y = (y_1, \ldots, y_n)$, denoted by $z \geq y$, if $z_i \geq y_i, \forall i \in \{1, 2, \ldots, n\}$.

Motivated by information retrieval systems, originally Kautz and Singleton (1964) introduced the concept of ZFD codes. Much later this code property had another interpretation by applying it as signature code via a multiple access OR channel.

**Definition 2.2 (ZFD Code).** A code is Zero False Drop of order $M$ (ZFD($M$)), if every Boolean sum of up to $M$ different code words covers no code word other than those used to form the sum.

The ZFD($M$) property means that for any subset $A \subset \{1, 2, \ldots, T\}, |A| \leq M$ and for any $j \notin A$
\[
x_j \leq \bigvee_{i \in A} x_i.
\]

This provides a really fast detection rule, i.e., if $x_i \leq y$, where $y$ is the output vector of the OR channel, then the receiver decides that user $i$ is active. The other name of ZFD code is superimposed code. Obviously, a ZFD($M$) code is a UD($M$) code.

We would like to construct a binary code which can be used efficiently (with small code length) on an OR channel. If the number of all potential users $T$ is known, and it can be guaranteed that only a little fraction of them are active simultaneously ($M \ll T$), then we can use a ZFD($M$) code with error-free detection shown above.
There exist bounds on the minimum code length \( n(T,M) \) in the error-free case. Dyachkov and Rykov (1983) obtained lower bound on the code length of ZFD code for fixed large \( M \) and \( T \to \infty \):

\[
\frac{1}{2} \frac{M^2}{\log M} \log T \lesssim n(T,M).
\]

(D.1)

Dyachkov and Rykov (1983) calculated an upper bound, too, by applying random coding method of binomially distributed code words

\[
n(T,M) \lesssim e \ln 2 (M+1)^2 \log T.
\]

A and Zeisel (1988) achieved a better upper bound by applying random coding method of constant weight code words

\[
n(T,M) \lesssim \frac{1}{\ln 2} (M+1)^2 \log T.
\]

2.2 Performance evaluation of the Kautz–Singleton code for random activity

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 \( q \) be a prime power and we denote by GF\((q)\) the finite field of \( q \) elements (Galois Field). We say that a number \( \alpha \) is a primitive element in GF\((q)\) when \( m = q - 1 \) is the smallest positive integer that \( \alpha^m = 1 \). One can prove that each finite field has a primitive element, and \( 1, \alpha, \alpha^2, \ldots, \alpha^{q-2} \) are all distinct elements of the field.

We introduce now the Reed–Solomon code of parameters \((N,K)\) over GF\((q)\) \((N \leq q - 1 \text{ and } K < N)\). Let us define for each message vector \( u = (u_0, u_1, \ldots, u_{K-1}) \) \((u_i \in \text{GF}(q) \text{ for all } 0 \leq i \leq K-1)\) the message polynomial

\[
u(x) = u_0 + u_1 x + \cdots + u_{K-1} x^{K-1},
\]

then the corresponding code word \( c = (c_0, c_1, \ldots, c_{N-1}) \) is \( c_i = \nu(\alpha^i) \) for all \( 0 \leq i \leq N-1 \). If \( N = q - 1 \) then this code is called Reed–Solomon code of maximum length.

It is easy to see that the Reed–Solomon code is a \( q \)-ary linear code for which

\[
G = \begin{pmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \alpha & \alpha^2 & \cdots & \alpha^{N-1} \\
1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{K-1} & \alpha^{2(K-1)} & \cdots & \alpha^{(K-1)(N-1)}
\end{pmatrix}
\]

(2.2)
is a generator matrix. The minimum distance of the \((N,K)\) Reed–Solomon code is
\[
d_{\text{min}} = N - K + 1,
\]
i.e., the Reed–Solomon codes are MDS codes.

For the Kautz–Singleton construction, 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:
\[
0 \mapsto 0 \cdots 001, \quad \alpha^0 \mapsto 0 \cdots 010, \\
\alpha^1 \mapsto 0 \cdots 100, \\
\vdots \\
\alpha^{q-2} \mapsto 1 \cdots 000.
\]
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:
\[
M_0 = \left\lfloor \frac{N - 1}{N - d_{\text{min}}} \right\rfloor = \left\lfloor \frac{N - 1}{K - 1} \right\rfloor = \left\lfloor \frac{q - 2}{\log T / \log q - 1} \right\rfloor \quad \text{(2.3)}
\]
(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. Győrffy, 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
\[
P_e(m) = P\{V_m = w\} = P\{V_m = q - 1\}.
\]

\(|S(U_i)|\) for all \( i = 1, \ldots, m \) are independent, identically distributed random variables, and their distribution can be calculated. Introduce the notation
\[
P\{|S(U_i)| = \ell\} = p_\ell, \quad 0 \leq \ell \leq K - 1
\]

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 \(|S(U_i)| \leq K - 1\). (This is because of the MDS property of the Reed–Solomon code. As the minimum distance is \( d_{\text{min}} = N - K + 1 \), the number of identical coordinates between two code words is at most \( N - d_{\text{min}} = K - 1 \).)

\( \{V_m\} \) forms a homogeneous Markov chain on the state space \( \{0, 1, \ldots, q - 1\} \), so its distribution can be calculated in a recursive way (cf. Győrfi, Jordán and Vajda (2000)). For \( m = 1 \) we have the initial distribution of the chain:
\[
P\{V_1 = \ell\} = P\{|S(U_1)| = \ell\} = p_\ell \quad (0 \leq \ell \leq K - 1).
\]

\( \{V_m\} \) 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 \( i + k \leq K - 1 \)) from which \( i \) are out of the previously non-covered ones and \( k \) have been previously covered. If the number of previously covered positions are \( j \), 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 \( (m \geq 2) \):
\[
P\{V_m = j + i \mid V_{m-1} = j\} = \sum_{k=0}^{\min\{K-1-i,j\}} p_{i+k} \binom{w-j}{i} \binom{w}{i+k}
\]

for \( 0 \leq j, j + i \leq w, 0 \leq i \leq K - 1 \).

**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 \( \ell \)th position of a code word is 0, then \( \alpha^\ell \) 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 \( i_1, i_2, \ldots, i_m \) and \( i'_1, i'_2, \ldots, i'_m \) \( (0 \leq m \leq K - 1) \), where
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 \( i_1, i_2, \ldots, i_m \), 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 \( \text{GF}(q) \), 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 \( i_1, i_2, \ldots, i_m \) and \( i'_1, i'_2, \ldots, i'_m \).

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:

\[
P_e(m) = P\{V_m = w\},\]

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

**Lemma 2.1 (GYÖRI (2004B)).** The distribution of \( |S(U_i)| \) is the following:

\[
p_\ell = P\{|S(U_i)| = \ell\} = \frac{(q-1)(q-1)^{K-\ell-1} \sum_{k=0}^{K-\ell-1} (-1)^k (q-\ell-2\choose k) q^{K-\ell-k-1}}{q^K - 1}
\]

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

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

for all \( d_{\text{min}} \leq w \leq N \), otherwise it is 0. The number of zero coordinates trivially equals to \( N - w \). We get probabilities \( p_\ell \) 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 \( d_{\text{min}} = N - K + 1 \) and code length \( N = q - 1 \), so \( p_{\ell} \) can be calculated in the following way:

\[
p_{\ell} = \frac{(q-1)(q-1) \sum_{k=0}^{K-\ell-1} (-1)^k q^{\ell-k} q^k}{q^{K-1}}
\]

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

\[
P_e(m) \approx \Phi\left(-\frac{m - \mathbb{E}\{V_m\}}{\sigma\{V_m\}}\right).
\]

For calculating this approximation of 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 \( K \leq 3 \) (however they solved the asynchronous access, too). This result can be extended to our case when \( K \) can be greater than 3.

**Lemma 2.2 (Győri (2004b)).** The mean value and the variance of Markov chain \( \{V_m\} \) are

\[
\mathbb{E}\{V_m\} = w \left( 1 - \left( 1 - \frac{\bar{\tau}}{w} \right)^m \right)
\]

and

\[
\sigma^2\{V_m\} = \mathbb{E}\{V_m\} - \mathbb{E}\{V_m\}^2 + w(w-1) \left( 1 - 2 \left( 1 - \frac{\bar{\tau}}{w} \right)^m + \left( \sum_{\ell=0}^{K-1} p_{\ell} \left( \frac{\bar{\tau}^2}{\ell^2} \right)^m \right) \right), \tag{2.5}
\]

where

\[
\bar{\tau} = \mathbb{E}\{\text{\#}(U_1)\} = \sum_{\ell=1}^{K-1} \ell p_{\ell}. \tag{2.6}
\]

**Proof.** As

\[
V_m = \left| \bigcup_{i=1}^{m} S(U_i) \right| = \sum_{j=1}^{w} I \{ j \in \bigcup_{i=1}^{m} S(U_i) \},
\]

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

\[
\mathbb{E}\{V_m\} = \sum_{j=1}^{w} \mathbb{P}\left\{ j \in \bigcup_{i=1}^{m} S(U_i) \right\}
\]

\[
= \sum_{j=1}^{w} \left( 1 - \left( 1 - \mathbb{P}\{ j \in S(U_1) \} \right)^m \right).
\]

One can write

\[
\mathbb{P}\{ j \in S(U_1) \} = \sum_{\ell=1}^{K-1} \mathbb{P}\{ j \in S(U_1) \mid |S(U_1)| = \ell \} \mathbb{P}\{|S(U_1)| = \ell \}
\]

\[
= \sum_{\ell=1}^{K-1} \frac{\binom{w-1}{\ell-1}}{w} \frac{\ell p_{\ell}}{w} = \frac{\bar{\tau}}{w},
\]
where
\[ c = \mathbb{E}\{|S(U_1)|\} = \sum_{\ell=1}^{K-1} \ell p_\ell, \]
so
\[ \mathbb{E}\{V_m\} = w \left( 1 - \left( 1 - \frac{c}{w} \right)^m \right). \]

For the second moment the following is true:
\[
\mathbb{E}\{V_m^2\} = \mathbb{E}\left\{ \left( \sum_{j=1}^{w} I\left\{ j \in \bigcup_{i=1}^{m} S(U_i) \right\} \right)^2 \right\} \\
= \mathbb{E}\{V_m\} + \sum_{j \neq k} P\left\{ j \in \bigcup_{i=1}^{m} S(U_i), k \in \bigcup_{i=1}^{m} S(U_i) \right\} \\
= \mathbb{E}\{V_m\} + \sum_{j \neq k} \left( 1 - P\left\{ j \not\in \bigcup_{i=1}^{m} S(U_i) \right\} \right) - P\left\{ k \not\in \bigcup_{i=1}^{m} S(U_i) \right\} \\
+ P\left\{ \left\{ j \not\in \bigcup_{i=1}^{m} S(U_i) \right\} \cap \left\{ k \not\in \bigcup_{i=1}^{m} S(U_i) \right\} \right\} \\
= \mathbb{E}\{V_m\} + w(w-1) \cdot \left( 1 - 2 \left( 1 - \frac{c}{w} \right)^m + \left( \sum_{\ell=0}^{K-1} p_\ell \binom{w-2}{\ell} \right)^m \right),
\]
from which it follows that
\[
\sigma^2\{V_m\} = \mathbb{E}\{V_m\} - \mathbb{E}\{V_m\}^2 + w(w-1) \cdot \left( 1 - 2 \left( 1 - \frac{c}{w} \right)^m + \left( \sum_{\ell=0}^{K-1} p_\ell \binom{w-2}{\ell} \right)^m \right).
\]

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 2.3 (Hoeffding (1963)).** Let \( X_1, \ldots, X_n \) be independent real-valued random variables, let \( a_1, b_1, \ldots, a_n, b_n \in \mathbb{R} \), and assume that \( X_i \in [a_i, b_i] \) with probability one \( (i = 1, \ldots, n) \). Then, for all \( \varepsilon > 0 \),
\[
P\left\{ \frac{1}{n} \sum_{i=1}^{n} (X_i - \mathbb{E}\{X_i\}) > \varepsilon \right\} \leq \exp \left( -\frac{2\varepsilon^2}{\frac{1}{n} \sum_{i=1}^{n} (b_i - a_i)^2} \right).
\]

**Lemma 2.4 (Győri (2004B)).** If \( \frac{w}{m} \geq \mathbb{E}\{|S(U_1)|\} \), then the detection error probability can be upper bounded as
\[
P_e(m) \leq \exp \left( -\frac{2m\left( \frac{w}{m} - \mathbb{E}\{|S(U_1)|\} \right)^2}{(K-1)^2} \right).
\]
CHAPTER 2. OR CHANNEL

PROOF.

\[ P_e(m) = P\{V_m = w\} \]
\[ = P\left\{ \bigcup_{i=1}^{m} S(U_i) = w \right\} \]
\[ \leq P\left\{ \sum_{i=1}^{m} |S(U_i)| \geq w \right\} \]
\[ = P\left\{ \frac{1}{m} \sum_{i=1}^{m} (|S(U_i)| - E\{|S(U_i)|\}) \geq \frac{w}{m} - E\{|S(U_1)|\} \right\}. \]

Let us apply Hoeffding’s inequality (Lemma 2.3) to this probability with \( a = 0, b = K - 1 \). If \( \varepsilon := \frac{w}{m} - E\{|S(U_1)|\} > 0 \) the following bound stands:
\[ P_e(m) \leq \exp\left( -\frac{2m\left(\frac{w}{m} - E\{|S(U_1)|\}\right)^2}{(K-1)^2} \right), \]

otherwise we use trivial bound
\[ P_e(m) \leq 1, \]
where \( E\{|S(U_1)|\} \) can be calculated by (2.6). □

The next bound follows from Bernstein’s inequality:

**LEMMA 2.5 (BERNSTEIN (1946)).** Let \( X_1, \ldots, X_n \) be independent real-valued random variables, let \( a, b \in \mathbb{R} \) with \( a < b \), and assume that \( X_i \in [a, b] \) with probability one \( (i = 1, \ldots, n) \). Let
\[ \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \sigma^2 \{X_i\} > 0. \]

Then, for all \( \varepsilon > 0 \),
\[ P\left\{ \frac{1}{n} \sum_{i=1}^{n} (X_i - E\{X_i\}) > \varepsilon \right\} \leq \exp\left( -\frac{m\varepsilon^2}{2\sigma^2 + \frac{3}{2}(b-a)} \right). \]

**LEMMA 2.6 (GYŐRI (2004B)).** If \( \frac{w}{m} \geq E\{|S(U_1)|\} \), then the detection error probability can be upper bounded as
\[ P_e(m) \leq \exp\left( -\frac{m\left(\frac{w}{m} - E\{|S(U_1)|\}\right)^2}{2\sigma^2\{|S(U_1)|\} + 2\left(\frac{w}{m} - E\{|S(U_1)|\}\right)(K-1)/3} \right). \]

**PROOF.** Similarly to the first part of the proof of Lemma 2.4, and then by applying Bernstein’s inequality (Lemma 2.5) an alternative upper bound can be calculated. If \( \varepsilon := \frac{w}{m} - E\{|S(U_1)|\} > 0 \), the error probability can be bounded
\[ P_e(m) \leq P\left\{ \frac{1}{m} \sum_{i=1}^{m} (|S(U_i)| - E\{|S(U_i)|\}) \geq \frac{w}{m} - E\{|S(U_1)|\} \right\} \]
\[ \leq \exp\left( -\frac{m\left(\frac{w}{m} - E\{|S(U_1)|\}\right)^2}{2\sigma^2\{|S(U_1)|\} + 2\left(\frac{w}{m} - E\{|S(U_1)|\}\right)(K-1)/3} \right). \]
Table 2.1: Detection error probabilities for at most $M_{\text{max}}$ active users

<table>
<thead>
<tr>
<th>$T$</th>
<th>$M_0$</th>
<th>$M_{\text{max}}$</th>
<th>$q$</th>
<th>$K$</th>
<th>$n$</th>
<th>$P_e$</th>
<th>Markov</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>42</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$10^3$</td>
<td>2</td>
<td>4</td>
<td>11</td>
<td>5</td>
<td>110</td>
<td>7.952 · $10^{-6}$</td>
<td></td>
</tr>
<tr>
<td>$10^3$</td>
<td>2</td>
<td>6</td>
<td>13</td>
<td>5</td>
<td>156</td>
<td>8.295 · $10^{-6}$</td>
<td></td>
</tr>
<tr>
<td>$10^3$</td>
<td>7</td>
<td>20</td>
<td>23</td>
<td>4</td>
<td>506</td>
<td>7.686 · $10^{-6}$</td>
<td></td>
</tr>
<tr>
<td>$10^3$</td>
<td>25</td>
<td>85</td>
<td>53</td>
<td>3</td>
<td>2756</td>
<td>8.247 · $10^{-6}$</td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>5</td>
<td>42</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>4</td>
<td>110</td>
<td>3.410 · $10^{-6}$</td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>10</td>
<td>21</td>
<td>23</td>
<td>3</td>
<td>506</td>
<td>7.249 · $10^{-6}$</td>
<td></td>
</tr>
</tbody>
</table>

otherwise we use trivial bound

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

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

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

Let us suppose that the maximum number of simultaneously active users $M_{\text{max}}$ is given. Kautz–Singleton construction guarantees error free detection if at most $M_0$ users are active (cf. eq. (2.3)). We have detection error if the number of active users $m$ is between $M_0$ and $M_{\text{max}}$. 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 model (cf. eq. (2.4)) for the worst case situation (when $M_{\text{max}}$ users are active):

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

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 $M_{\text{max}}$ 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 $T = 10^3$ 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:

$$P_e(T, M_0, p) = \sum_{m=M_0+1}^{T} \binom{T}{m} p^m (1-p)^{T-m} P_e(m).$$
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. 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 “$P_e \Phi$” of Tables 2.2 and 2.3,
Table 2.2: Detection error probabilities for $T = 10^5$

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

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

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

It can be seen that for some parameters Gaussian approximation can be smaller than the exact value of the detection error probability.

### 2.3 Fast frequency hopping

In this section fast frequency hopping (FFH) communications systems 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. 2.4).

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 $\alpha$ be a primitive element of $\text{GF}(L)$. If $G$ denotes the generator matrix of a Reed–Solomon code with parameters $(N, K)$ (see (2.2)), then for $K = 2$ Einarsson (1980) defined the code words as

$$c = (m, a)G,$$

where $a \in \text{GF}(L)$ is the address (identifier) of the user, and $m \in \text{GF}(L)$ is his message. As mentioned previously, the code word $c$ 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 $K = 3$

$$c = (m, 1, a)G.$$

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.

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.

Dyachkov and Rykov (1983) and Erdős et al. (1985) proved the following:

**Lemma 2.7 (Dyachkov and Rykov (1983), Erdős et al. (1985)).**

$$T \leq M \left( \frac{Ln}{n/M} \right) \left( \frac{n}{n/M} \right)$$  (2.7)
We note that this lemma forms the basis of the proof given to the lower bound on the minimum code length for OR channel in (2.1). Generally, on the OR channel arbitrary binary code words can be applied, but in the case of FFH only code words of constant weight are allowed. So, from this a lower bound on the minimum code length (for FFH) can be given by applying the next lemma.

**Lemma 2.8 (Binomial Bound, cf. Problem 5.8 in Gallager (1968)).** For $1 \leq k \leq n-1, n \geq 2$,

$$\sqrt{\frac{n}{8k(n-k)}} \leq \binom{n}{k} 2^{-nh(\frac{k}{n})} < \sqrt{\frac{n}{2\pi k(n-k)}}.$$

**Theorem 2.1 (Győri (2005a)).** If $M, L$ are fixed and $T \to \infty$, then

$$n_{\text{asyn}}(T, M, L) \geq n_{\text{syn}}(T, M, L) \gtrsim \frac{M}{\log L} \log T.$$

**Proof.** Let us apply Lemma 2.8 on (2.7), then we get

$$T \lesssim 2^{Lh(\frac{1}{LM}) - nh(\frac{1}{M})}$$

from which the statement follows. \(\square\)

**Remark.** If $M \to \infty$, too, then the lower bound can be given in a rather simple form:

$$n_{\text{asyn}}(T, M, L) \geq n_{\text{syn}}(T, M, L) \gtrsim \frac{M}{\log L} \log T.$$

Next a random coding argument will be applied to give an upper bound on the minimum code length $n$. A and Zeisel (1988) studied the frame synchronous case. They found that the minimum code length $n$ can be upper bounded by the following:

**Theorem 2.2 (A and Zeisel (1988)).** For frame synchronous access, if $M, L$ are fixed and $T \to \infty$, then

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

**Remark.** In the case of synchronous access the probability of false detection, i.e., if a given code word is covered by $M$ others, is

$$\left(1 - \left(1 - \frac{1}{L}\right)^M\right)^n. \quad (2.8)$$

In Theorem 2.3 we prove that, asymptotically, the upper bound in Theorem 2.2 is true for the asynchronous case, too.

**Theorem 2.3 (Győri (2005a)).** For frame asynchronous access, if $M, L$ are fixed and $T \to \infty$, then

$$n_{\text{asyn}}(T, M, L) \lesssim \frac{M+1}{-\log \left(1 - \left(1 - \frac{1}{L}\right)^M\right)} \log T.$$
REMARK. This result can be further upper bounded to clearly show how the minimum code length depends on the parameters $M, L$ and $T$:

$$n_{\text{async}}(T, M, L) \lesssim \ln 2 (M + 1) 4^M \log T.$$  

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 chosen independently 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 whose 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 $M$ shifted code words cover another code word.

**Lemma 2.9 (Győri (2005a)).** For frame asynchronous access

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

(2.9)

**Proof.** Let us fix $M$ arbitrarily shifted hopping sequences and choose an $(M + 1)$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
\[ P\{ \text{false identification}\} \leq \left( \frac{T}{M} \right) (T-M)n^M \left( 1 - \left( \frac{1}{L} \right)^M \right)^n \]
\[ \leq T^{M+1} n^M \left( 1 - \left( \frac{1}{L} \right)^M \right)^n \]
\[ = e^{(M+1)\ln T + Mn\ln \left( 1 - \left( \frac{1}{L} \right)^M \right)}, \]
where the factor \( n^M \) is needed because of the shift of the other hopping sequences.

The tagged user may be 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 \( D_1, \ldots, D_d \) can be distinguished, where
\[ D_j = \left\{ j + \ell d : \ell = 0, 1, \ldots, k-1 \text{ and } k = \left\lfloor \frac{n-j}{d} \right\rfloor + 1 \right\} \]
\((j = 1, \ldots, d)\). 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 \(|D_1| + \cdots + |D_d| = n\).

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 \( 1, 2, \ldots, k \) as slot indices instead of \( j, j+d, \ldots, j+(k-1)d \). \( c_\ell \) denotes the location of the 1 for the code word of the tagged user at slot \( \ell \) \((\ell = 1, \ldots, k)\), and let \( U_\ell = (U_\ell(1), \ldots, U_\ell(L)) \) 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 \( \ell \)). 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 \( U_1 \). (Remember, that exactly \( M \) active users were supposed in each time slot.) Thus, for all \( \ell = 1, \ldots, k \)
\[ P\{c_\ell = \phi\} = \frac{1}{L}, \]
and
\[ P\{U_\ell(\phi) = 0\} = \begin{cases} (1 - \frac{1}{L})^{M-1}, & \text{if } \ell = 2, \ldots, k, \\ (1 - \frac{1}{L})^M, & \text{if } \ell = 1 \end{cases}, \]
\[ P\{U_\ell(\phi) = 1\} = 1 - P\{U_\ell(\phi) = 0\}, \]
where \( \phi = 1, \ldots, L \), and the components of vector \( U_\ell \) are independent of each other (cf. Fig. 2.5).

**Lemma 2.10.** If \( V, W \) and \( Z \) are independent random variables, and \( f(\cdot), g(\cdot) \) are arbitrary functions, then
\[ E\{ f(V, W)g(V, Z) \mid V \} = E\{ f(V, W) \mid V \} E\{ g(V, Z) \mid V \}. \]
Figure 2.5: Components of the hopping sequences of the tagged user and other active users

**PROOF.**

\[
E\{f(V,W)g(V,Z) \mid V\} = E\{E\{f(V,W)g(V,Z) \mid V,W\} \mid V\} \\
= E\{f(V,W)E\{g(V,Z) \mid V,W\} \mid V\} \\
= E\{f(V,W)E\{g(V,Z) \mid V\} \mid V\} \\
= E\{g(V,Z) \mid V\} E\{f(V,W) \mid V\}.
\]

\[
\]

**LEMMA 2.11 (GYÖRI (2005A)).** For frame asynchronous access

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

**PROOF.** Let us introduce the sequence of events

\[
A_\ell := \{\text{time slot } \ell \text{ is covered}\} \\
= \left\{\begin{array}{ll}
\{c_{\ell-1} = c_\ell\} \cup \{c_{\ell-1} \neq c_\ell\} \cap \{U_\ell(c_\ell) = 1\}, & \text{if } \ell = 2, \ldots, k, \\
\{U_1(c_1) = 1\}, & \text{if } \ell = 1.
\end{array}\right.
\]

Thus

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

We denote by \(a_{\ell}^{\phi}\) (\(i = 1, \ldots, k, \phi = 1, \ldots, L\)) 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 \(\phi\) at the \(i^{th}\) position \(c_i = \phi\).

\[
a_{i}^{\phi} := P\left\{\bigcap_{\ell=1}^i A_\ell \mid c_i = \phi\right\}.
\]

Therefore

\[
f(D_j) = P\left\{\bigcap_{\ell=1}^k A_\ell\right\} \\
= \sum_{\phi=1}^L P\left\{\bigcap_{\ell=1}^k A_\ell \mid c_k = \phi\right\} P\{c_k = \phi\} \\
= \frac{1}{L} \sum_{\phi=1}^L a_{k}^{\phi}.
\]
Let us apply Lemma 2.10 with \( V = \{ c_{i-1}, c_i \}, W = \{ c_1, \ldots, c_{i-2}, U_1, \ldots, U_{i-1} \}, Z = \{ U_i \}, \) and \( f(V, W) = I_{\bigcap_{\ell=1}^{i-1} A_{\ell}} \), \( g(V, Z) = I_{\{A_i\}} \). (Note, that \( \mathbf{P}\{B\} = \mathbf{E}\{I_B\} \) for an arbitrary event \( B \).)

\[
\mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i, c_{i-1} \right\} = \mathbf{E}\left\{ I_{\bigcap_{\ell=1}^{i-1} A_{\ell}} \mid I_{\{A_i\}} \mid c_i, c_{i-1} \right\}
= \mathbf{E}\{f(V, W)g(V, Z) \mid V\}
= \mathbf{E}\{f(V, W) \mid V\} \mathbf{E}\{g(V, Z) \mid V\}
= \mathbf{E}\left\{ I_{\bigcap_{\ell=1}^{i-1} A_{\ell}} \mid c_i, c_{i-1} \right\} \mathbf{E}\{I_{\{A_i\}} \mid c_i, c_{i-1} \}
= \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_i, c_{i-1} \right\} \mathbf{P}\{A_i \mid c_i, c_{i-1}\}
\]

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

\[
da_i^\phi = \mathbf{P}\left\{ \bigcap_{\ell=1}^{i} A_{\ell} \mid c_i = \phi \right\}
= \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i = \phi \right\}
= \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i = \phi, c_{i-1} = \psi \right\} \mathbf{P}\{c_{i-1} = \psi\}
= \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_i = \phi, c_{i-1} = \psi \right\} \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\}
= \frac{1}{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_{i-1} = \phi \right\} \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \phi\}
+ \frac{1}{L} \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_{i-1} = \psi \right\} \mathbf{P}\{A_i \mid c_i = \phi, c_{i-1} = \psi\}
= \frac{1}{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_{i-1} = \phi \right\} \cdot 1
+ \frac{1}{L} \sum_{\psi=1}^{L} \mathbf{P}\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_{i-1} = \psi \right\} \left( 1 - \left(1 - \frac{1}{L}\right)^{M-1} \right)
= \frac{1}{L} a_{i-1}^\phi + \frac{1}{L} \left(1 - \left(1 - \frac{1}{L}\right)^{M-1} \right) \sum_{\psi=1}^{L} a_{i-1}^\psi
\]

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.

\[
a^\phi_1 := P\{A_1 \mid c_1 = \phi\} = P\{U_1(c_1) = 1 \mid c_1 = \phi\} = P\{U_1(\phi) = 1\} = 1 - (1 - \frac{1}{L})^M.
\]

Introduce the \(L \times L\) matrix

\[
A = \begin{pmatrix}
1 & 1 - (1 - \frac{1}{L})^{M-1} & \cdots & 1 - (1 - \frac{1}{L})^{M-1} \\
1 - (1 - \frac{1}{L})^{M-1} & 1 & \cdots & 1 - (1 - \frac{1}{L})^{M-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 - (1 - \frac{1}{L})^{M-1} & 1 - (1 - \frac{1}{L})^{M-1} & \cdots & 1
\end{pmatrix},
\]

then

\[
f(D_j) = \left(\frac{1}{L}, \frac{1}{L}, \ldots, \frac{1}{L}\right) (a^1_k, a^2_k, \ldots, a^L_k)^T = \left(\frac{1}{L}, \frac{1}{L}, \ldots, \frac{1}{L}\right) \frac{1}{L}^k \lambda a^1_k, a^2_k, \ldots, a^L_k)^T
\]

\[
\vdots
\]

\[
= \left(\frac{1}{L}, \frac{1}{L}, \ldots, \frac{1}{L}\right) \frac{1}{L^{k-1}} \lambda^{k-1} (a^1_k, a^2_k, \ldots, a^L_k)^T
\]

\[
= \frac{1}{L^{k-1}} \left(\frac{1}{L}, \frac{1}{L}, \ldots, \frac{1}{L}\right) \lambda^{k-1} \begin{pmatrix}
1 - (1 - \frac{1}{L})^M \\
1 - (1 - \frac{1}{L})^M \\
\vdots \\
1 - (1 - \frac{1}{L})^M
\end{pmatrix}.
\]

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

\[
\lambda_1 = L \left(1 - (1 - \frac{1}{L})^M \right),
\]

\[
\lambda_2 = \cdots = \lambda_L = \left(1 - \frac{1}{L}\right)^{M-1},
\]

and the corresponding eigenvectors are

\[
v_1 = \left(1, 1, \cdots 1\right)^T, \\
v_2 = \left(1, -1, 0, \cdots 0\right)^T, \\
v_3 = \left(1, 0, -1, \cdots 0\right)^T, \\
\vdots \\
v_L = \left(1, 0, 0, \cdots -1\right)^T.
\]
Thus, the decomposition of matrix $A$ is

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

where

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

$$\Lambda = \begin{pmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_2
\end{pmatrix},$$

$$V^{-1} = \begin{pmatrix}
\frac{1}{\ell} & \frac{1}{\ell} & \cdots & \frac{1}{\ell} \\
\frac{1}{\ell} & \frac{1}{\ell} & -1 & \cdots & \frac{1}{\ell} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\frac{1}{\ell} & \frac{1}{\ell} & \cdots & \frac{1}{\ell} & -1
\end{pmatrix},$$

the $(k-1)$th power of $A$ is

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

where

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

and the probability $f(D_j)$ is

$$f(D_j) = \frac{1}{L^{k-1}} \left(1 - \left(1 - \frac{1}{\ell}\right)^M\right) \lambda_1^{k-1}$$

$$= \frac{1}{L^{k-1}} \left(1 - \left(1 - \frac{1}{\ell}\right)^M\right) L^{k-1} \left(1 - \left(1 - \frac{1}{\ell}\right)^M\right)^{k-1}$$

$$= \left(1 - \left(1 - \frac{1}{\ell}\right)^M\right)^k,$$

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

\[
P\{\text{code word of the tagged user is covered}\} = \prod_{j=1}^{d} P\{\text{all positions in } D_j \text{ are covered}\}
\]

\[
= \prod_{j=1}^{d} P\{\text{all positions in } D_j \text{ are covered}\}
\]

\[
= \prod_{j=1}^{d} f(D_j)
\]

\[
= \prod_{j=1}^{d} \left(1 - \left(1 - \frac{1}{M}\right)^{|D_j|}\right)
\]

\[
= \left(1 - \left(1 - \frac{1}{M}\right)^{\sum_{j=1}^{d} |D_j|}\right)^n
\]

\[
= \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)^n,
\]

so the probability of false synchronization in the asynchronous case equals the probability of false detection in the synchronous case (2.8).

\[\tag{2.10}\]

**Lemma 2.12 (Győri (2005 A)).** For frame asynchronous access

\[
P\{\text{false synchronization}\} \leq e^{M \ln T + M \ln n + n \ln \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)}.
\]

**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 2.11 the probability that the code word of the tagged user is covered by the others can be upper bounded by \(\left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)^n\), so

\[
P\{\text{false synchronization}\} \leq \left(\frac{T}{M - 1}\right) (T - M + 1) n^M \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)^n
\]

\[
\leq T^M n^M \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)^n
\]

\[
e^{M \ln T + M \ln n + n \ln \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right)},
\]

where the factor \(n^M\) is needed because of the shifts of the hopping sequences.

**Proof of Theorem 2.3.** If a randomly chosen code \(C\) which has \(T\) hopping sequences of length \(n\) satisfy the requirements of identification and synchronization, then \(C\) can be applied for \(T\) users in communication by the fast frequency hopping scheme. If we choose the code length \(n\) to

\[
n = (1 + \delta) \frac{M + 1}{-\log \left(1 - \left(1 - \frac{1}{M}\right)^{M}\right) \log T}
\]
for an arbitrary constant \( \delta > 0 \), the exponents in (2.9) and (2.10) become

\[-(M + 1) \log T \left( \delta \left(1 - \frac{\gamma}{M+1}\right) \ln 2 - (1 - \frac{1}{M+1}) \right) \frac{\ln \left(1 + \delta\right)}{-\log \left(1 - \left(1 - \frac{1}{T}\right)^M\right) \log T} \]

where constant \( \gamma = 0 \) and 1, respectively. Both exponents tend to \(-\infty\) when \( T \to \infty \), hence concerning to (1.1) and (1.2) there exists a good code \( C \).

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

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

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

2.4 Asynchronous OR channel

There are a lot of studies on signature coding for multiple access OR channel in the literature, but all of them assume frame synchronous access among users (all active users begin transmitting their code words at the same time). Dyachkov and Rykov (1983), Erdős et al. (1985), Hwang and T. Sós (1987), A and Zeisel (1988), Ruszinkó (1994), Füredi (1996) gave lower and upper bounds on the minimum code length \( n \).

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 2.3 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.

In Theorem 2.4 we repeat the result of Dyachkov and Rykov (1983) on the upper bound on code length \( n \) for the easy comparability with the asynchronous access in Theorem 2.5.

Consider a binary random code of 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 is binomially distributed.

**THEOREM 2.4 (DYACHKOV AND RYKOV (1983)).** If \( M \) is fixed, \( T \to \infty \) and frame synchronous access is used, then

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

**REMARK.** In the case of synchronous access the probability of false detection, i.e., if a given code word is covered by \( M \) others, is

\[ (1 - p(1 - p)^M)^n. \] (2.11)
CHAPTER 2. OR CHANNEL

If frame asynchronous access is assumed, the coding method have to ensure not just the identification but the synchronization, too. In Theorem 2.5 we give upper bound on \( n_{\text{asyn}}(T, M) \), and show that the bounds for synchronous and asynchronous access are asymptotically equal.

**THEOREM 2.5 (GYŐRI (2005 C)).** For frame asynchronous access, if \( M \) is fixed and \( T \to \infty \)

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

The detection is done by the following algorithm. A sliding window is used whose 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 another user.

**LEMMA 2.13 (GYŐRI (2005 C)).** For frame asynchronous access, if \( p = \frac{1}{M+1} \)

\[
P\{\text{false identification}\} \leq e^{(M+1) \ln T + M \ln n - \frac{n}{M+1^2} e^{-1}}.
\]

**PROOF.** 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

\[
P\{\text{false identification}\} \leq \binom{T}{M} (T-M)n^M (1-p(1-p)^M)^n,
\]

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

\[
P\{\text{false identification}\} \leq \binom{T}{M} (T-M)n^M \left(1 - \frac{1}{M+1}\right)^M \left(1 - \frac{1}{M+1}\right)^n
\]

\[
\leq T^{M+1} n^M \left(1 - \frac{e^{-1}}{M+1}\right)^n
\]

\[
\leq T^{M+1} n^M e^{n \frac{e^{-1}}{M+1}}
\]

\[
eq e^{(M+1) \ln 2 \log T + M \ln n - \frac{n}{M+1^2} e^{-1}},
\]

(2.12)

where we applied that \( (1 - \frac{1}{M+1})^M \geq e^{-1} \), and \( 1 + x \leq e^x \) 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 $D_1, \ldots, D_d$ can be distinguished, where

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

$(j = 1, \ldots, d)$. 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 $j, j + d, \ldots, j + (k-1)d$. $c_\ell$ denotes the component of the code word of the tagged user at position $\ell$ ($\ell = 1, \ldots, k$), and let $U_\ell$ 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 $\ell = 1, \ldots, k$

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

and

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

$$P\{U_\ell = 1\} = 1 - P\{U_\ell = 0\}$$

(cf. Fig. 2.6).

**Lemma 2.14 (Győri (2005c)).** For frame asynchronous access, if $p = \frac{1}{M+1}$

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

**Proof.** Let us introduce the sequence of events

$$A_\ell := \{\text{position } \ell \text{ is covered}\} = \begin{cases} \{c_{\ell-1} = 1\} \cup \{c_{\ell-1} = 0\} \cap \{c_\ell = 1, U_\ell = 0\}^c, & \text{if } \ell = 2, \ldots, k, \\ \{c_1 = 1, U_1 = 0\}^c, & \text{if } \ell = 1, \end{cases}$$
where \( \{ \}^c \) denotes the complement of an event. Thus

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

Hence

\[
f(D_j) = P\left\{ \bigcap_{\ell=1}^{k} A_{\ell} \right\} = P\left\{ \bigcap_{\ell=1}^{k} A_{\ell} \mid c_k = 1 \right\} P\{c_k = 1\} + P\left\{ \bigcap_{\ell=1}^{k} A_{\ell} \mid c_k = 0 \right\} P\{c_k = 0\} = p a_k^1 + (1 - p) a_k^0.
\]

Let us apply Lemma 2.10 with \( V = \{c_{i-1}, c_i\}, W = \{c_1, \ldots, c_{i-2}, U_1, \ldots, U_{i-1}\}, Z = \{U_i\}, \) and \( f(V, W) = I_{\bigcap_{\ell=1}^{i-1} A_{\ell}}, g(V, Z) = I_{(A_i)}. \) (Note, that \( P\{B\} = E\{I_{(B)}\} \) for an arbitrary event \( B. \))

\[
P\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i, c_{i-1} \right\} = E\left\{ I_{\bigcap_{\ell=1}^{i-1} A_{\ell}} I_{(A_i)} \mid c_i, c_{i-1} \right\} = E\{f(V, W)g(V, Z) \mid V\} = E\{f(V, W) \mid V\} E\{g(V, Z) \mid V\} = E\left\{ I_{\bigcap_{\ell=1}^{i-1} A_{\ell}} \mid c_i, c_{i-1} \right\} E\{I_{(A_i)} \mid c_i, c_{i-1} \} = P\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_i, c_{i-1} \right\} P\{A_i \mid c_i, c_{i-1} \}\]

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

\[
a_i^\phi = P\left\{ \bigcap_{\ell=1}^{i} A_{\ell} \mid c_i = \phi \right\} = P\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i = \phi \right\} = \sum_{\psi=0}^{1} P\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \cap A_i \mid c_i = \phi, c_{i-1} = \psi \right\} P\{c_{i-1} = \psi\} = \sum_{\psi=0}^{1} P\left\{ \bigcap_{\ell=1}^{i-1} A_{\ell} \mid c_i = \phi, c_{i-1} = \psi \right\} P\{c_i = \phi, c_{i-1} = \psi\} P\{c_{i-1} = \psi\}.
\]
\[ a_i = \sum_{\psi=0}^{\psi_i} P\{A_i \mid c_i = \psi, c_{i-1} = \psi\} P\{c_{i-1} = \psi\}, \]

thus
\[ a_i^1 = pa_i^1 + (1 - p)(1 - (1 - p)^{M-1}) a_i^{0}, \]
\[ a_i^0 = pa_i^1 + (1 - p)a_i^{0}. \]

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.

\[
\begin{align*}
a_1^\phi &:= P\{A_1 \mid c_1 = \phi\} \\
&= P\{\{c_1 = 1, U_1 = 0\} \mid c_1 = \phi\} \\
&= 1 - P\{c_1 = 1, U_1 = 0 \mid c_1 = \phi\},
\end{align*}
\]

so
\[
\begin{align*}
a_1^1 &= 1 - (1 - p)^M, \\
a_1^0 &= 1.
\end{align*}
\]

Introduce the notation
\[
A = \begin{pmatrix} p & (1 - p)(1 - (1 - p)^{M-1}) \\ p & 1 - p \end{pmatrix},
\]

then
\[
\begin{align*}
f(D_j) &= (p, 1 - p) \begin{pmatrix} a_k^1 \\ a_k^0 \end{pmatrix} \\
&= (p, 1 - p) A^{k-1} \begin{pmatrix} a_1^1 \\ a_1^0 \end{pmatrix} \\
&\vdots \\
&= (p, 1 - p) A^{k-1} \begin{pmatrix} 1 - (1 - p)^M \\ 1 \end{pmatrix}.
\end{align*}
\]

For calculating the power of matrix \( A \) firstly its diagonal form is needed. It has two eigenvalues
\[
\begin{align*}
\lambda_1 &= \frac{1}{2} + \frac{1}{2} \sqrt{1 - 4p(1 - p)^M}, \\
\lambda_2 &= \frac{1}{2} - \frac{1}{2} \sqrt{1 - 4p(1 - p)^M},
\end{align*}
\]
and the corresponding eigenvectors are
\[ v_1 = \left( \frac{\lambda_1-1+p}{p}, 1 \right)^T, \quad v_2 = \left( \frac{\lambda_2-1+p}{p}, 1 \right)^T. \]

Thus, the decomposition of matrix \( A \) is
\[
A = \begin{pmatrix}
\frac{\lambda_1-1+p}{p} & \frac{\lambda_2-1+p}{p} \\
1 & 1
\end{pmatrix}
\begin{pmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{pmatrix}
\begin{pmatrix}
\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_2-1+p}{\lambda_1-\lambda_2} \\
\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_1-1+p}{\lambda_1-\lambda_2}
\end{pmatrix},
\]
the \((k-1)\)th power of \( A \) is
\[
A^{k-1} = \begin{pmatrix}
\frac{\lambda_1-1+p}{p} & \frac{\lambda_2-1+p}{p} \\
1 & 1
\end{pmatrix}
\begin{pmatrix}
\lambda_1^{k-1} & 0 \\
0 & \lambda_2^{k-1}
\end{pmatrix}
\begin{pmatrix}
\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_2-1+p}{\lambda_1-\lambda_2} \\
\frac{p}{\lambda_1-\lambda_2} & \frac{\lambda_1-1+p}{\lambda_1-\lambda_2}
\end{pmatrix},
\]
and the probability \( f(D_j) \) is
\[
f(D_j) = \left( 1 + \sqrt{1-4q} \right)^{k-2} 2^{-(k-2)} \left( \frac{\frac{1}{2} - 2q + q^2}{\sqrt{1-4q}} + \frac{1}{2} \right)
- \left( 1 - \sqrt{1-4q} \right)^{k-2} 2^{-(k-2)} \left( \frac{\frac{1}{2} - 2q + q^2}{\sqrt{1-4q}} - \frac{1}{2} \right),
\]
where \( q = p(1-p)^M \). Notice, that \( 0 \leq q \leq \frac{4}{27} \approx 0.148 \) for all \( M \geq 2 \) and \( p = \frac{1}{M+1} \). By considering that for such a \( q \)
\[
\left( \frac{\frac{1}{2} - 2q + q^2}{\sqrt{1-4q}} - \frac{1}{2} \right) \geq 0,
\]
and
\[
\left( \frac{\frac{1}{2} - 2q + q^2}{\sqrt{1-4q}} + \frac{1}{2} \right) \leq (1-q)^2,
\]
\( f(D_j) \) can be upper bounded
\[
f(D_j) \leq \left( 1 + \sqrt{1-4q} \right)^{k-2} 2^{-(k-2)} (1-q)^2
= \left( \frac{1}{2} + \sqrt{\frac{1}{4} - q} \right)^{k-2} (1-q)^2
\leq (1-q)^{k-2} (1-q)^2
= (1-q)^k
= (1 - p(1-p)^M)^k.
\]
If $M = 1$ and $p = \frac{1}{M+1}$, then

$$f(D_j) = \left(\frac{1}{2}, \frac{1}{2}\right) \left(\begin{array}{cc} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{array}\right) \left(\frac{1}{2}\right)$$

$$= \left(\frac{1}{2}, \frac{1}{2}\right) \left(\begin{array}{c} \frac{k-1}{2k+1} \\ \frac{k-1}{2k+1} \end{array}\right) \left(\frac{1}{2}\right)$$

$$= \frac{k+2}{2^{k+1}}$$

$$\leq \left(\frac{3}{4}\right)^k,$$

where $1 - p(1 - p)^M = \frac{3}{4}$ for $M = 1$, so inequality (2.14) 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

$$P\{\text{code word of the tagged user is covered}\} = P\left\{ \bigcap_{j=1}^{d} \{ \text{all 1's in class } D_j \text{ are covered} \} \right\}$$

$$= \prod_{j=1}^{d} P\{ \text{all 1's in class } D_j \text{ are covered} \}$$

$$= \prod_{j=1}^{d} f(D_j)$$

$$\leq \prod_{j=1}^{d} (1 - p(1 - p)^M)^{|D_j|}$$

$$= (1 - p(1 - p)^M)^{\sum_{j=1}^{d} |D_j|}$$

$$= (1 - p(1 - p)^M)^{n},$$

so the probability of false synchronization in the asynchronous case is upper bounded by the probability of the false detection in synchronous case (2.11).

**Remark.** Lemma 2.14 is true for all $0 \leq p \leq 1$. Reader can find the details in Győri (2005c). The proof is similar to the previous proof, the difference is only in the case of $M = 1$.

**Lemma 2.15 (Győri (2005c)).** For frame asynchronous access, if $p = \frac{1}{M+1}$

$$P\{\text{false synchronization}\} \leq e^{M \ln(T+M)n - \frac{n}{M+1} - \frac{n}{2M+1} - 1}.$$
where the factor $n^{M-1}$ is needed because of the shift of the code words. Let $p := \frac{1}{M+1}$, then

$$P\{\text{false identification}\} \leq \left( \frac{T}{M-1} \right)^n (T - M + 1) n^M \left( 1 - \frac{1}{M+1} \left( 1 - \frac{1}{M+1} \right)^M \right)^n$$

$$\leq T^M n^M \left( 1 - \frac{e^{-1}}{M+1} \right)^n$$

$$\leq T^M n^M e^{-\frac{n}{M+1} e^{-1}}$$

$$= e^{M \ln T + M \ln n - \frac{n^2}{M+1} e^{-1}}. \quad (2.15)$$

**Proof of Theorem 2.5.** If a randomly chosen code $C$ which has $T$ code words of length $n$ satisfy the requirements of identification and synchronization, then $C$ can be applied for $T$ users in communication via a multiple access OR channel. If we choose $p = \frac{1}{M+1}$, and set the code length $n$ to

$$n = (1 + \delta) \ln 2 (M + 1)^2 \log T$$

for an arbitrary constant $\delta > 0$, the exponents in (2.12) and (2.15) become

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

where constant $\gamma = 0$ and 1, respectively. Both exponents tend to $-\infty$ when $T \to \infty$, that is why concerning to (1.1) and (1.2) there exists a good code $C$.

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

$$n_{\text{asyn}}(T, M) \lesssim e \ln 2 (M + 1)^2 \log T.$$
Chapter 3

Collision channel

3.1 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 and one output (cf. Figure 3.1). 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 $0$ symbol. The output of the channel can be $0$ 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 3.2 frame synchronization is assumed, so frames of the users begin at the same slots (no time shift), while in Section 3.3 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. For a given user there are previously fixed slots of a frame when he can send packets into the channel. So users have protocol sequences of length $n$, $q_i$ for user $i$, which has $1$ in the $j$th position if the user is allowed to send a packet in the $j$th slot. The protocol sequences can be considered as an outer code.

\[
\begin{align*}
  x_1 &\in I \cup \{0\} \\
  x_2 &\in I \cup \{0\} \\
  \vdots \\
  x_T &\in I \cup \{0\} \\
  y &\in I \cup \{0, *\}
\end{align*}
\]

Figure 3.1: Multiple access collision channel
Let $A$ be the set of messages of the users and suppose that $|A| = S$. User $i$ encodes each message $a_j \in A$ into a code word $c_i^j \in C_i$ of length $w(q_i)$, where $w(\cdot)$ is the Hamming weight, and $c_i^j \in I$ (1 $\leq j \leq S$, 1 $\leq i \leq T$). The components of $c_i^j$ are sent according to the protocol sequence $q_i$. $C_i$ is called the code of user $i$. If the protocol sequences have the same weight, then $C_i$ can be the same for all users. Because of collisions, some packets are erased during the transmissions, and these erasure errors are corrected using $C_i$. $C_i$ is the inner code.

If all the $T$ users were active all the time, then—for synchronous access—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 $C_i$ 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.


We derive asymptotic lower and upper bounds on the minimum frame size $n$ when $T \to \infty$, $S \to \infty$ and $\frac{\log T}{\log S} \to 0$.

Bassalygo and Pinsker (1983) considered this problem, even for asynchronous case, for binary packets, $|I| = 2$.

**Theorem 3.1** (Bassalygo and Pinsker (1983)). *For binary packets, if $M$ is fixed, $T \to \infty$ and $S \to \infty$ then*

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

**Theorem 3.2** (Bassalygo and Pinsker (1983)). *For binary packets, if $M$ is fixed, $T \to \infty, S \to \infty$ and $\frac{\log T}{\log S} \to 0$*

$$n(T, M, S) \lesssim M \left( 1 - \frac{1}{M} \right)^{1-M} \log S.$$  

Let us introduce the sum-rate

$$R_{\text{sum}} = \frac{M \log |I| S}{n}$$  

of communication, as usual, and denote by $R_{\text{sum}}(T, M)$ the maximum achievable sum-rate for parameters $T, M$.

**Corollary 3.1** (Bassalygo and Pinsker (1983)). *For binary packets, if $M$ is fixed, $T \to \infty, S \to \infty$ and $\frac{\log T}{\log S} \to 0$, then*

$$R_{\text{sum}}(T, M) \approx (1 - \frac{1}{M})^{M-1}.$$  

If, in addition, $M \to \infty$, then

$$R_{\text{sum}}(T, M) \approx e^{-1}.$$
In Sections 3.2 and 3.3 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, Győrfi, Massey (1992) and Győrfi, Vajda (1993)) are far from \( e^{-1} \).

### 3.2 Bounds for non-binary packets, synchronous access

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

\[
R_{\text{sum}} = \frac{kM}{n}.
\]

For getting the lower bound on the minimum block length \( n(T,M,k) \), the entropy bound will be used.

**Theorem 3.3 (Győrfi and Győrő (2004)).** For non-binary packets, if \( M \) is fixed, \( T \to \infty \), \( |I| \to \infty \) and \( \frac{\log T}{\log |I|} \to 0 \), then

\[
n(T,M,k) \gtrsim kM \left( 1 - \frac{1}{M} \right)^{1-M},
\]

and for the sum-rate

\[
R_{\text{sum}}(T,M) \lesssim (1 - \frac{1}{M})^{M-1}.
\]

If, in addition, \( M \to \infty \), then

\[
n(T,M,k) \gtrsim kMe,
\]

and for the sum-rate

\[
R_{\text{sum}}(T,M) \lesssim e^{-1}.
\]

**Proof.** For the minimum block length, an 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

\[
\log \left( \binom{T}{M} |I|^k_M \right) = H(O_1, \ldots, O_n)
\]

\[
\leq \sum_{i=1}^{n} H(O_i)
\]

\[
\leq n \max_i H(O_i),
\]
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:

$$P\{O_i = 0\} = \frac{(T-w_i)}{M} \left( \frac{T}{M} \right)^M := p_0,$$

$$P\{O_i = a\} = \frac{w_i}{|I|} \left( \frac{T-M}{M} \right)^{|I|} := \frac{p_1}{|I|}, \quad \forall a \in I$$

$$P\{O_i = \ast\} = 1 - \left( \frac{T}{M} \right) - \frac{w_i}{|I|} = 1 - p_0 - p_1.$$ 

The entropy of $O_i$ can be upper bounded as

$$H(O_i) \leq -p_0 \log p_0 - |I| \cdot \frac{p_1}{|I|} \log p_1 - (1 - p_0 - p_1) \log(1 - p_0 - p_1)$$

$$= -p_0 \log p_0 - p_1 \log p_1 - (1 - p_0 - p_1) \log(1 - p_0 - p_1) + p_1 \log |I|$$

$$\leq \log 3 + p_1 \log |I|$$

$$= \log 3 + \frac{w_i(T-w_i)}{(M-1)} \log |I|$$

$$= \log 3 + \frac{w_iM}{T} \left( \frac{T-w_i}{M} \right) \log |I|$$

$$\leq \log 3 + \frac{1}{1 - \frac{M-1}{T}} \left( \frac{1}{M} \right)^{M-1} \log |I|$$

$$\simeq \log 3 + \left( 1 - \frac{1}{M} \right)^{M-1} \log |I|,$$

where we used that $\frac{w_iM}{T} \left( 1 - \frac{w_i}{T} \right)^{M-1}$ takes its maximum at $w_i = \frac{T}{M}$. The calculation above implies that

$$\log \left( \frac{T}{M} \right)^{|I|^k} \simeq n \left( (1 - \frac{1}{M})^{M-1} \log |I| + \log 3 \right).$$

For the minimum code length we get (by using $\left( \frac{T}{M} \right)^M \leq \left( \frac{T}{M} \right)$)

$$n \geq \frac{M \log \frac{T}{M} + kM \log |I|}{(1 - \frac{1}{M})^{M-1} \log |I| + \log 3}$$

$$= \frac{M \log \frac{T}{M} + kM}{(1 - \frac{1}{M})^{M-1} + \log^3 |I|}.$$
If $|I| \to \infty$ and $\frac{\log T}{\log |I|} \to 0$, then
\[ n(T, M, k) \gtrsim kM \left( 1 - \frac{1}{M} \right)^{1-M}. \]

If, in addition, $M \to \infty$, then
\[ n(T, M, k) \gtrsim kMe, \]
and for the sum-rate
\[ R_{\text{sum}}(T, M) = \frac{kM}{n} \lesssim e^{-1}. \]

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 $C_i = C$ a Reed–Solomon code of parameters $(w, k)$ is applied over $\mathbb{GF}(|I|)$ ($w \leq |I|$). (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, the output of the channel is the erasure symbol $\ast$, so the erroneous positions are known. A Reed–Solomon code of parameters $(w, k)$ can correct up to $w - k$ erasure error.

In Figure 3.2 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.

![Step 1: Encoding of packets by Reed–Solomon code](image1)

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](image2)

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](image3)

Step 3: Packets received with two active users

| $c_2$ | $c_3$ | $c_4$ |

![Step 4: Decoding of packets (correcting erasure errors)](image4)

Step 4: Decoding of packets (correcting erasure errors)

| $c_2$ | $c_3$ | $u_1$ | $u_2$ |

Figure 3.2: Packet communication scheme on a collision channel
Theorem 3.4 (Györfi and Györi (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 $|I| > ek$, then

$$n(T,M,k) \lesssim kM \left(1 - \frac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim \left(1 - \frac{1}{M}\right)^{M-1}.$$  

If, in addition, $M \to \infty$, then

$$n(T,M,k) \lesssim kMe,$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim e^{-1}.$$  

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 0 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 3.1 (Györfi and Györi (2004)). For synchronous access and non-binary packets

$$P\{\text{false identification}\} \leq \exp \left(-(M+1)\ln T - w \left(1 - \frac{w}{n}\right)^M\right). \quad (3.1)$$

Proof. Let us fix at most $M$ interfering protocol sequences (users) and choose an $(M+1)^{\text{th}}$ (tagged) protocol sequence distinctly from the others. The probability that in a fixed segment some of the $M$ interfering protocol sequences have 1 on the same position where the tagged user has 1 (i.e., the tagged user has a covered 1 in a fixed segment) is $1 - \left(1 - \frac{w}{n}\right)^M$. The probability that all the $w$ 1’s of the tagged user are covered by the other $M$ users is $\left(1 - \left(1 - \frac{w}{n}\right)^M\right)^w$, then the probability that there exists a protocol sequence such that all positions of it are cov-
ered by another \(M\) protocol sequences is at most
\[
P\{\text{false identification} \} \leq T \binom{T-1}{M} \left(1 - \left(1 - \frac{w}{n}\right)^M\right)^w \\
\leq T^{M+1} \left(1 - \left(1 - \frac{w}{n}\right)^M\right)^w \\
= \exp\left((M+1) \ln T + w \ln \left(1 - \left(1 - \frac{w}{n}\right)^M\right)\right) \\
\leq \exp\left((M+1) \ln T - w \left(1 - \frac{w}{n}\right)^M\right),
\]
where in the last step we applied that \(\ln(1-x) \leq -x, \forall x \in \mathbb{R}\).

Decoding error occurs if there are less than \(k\) successfully transmitted packets (uncovered 1’s in the protocol sequence) of an active user.

**Lemma 3.2 (Győrgy and Győri (2004)).** For synchronous access and non-binary packets, if \(w \geq \frac{k}{p}\) then
\[
P\{\text{false decoding} \} \leq \exp\left(-\frac{(wp-k)^2}{3wp} + M \ln T\right), \tag{3.2}
\]
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
\[
P\{\text{false decoding} \} \leq T \binom{T-1}{M-1} \sum_{i<k} \binom{w}{i} p^i (1-p)^{w-i}.
\]
Let us apply now Lemma 2.5 for upper bounding the tail of binomially distributed random variable (which is the sum of indicator variables)

\[ \mathbb{P}\{\text{false decoding}\} \leq T \left( \frac{T-1}{M-1} \right) \sum_{i<k} \binom{w}{i} p^i (1-p)^{w-i} \]

\[ = T \left( \frac{T-1}{M-1} \right) \mathbb{P} \left\{ \sum_{i=1}^{w} X_i < k \right\} \]

\[ = T \left( \frac{T-1}{M-1} \right) \mathbb{P} \left\{ \frac{1}{w} \sum_{i=1}^{w} (X_i - \mathbb{E}\{X_i\}) < -\left( p - \frac{k}{w} \right) \right\} \]

\[ \leq \exp \left( -\frac{w(p - \frac{k}{w})^2}{2p(1-p) + \frac{2}{3} (p - \frac{k}{w})} + M \ln T \right) \]

\[ \leq \exp \left( -\frac{w(p - \frac{k}{w})^2}{2p + p} + M \ln T \right) \]

\[ = \exp \left( -\frac{(wp - k)^2}{3wp} + M \ln T \right), \]

where \( X_1, X_2, \ldots, X_w \) are independent indicator random variables with parameter \( p \), and in Lemma 2.5 \( a = 0, b = 1, \sigma^2 = p(1-p) \) and \( \varepsilon = p - \frac{k}{w} \).

\[ \square \]

**Proof of Theorem 3.4.** 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.

Concerning to (1.1) and (1.2) for the decodability property we get by taking the logarithm of (3.2)

\[ -\frac{(wp - k)^2}{3wp} + M \ln T < 0. \]  

(3.3)

The solution of this inequality with respect to positive weight \( w \) is

\[ w > \frac{k}{p} \left( 1 + \alpha \right) \left( 1 + \sqrt{1 - \frac{1}{(1 + \alpha)^2}} \right), \]

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 \( M \)

\[ p = \left( 1 - \frac{1}{M} \right)^{M-1}. \]
If we choose the weight of the protocol sequences \( w \) to
\[
\frac{w}{p} = (1 + \delta) \frac{k}{p}
\]  
for an arbitrary constant \( \delta > 0 \), the exponent in (3.2) 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 \( w \)
\[
P\{\text{false decoding}\} \to 0.
\]
By the choice of (3.4) the exponent in (3.1) become
\[
-k \left( (1 + \delta) \left( 1 - \frac{1}{M} \right) - \frac{(M + 1) \ln T}{k} \right)
\]
which also tends to \(-\infty\) when \( k \to \infty \) and \( \frac{\ln T}{k} \to 0 \), that is why
\[
P\{\text{false identification}\} \to 0,
\]
so there exists a good code \( C \). As the reasoning above is true for all arbitrarily small \( \delta > 0 \),
the next asymptotic upper bound on the minimum weight \( w \) is true:
\[
w \leq \frac{k}{p} = k \left( 1 - \frac{1}{M} \right)^{1-M}.
\]
Finally, we have shown the following asymptotic upper bound on the minimum frame size \( n \):
\[
n(T, M, k) = Mw \lesssim kM \left( 1 - \frac{1}{M} \right)^{1-M},
\]
and for the sum-rate
\[
R_{\text{sum}}(T, M) \gtrsim \left( 1 - \frac{1}{M} \right)^{M-1}.
\]
If, in addition, \( M \to \infty \), then
\[
n(T, M, k) \lesssim kMe,
\]
and for the sum-rate
\[
R_{\text{sum}}(T, M) \gtrsim e^{-1}.
\]
3.3 Bounds for non-binary packets, asynchronous access

As the minimum block length for asynchronous access is lower bounded by the minimum block length for synchronous access, Theorem 3.3 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 $C_i = C$ a Reed–Solomon code of parameters $(w,k)$ is applied over $\text{GF}(|I|)$ ($w \leq |I|$).

For asynchronous access the upper bound on the minimum length of the protocol sequences is the same as for synchronous case.

**Theorem 3.5 (Győrfi and Győri (2005A)).** For asynchronous access and non-binary packets, if $M$ is fixed, $T \to \infty$, $k \to \infty$, $\frac{\log T}{k} \to 0$ and $|I| > ek$, then

$$n(T,M,k) \lesssim kM \left(1 - \frac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim \left(1 - \frac{1}{M}\right)^{M-1}.$$  

If, in addition, $M \to \infty$, then

$$n(T,M,k) \lesssim kMe,$$

and for the sum-rate

$$R_{\text{sum}}(T,M) \gtrsim e^{-1}.$$  

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 0 symbol to bit 0. The resulting 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 3.3 (Győrfi and Győri (2005a)).** For asynchronous access and non-binary packets

$$P\{\text{false identification}\} \leq \exp\left(\frac{2M+1}{2} T + 2M \ln n - w \left(1 - \frac{w}{n}\right)^M\right). \quad (3.5)$$

**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 3.1, 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

$$P\{\text{false identification}\} \leq T^{\frac{T-1}{2M}} n^{2M} \left(1 - \left(1 - \frac{w}{n}\right)^M\right)^w \leq T^{2M+1} n^{2M} \left(1 - \left(1 - \frac{w}{n}\right)^M\right)^w = \exp\left((2M + 1) \ln T + 2M \ln n + w \ln \left(1 - \left(1 - \frac{w}{n}\right)^M\right)\right) \leq \exp\left((2M + 1) \ln T + 2M \ln n - w \left(1 - \frac{w}{n}\right)^M\right)$$

where in the last step we applied that $\ln(1-x) \leq -x$, $\forall x \in \mathbb{R}$. \hfill $\square$

Now, we consider the synchronization task.

**Lemma 3.4 (Győrfi and Győri (2005a)).** For asynchronous access and non-binary packets

$$P\{\text{false synchronization}\} \leq \exp\left(2M \ln T + 2M \ln n - \frac{w}{6} \left(1 - \frac{w}{n}\right)^M\right). \quad (3.6)$$

**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 ($0 \leq s_2 \leq \frac{n}{w} - 1$), and fix some arbitrarily shifted protocol sequences (users) such that there are at most $M$ active ones in every time slot.

We denote by $A_i$ ($1 \leq i \leq w$) 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

$$P\{A_i\} = 1 - \left(1 - \frac{w}{n}\right)^M,$$

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 $c_i$ ($1 \leq i \leq w$) correspond to the encoded packets, so their values tell the position of the 1 in a segment. In Figure 3.3 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 independent classes \( B_j \), \( 1 \leq j \leq 6 \) of \( A_i \)'s can be formed, where \( B_j \cap B_\ell = \emptyset \), and \( \bigcup_{j=1}^6 B_j = \{ A_1, \ldots, A_w \} \).

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 \( c_\ell \). 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.

\[
\Pr \left\{ \bigcap_{i=1}^w A_i \right\} = \Pr \left\{ \bigcap_{j=1}^6 \{ A_i : A_i \in B_j \} \right\} \\
\leq \min_{j=1,\ldots,6} \Pr \left\{ \bigcap_{A_i \in B_j} A_i \right\} \\
\leq \left( 1 - \left( 1 - \frac{w}{n} \right)^M \right)^{\frac{w}{6}},
\]

therefore

\[
\Pr \{ \text{false synchronization} \} \leq T \left( \frac{T-1}{2M-1} \right)^{2M} \left( 1 - \left( 1 - \frac{w}{n} \right)^M \right)^{\frac{w}{6}} \\
\leq T^{2M} \left( 1 - \left( 1 - \frac{w}{n} \right)^M \right)^{\frac{w}{6}} \\
= \exp \left( 2M \ln T + 2M \ln n + \frac{w}{6} \ln \left( 1 - \left( 1 - \frac{w}{n} \right)^M \right) \right) \\
\leq \exp \left( 2M \ln T + 2M \ln n - \frac{w}{6} \left( 1 - \frac{w}{n} \right)^M \right).
\]

Decoding error occurs if there are less than \( k \) successfully transmitted packets (uncovered 1’s in the protocol sequence) of an active user.

**Lemma 3.5 (Györfi and Györi (2005a)).** For asynchronous access and non-binary packets, if \( w \geq \frac{k}{p} \) then

\[
\Pr \{ \text{false decoding} \} \leq \exp \left( -\frac{(wp-k)^2}{3wp} + (2M-2) \ln n + (2M-1) \ln T \right), \quad (3.7)
\]
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 3.2 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
\[
\mathbb{P}\{\text{false decoding}\} \leq T \left(\frac{T-1}{2M-2}\right) n^{2M-2} \sum_{i<k} \binom{w}{i} p^i (1 - p)^{w-i},
\]
and by applying Bernstein’s inequality (Lemma 2.5) for upper bounding the tail of binomially distributed random variable we get
\[
\mathbb{P}\{\text{false decoding}\} \leq \exp \left(\frac{(wp - k)^2}{3wp} + (2M - 2) \ln n + (2M - 1) \ln T\right). \]

**Proof of Theorem 3.5.** 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.

Concerning to (1.1) and (1.2) for the decodability property we get by taking the logarithm of (3.7)
\[
-\frac{(wp - k)^2}{3wp} + (2M - 2) \ln n + (2M - 1) \ln T < 0. \tag{3.8}
\]
If the middle term is temporarily ignored, the solution of this inequality with respect to positive weight \( w \) is
\[
w > \frac{k}{p} \left(1 + \alpha\right) \left(1 + \sqrt{1 - \frac{1}{(1 + \alpha)^2}}\right),
\]
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 \( M \)
\[
p = \left(1 - \frac{1}{M}\right)^{M-1}.
\]
If we choose the weight of the protocol sequences \( w \) to
\[
w = (1 + \delta) \frac{k}{p}
\] for an arbitrary constant \( \delta > 0 \), the exponent in (3.7) becomes
\[
-k \left( \frac{\delta^2}{3(1 + \delta)} - \frac{(2M - 2) \ln \left( \frac{(1 + \delta) kM}{p} \right)}{k} - \frac{(2M - 1) \ln T}{k} \right)
\]
which tends to \(-\infty\) when \( k \to \infty \) and \( \ln T/k \to 0 \), that is why for such a weight \( w \)
\[
P\{\text{false decoding}\} \to 0.
\]
By the choice of (3.9) the exponent in (3.5) becomes
\[
-k \left( (1 + \delta) \left( 1 - \frac{1}{M} \right) - \frac{(2M + 1) \ln T}{k} - \frac{2M \ln \left( \frac{(1 + \delta) kM}{p} \right)}{k} \right)
\]
and the exponent in (3.6) becomes
\[
-k \left( \frac{1 + \delta}{6} \left( 1 - \frac{1}{M} \right) - \frac{2M \ln T}{k} - \frac{2M \ln \left( \frac{(1 + \delta) kM}{p} \right)}{k} \right)
\]
and both of them tend to \(-\infty\) when \( k \to \infty \) and \( \ln T/k \to 0 \), that is why
\[
P\{\text{false identification}\} \to 0,
\]
and
\[
P\{\text{false synchronization}\} \to 0,
\]
so there exists a good code \( C \). As the reasoning above is true for all arbitrarily small \( \delta > 0 \),
the next asymptotic upper bound on the minimum weight \( w \) is true:
\[
w \preceq \frac{k}{p} = k \left( 1 - \frac{1}{M} \right)^{1-M}.
\]
Finally, we have shown the following asymptotic upper bound on the minimum frame size \( n \):
\[
n(T, M, k) \preceq kM \left( 1 - \frac{1}{M} \right)^{1-M},
\]
and for the sum-rate
\[
R_{\text{sum}}(T, M) \geq \left( 1 - \frac{1}{M} \right)^{M-1}.
\]
If, in addition, \( M \to \infty \), then
\[
n(T, M, k) \preceq kMe,
\]
and for the sum-rate
\[
R_{\text{sum}}(T, M) \geq e^{-1}.
\]
From Theorem 3.3 and 3.5 we have the following:

**Corollary 3.3 (Győrfi and Györi (2005a)).** For asynchronous access and non-binary packets, if $M$ is fixed, $T \to \infty$, $|I| \to \infty$, $k \to \infty$, $|I| > e k \frac{\log T}{\log |I|} \to 0$ and $\frac{\log T}{k} \to 0$, then

$$n(T, M, k) \simeq k M \left(1 - \frac{1}{M}\right)^{1-M},$$

and for the sum-rate

$$R_{\text{sum}}(T, M) \simeq \left(1 - \frac{1}{M}\right)^{M-1}.$$

If, in addition, $M \to \infty$, then

$$n(T, M, k) \simeq k M e,$$

and for the sum-rate

$$R_{\text{sum}}(T, M) \simeq e^{-1}.$$
Chapter 4

Collision Channel with Ternary Feedback

4.1 Channel model

In this chapter we consider the multiple access collision channel with feedback (cf. Abramson (1970)). An unlimited number of users are allowed to transmit packets of a fixed length whose duration is taken as a time unit and is called slot. Stations can begin to transmit packets only at time $t \in \{0, 1, 2, 3, \ldots \}$. A slot is a time interval $[t, t+1)$. 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 exchange 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 intensity $\lambda$.

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 is necessary. However, all users, also those who were not transmitting, can learn—from the ternary feedback just before time instant $t+1$—the story of time slot $[t, t+1)$:

- 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 ensure the eventually 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 occurred 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).

**4.2 Tree Algorithm for Collision Resolution**

Independently of each other, Capetanakis (1979), Tsybakov and Mikhailov (1978) introduced the first CRA, called the tree algorithm resulting in a stable conflict resolution protocol.

Let $N$ denote the number of active transmitters, i.e., the multiplicity of the collision. According to the tree 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 tree 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 $\ast$. After this collision, all transmitters involved flip a (non-biased) 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. (The cases, when the transmitters flip a biased and/or non-binary coin, were investigated in every detail; cf. Mathys and Flajolet (1985).)

The algorithm can be represented by a binary rooted search tree. Collisions correspond to intermediate nodes, while empty slots and successful slots correspond to terminal nodes.

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

$$L_N = \mathbb{E}\{Y \mid X = N\},$$

then $L_N$ is the conditional expectation of the collision resolution time, given the multiplicity of the conflict $N$.

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 $L_N/N$:

$$\frac{L_N}{N} = \frac{2}{\ln 2} + A \sin(2\pi \log_2 N + \varphi) + O(N^{-1}),$$

where

$$A = 3.127 \cdot 10^{-6}, \quad \varphi = 0.9826.$$  

These imply that

$$2.8853869 \leq \liminf_{N \to \infty} \frac{L_N}{N} \leq \limsup_{N \to \infty} \frac{L_N}{N} \leq 2.8853932.$$

(4.1)
Introduce the notation
\[ L(z) = \sum_{N=0}^{\infty} L_N \frac{z^N}{N!} e^{-z}. \]

\( L(z) \) is called the Poisson transform of the sequence \( \{L_N\} \) (cf. Szpankowski (2001)).

For evaluating the throughput of the various random access protocols, the calculation of \( L(z) \) (actually \( \frac{L(z)}{z} \)) is very important (cf. Capetanakis (1979), Gallager (1978), Mathys and Flajolet (1985), Kerekes (1988)).

We assumed that new packets arrive according to a Poisson process, therefore \( L(z) \) plays an important role in the analysis. If \( N \) is a random variable with Poisson \( (z) \) distribution, then
\[ L(z) = E \{L_N\}. \]

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 \( L(z) \):
\[ L(z) = E \{L_N\} = E \{I_{(N \leq 1)}\} + E \{I_{(N \geq 2)}(1 + L_{N_1} + L_{N_2})\} = E \{I_{(N \leq 1)}(1 - (1 + L_{N_1} + L_{N_2}))\} + E \{1 + L_{N_1} + L_{N_2}\} = -2E \{I_{(N \leq 1)}\} + E \{1 + L_{N_1} + L_{N_2}\} = -2(e^{-z} + ze^{-z}) + 1 + 2L(z/2) \]
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
\[ L(z) = 1 - 2(1 + z)e^{-z} + 2L(z/2). \]  
(4.2)

The recursive equation (4.2) gives an easy and quick way of calculation of \( L(z) \) numerically.

**Algorithm 4.1 (Győrfi and Győri (2005b)).** For any fixed \( z > 0 \) choose \( k_0 \) such that \( \hat{z} := z/2^{k_0} < 10^{-5}, \) and apply the following iteration:
\[ L(2^k \hat{z}) = 1 - 2(1 + 2^k \hat{z})e^{-2^k \hat{z}} + 2L(2^{k-1} \hat{z}), \]
\( (1 \leq k \leq k_0) \)
then in the \( k_0^{th} \) step we get \( L(z). \) In order to get a good initial value for \( L(\hat{z}) \), we use the second
order Taylor polynomial approximation:

\[ L(\hat{z}) = \left( L_0 + L_1 \hat{z} + L_2 \frac{\hat{z}^2}{2} + \cdots \right) \left( 1 - \hat{z} + \frac{\hat{z}^2}{2} + \cdots \right) \]

\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.

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 4.1). Using Algorithm 4.1, 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 show its oscillation.

**Algorithm 4.2 (Győrfi and Győri (2005b)).** From the equation (4.2) it follows that

\[ F(2z) = \frac{1 - 2(1 + 2z)e^{-2z}}{2z} + F(z), \]

so if \( F(z) \) is given for \( 1 \leq z \leq 2 \) (e.g., using Algorithm 4.1), then \( F(2z) \) can be calculated for \( 1 \leq z \leq 2 \), therefore \( F(z) \) is given for \( 2 \leq z \leq 4 \) 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 \( k \geq 2 \). For large enough \( k \) we can write that for any \( 1 \leq z \leq 2 \)

\[ F\left(2^k z\right) \simeq G(z) + F(z), \quad (4.3) \]
where $k \geq k_0$, and

$$G(z) = \sum_{i=1}^{\infty} \frac{1-2(1+2^i z)e^{-2^i z}}{2^i z} \asymp \sum_{i=1}^{k_0} \frac{1-2(1+2^i z)e^{-2^i z}}{2^i z}.$$ 

If $k \geq k_0 = 30$, then this approximation error is of order $10^{-9}$.

Equation (4.3) gives us an easy way of studying $F(z)$ (see Figure 4.2). We found that

$$2.8853869 \leq \liminf_{z \to \infty} \frac{L(z)}{z} \leq \limsup_{z \to \infty} \frac{L(z)}{z} \leq 2.8853932. \quad (4.4)$$

### 4.3 Oscillation of $L_N - L(N)$

The asymptotic behavior of $L_N$ is mostly investigated in the literature through its Poisson transform $L(z)$. In generally, if the Poisson transform satisfies some conditions, it is very close to the original sequence (cf. Szpankowski (2001) Theorem 10.3), but $L(z)$ fails these conditions. So, the question naturally arises how small is the difference between $L_N$ and $L(N)$ if $N \to \infty$. Mathys (1984) proved that $L_N - L(N) = O(1)$. Next we extend it showing its oscillation.

**Theorem 4.1 (Győrfi and Győri (2005B)).** If $N \to \infty$, then

$$L_N - L(N) \simeq A \cos(2\pi \log_2 N + \varphi),$$

where

$$A = 1.29 \cdot 10^{-4}, \quad \varphi = 0.698.$$ 

**Proof.** 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 $N \geq 2$,

$$L_N = 1 + 2 \sum_{j=0}^{\infty} \frac{2^j (1 - (1 - 2^{-j})^N) - N(1 - 2^{-j})^{N-1}}{1 - 2^{-j}}. \quad (4.5)$$
Let us calculate the Poisson transform of $L_N$ (cf. Mathys and Flajolet (1985)). By (4.5)

$$L(z) = \sum_{N=0}^{\infty} \frac{L_N}{N!} e^{-z}$$

$$= \sum_{N=0}^{\infty} \frac{z^N}{N!} e^{-z} + 2 \sum_{N=2 \to j=0}^{\infty} \left(2^j \left(1 - (1 - 2^{-j})^N\right) - N(1 - 2^{-j})^{N-1}\right) \frac{z^N}{N!} e^{-z}$$

$$= 1 + 2 \sum_{j=0}^{\infty} \left(2^j \left(1 - e^{-2^{-j}z}\right) - z e^{-2^{-j}z}\right). \tag{4.6}$$

By (4.5) and (4.6),

$$L_N - L(N) = 1 + 2 \sum_{j=0}^{\infty} \left(2^j \left(1 - (1 - 2^{-j})^N\right) - N(1 - 2^{-j})^{N-1}\right)$$

$$- 1 - 2 \sum_{j=0}^{\infty} \left(2^j \left(1 - e^{-2^{-j}N}\right) - Ne^{-2^{-j}N}\right)$$

$$= 2 \sum_{j=0}^{\infty} 2^j \left(e^{-2^{-j}N} - (1 - 2^{-j})^N\right) + 2 \sum_{j=0}^{\infty} N \left(e^{-2^{-j}N} - (1 - 2^{-j})^{N-1}\right)$$

$$= 2 \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \left(1 - \left(e^{-j}(1 - 2^{-j})\right)^N\right)$$

$$+ 2 \sum_{j=0}^{\infty} N \left(e^{-2^{-j}N}e^{-2^{-j}(N-1)} - (1 - 2^{-j})^{N-1}\right)$$

$$=: 2A + 2B.$$

For getting lower and upper bounds we use the following inequalities. If $0 \leq x \leq 1$, then

$$1 + x + \frac{x^2}{2} \leq e^x \leq 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$

$$1 - \frac{x^2}{2} - \frac{x^3}{6} \leq e^x(1 - x) \leq 1 - \frac{x^2}{2} - \frac{x^3}{6} \leq 1 - \frac{x^2}{2}$$

$$1 - x \leq e^{-x} \leq 1 - x + \frac{x^2}{2}$$

and if $a \geq b \geq 0$, then

$$(a - b)Nb^{N-1} \leq a^N - b^N \leq (a - b)Na^{N-1}.$$

Lower bound:

$$A \geq \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \left(1 - \left(\frac{2^{-2j}}{2}\right)^N\right)$$

$$\geq \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} 2^{-2j} \left(1 - \left(\frac{2^{-2j}}{2}\right)^N\right)$$

$$\geq \frac{1}{2} \sum_{j=0}^{\infty} 2^{-j} Ne^{-2^{-j}N} \left(1 - (N-1)\frac{2^{-2j}}{2}\right).$$
\[
\begin{align*}
\geq & \quad \frac{1}{2} \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} \left( 1 - N \frac{2^{-2j}}{2} \right) \\
= & \quad \frac{1}{2} \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} - \frac{1}{4} \sum_{j=0}^{\infty} 2^{-3j} N^2 e^{-2^{-j}N} \\
=: & \quad I_1 + I_2,
\end{align*}
\]

and

\[
B \geq \sum_{j=0}^{\infty} N \left( (1 - 2^{-j}) e^{-2^{-j}(N-1)} - (1 - 2^{-j})^{N-1} \right) \\
= \sum_{j=0}^{\infty} N e^{-2^{-j}(N-1)} \left( 1 - \left( e^{2^{-j}} (1 - 2^{-j}) \right)^{N-1} \right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)} \\
\geq \sum_{j=0}^{\infty} (N - 1) e^{-2^{-j}(N-1)} \left( 1 - \left( 1 - \frac{2^{-2j}}{2} \right)^{N-1} \right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)} \\
\geq \frac{1}{2} \sum_{j=0}^{\infty} 2^{-2j} (N - 1)^2 e^{-2^{-j}(N-1)} \left( 1 - \frac{2^{-2j}}{2} \right)^{N-1} - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)} \\
\geq \frac{1}{2} \sum_{j=0}^{\infty} 2^{-2j} (N - 1)^2 e^{-2^{-j}(N-1)} \left( 1 - (N - 1) \frac{2^{-2j}}{2} \right) - \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}(N-1)} \\
= \frac{1}{2} \sum_{j=0}^{\infty} 2^{-2j} (N - 1)^2 e^{-2^{-j}(N-1)} - \frac{1}{4} \sum_{j=0}^{\infty} 2^{-4j} (N - 1)^3 e^{-2^{-j}(N-1)} \\
- \sum_{j=0}^{\infty} 2^{-j} (N - 1) e^{-2^{-j}(N-1)} - \sum_{j=0}^{\infty} 2^{-j} e^{-2^{-j}(N-1)} \\
=: & \quad J_1 + J_2 + J_3 + J_4.
\]

Upper bound:

\[
A \leq \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \left( 1 - \left( 1 - \frac{2^{-2j}}{2} - \frac{2^{-3j}}{2} \right)^N \right) \\
\leq \sum_{j=0}^{\infty} 2^j e^{-2^{-j}N} \frac{2^{-2j}}{2} (1 + 2^{-j}) N \\
= \frac{1}{2} \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} (1 + 2^{-j}) \\
= \frac{1}{2} \sum_{j=0}^{\infty} 2^{-j} N e^{-2^{-j}N} + \frac{1}{2} \sum_{j=0}^{\infty} 2^{-2j} N e^{-2^{-j}N} \\
=: & \quad I_1 + I_2,
\]

and

\[
B \leq \sum_{j=0}^{\infty} N \left( (1 - 2^{-j} + \frac{2^{-2j}}{2}) e^{-2^{-j}(N-1)} - (1 - 2^{-j})^{N-1} \right)
\]
It can be further simplified by showing that

\[ 2I_1 + J_3 = \sum_{j=0}^{\infty} 2^{-j}N \left( e^{-2^{-j} - 1} \right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j}e^{-2^{-j}(N-1)} \]

\[ \geq \sum_{j=0}^{\infty} 2^{-j}N \left( 1 - 2^{-j} - 1 \right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j}e^{-2^{-j}(N-1)} \]

\[ = -\sum_{j=0}^{\infty} 2^{-2j}Ne^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j}e^{-2^{-j}(N-1)}, \]

and then upper bound it

\[ 2I_1 + J_3 \leq \sum_{j=0}^{\infty} 2^{-j}N \left( 1 - 2^{-j} + \frac{2^{-2j}}{2} - 1 \right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j}e^{-2^{-j}(N-1)} \]

\[ = -\sum_{j=0}^{\infty} 2^{-2j}N \left( 1 - \frac{2^{-j}}{2} \right) e^{-2^{-j}(N-1)} + \sum_{j=0}^{\infty} 2^{-j}e^{-2^{-j}(N-1)}. \]
Notice that both bounds tend to 0. That is why $L_N - L(N)$ asymptotically equals to

$$L_N - L(N) = 2I_1 + J_3 + o(1)$$

$$= \sum_{j=0}^{\infty} 2^{-2j(N-1)}e^{-2^{-j}(N-1)} - \sum_{j=0}^{\infty} 2^{-j}(N-1)e^{-2^{-j}(N-1)} + o(1)$$

$$=: \Delta(N-1) + o(1)$$

The technique being used here is Mellin transform (cf. Szpankowski (2001)). Reader can find an excellent survey on Mellin transform in Flajolet et al. (1995), and applications of Mellin transform to similar problems in Knuth (1973) pages 131–134, and Jacquet and Regnier (1986). The Mellin transform of a complex valued function $f(x)$ defined over positive reals is

$$M[f(x); s] = F(s) = \int_0^\infty x^{s-1}f(x)\,dx, \quad a < \Re(s) < b,$$

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

$$f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s}F(s)\,ds, \quad a < c < b,$$

where $c$ is an arbitrary real number from the fundamental strip $(a, b)$.

One of the basic properties of the Mellin transform is that if

$$M[f(x); s] = F(s), \quad a < \Re(s) < b,$$

then

$$M[\alpha x^\beta f(\gamma x); s] = \alpha \gamma^{-s}F(s+\beta), \quad a - \beta < \Re(s) < b - \beta. \quad (4.7)$$

If we consider an elementary Mellin transform (cf. Szpankowski (2001) page 401):

$$M[e^{-x}; s] = \Gamma(s), \quad 0 < \Re(s) < \infty, \quad (4.8)$$

then from (4.7) and (4.8) we have

$$M[x^2e^{-x}; s] = \Gamma(s+2), \quad -2 < \Re(s) < \infty, \quad (4.9)$$

$$M[xe^{-x}; s] = \Gamma(s+1), \quad -1 < \Re(s) < \infty, \quad (4.10)$$

where $\Gamma(\cdot)$ denotes the complete gamma function that is in Euler’s limit form (cf. Szpankowski (2001) page 41):

$$\Gamma(s) = \lim_{n\to\infty} \frac{n^s n!}{s(s+1)(s+2)\cdots(s+n)}. \quad (4.11)$$

By applying Mellin transform technique the difference $\Delta(N)$ can be expressed in terms of the gamma function. Thus, with using (4.9) and (4.10) (while considering property (4.7)) we
have

\[
\mathcal{M}[\Delta(N);s] = \sum_{j=0}^{\infty} \mathcal{M}[2^{-2j}N^2e^{-2^{-j}/N};s] - \sum_{j=0}^{\infty} \mathcal{M}[2^{-j}Ne^{-2^{-j}/N};s]
\]

\[
= \sum_{j=0}^{\infty} (2^{-j})^{-s} \Gamma(s+2) - \sum_{j=0}^{\infty} (2^{-j})^{-s} \Gamma(s+1)
\]

\[
= \frac{\Gamma(s+2)}{1-2^s} - \frac{\Gamma(s+1)}{1-2^s},
\]

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. For this calculation some residues are needed (cf. Figure 4.3). \(\frac{1}{1-2^s}\) has simple poles at the roots of equation \(2^s = 1\), so if \(s = \frac{2k\pi i}{\ln 2}\)

\[
\text{res}_{s=s_0} \frac{1}{1-2^s} = \lim_{s \to s_0} \frac{1}{(1-2^s)'} = \lim_{s \to s_0} \frac{1}{\ln 2} = -\frac{1}{\ln 2},
\]

for all \(s_0 \in \{ \frac{2k\pi i}{\ln 2}, \ k \in \mathbb{Z} \}\).

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

\[
\Delta(N) = -\frac{1}{2\pi i} \left( 2\pi i \sum_{k=-\infty}^{\infty} \text{res}_{s=s_0} N^{-s} \left( \frac{\Gamma(s+2)}{1-2^s} - \frac{\Gamma(s+1)}{1-2^s} \right) \right).
\]
\[ L_N - L(N) = \Delta(N - 1) + o(1) \]
\[ \simeq \frac{1}{\ln 2} + \frac{1}{\ln 2} \sum_{k \neq 0} \Gamma \left(2 + \frac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2(N-1)} \]
\[ - \left( \frac{1}{\ln 2} + \frac{1}{\ln 2} \sum_{k \neq 0} \Gamma \left(1 + \frac{2k\pi i}{\ln 2}\right) e^{-2k\pi i \log_2(N-1)} \right) \]
\[ = \frac{1}{\ln 2} \sum_{k \neq 0} \left( \Gamma \left(2 + \frac{2k\pi i}{\ln 2}\right) - \Gamma \left(1 + \frac{2k\pi i}{\ln 2}\right) \right) e^{-2k\pi i \log_2(N-1)} \quad (4.12) \]

As the gamma function decays exponentially fast over imaginary lines, for \( L_N - L(N) \) a sharp approximation can be given by (4.12) 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 \( 10^{-9} \).

\[ L_N - L(N) \simeq A \cos(2\pi \log_2(N - 1) + \varphi), \]

where
\[ A = \frac{2}{\ln 2} \sqrt{\left( \Re \left( \Gamma \left(2 + \frac{2\pi i}{\ln 2}\right) - \Gamma \left(1 + \frac{2\pi i}{\ln 2}\right) \right) \right)^2 + \left( \Im \left( \Gamma \left(2 + \frac{2\pi i}{\ln 2}\right) - \Gamma \left(1 + \frac{2\pi i}{\ln 2}\right) \right) \right)^2} \]
\[ = 1.29 \cdot 10^{-4}, \]

and
\[ \varphi = \arctg \frac{\Im \left( \Gamma \left(2 + \frac{2\pi i}{\ln 2}\right) - \Gamma \left(1 + \frac{2\pi i}{\ln 2}\right) \right)}{\Re \left( \Gamma \left(2 + \frac{2\pi i}{\ln 2}\right) - \Gamma \left(1 + \frac{2\pi i}{\ln 2}\right) \right)} = 0.698. \]

Thus
\[ |L_N - L(N)| \lesssim A = 1.29 \cdot 10^{-4}. \]

In order to finish the proof we have to show that if \( N \to \infty \), then
\[ \cos(2\pi \log_2 N + \varphi) - \cos(2\pi \log_2(N - 1) + \varphi) = o(1). \]

It is easy since
\[ \cos(2\pi \log_2 N + \varphi) - \cos(2\pi \log_2(N - 1) + \varphi) \]
\[ = (\cos(2\pi \log_2 N) - \cos(2\pi \log_2(N - 1))) \cos \varphi \quad (4.13) \]
\[ - (\sin(2\pi \log_2 N) - \sin(2\pi \log_2(N - 1))) \sin \varphi. \quad (4.14) \]

Then the first term of (4.13) can be written as
\[ \cos(2\pi \log_2 N) - \cos(2\pi \log_2(N - 1)) \]
\[ = -2 \sin \left( \pi (\log_2 N + \log_2(N - 1)) \right) \sin \left( \pi (\log_2 N - \log_2(N - 1)) \right) \]
\[ = -2 \sin \left( \pi \log_2 \left(\frac{N}{N-1}\right) \right) \sin \left( \pi \log_2 \left(1 + \frac{1}{N-1}\right) \right). \]

As the function \( \log_2(\cdot) \) is continuous in 1 and function \( \sin(\cdot) \) is continuous in 0
\[ \lim_{N \to \infty} \sin \left( \pi \log_2 \left(1 + \frac{1}{N-1}\right) \right) = 0, \]
that is why (4.13) is $o(1)$. With similar reasoning it can be easily seen that (4.14) is also $o(1)$. So, we have proved that

$$L_N - L(N) \simeq A \cos(2\pi \log_2 N + \varphi).$$

From Theorem 4.1 it follows that

$$\lim_{N \to \infty} \left( \frac{L_N}{N} - \frac{L(N)}{N} \right) = 0$$

which gives reasoning to the equality of lower and upper bounds of $\frac{L_N}{N}$ in (4.1) and $\frac{L(z)}{z}$ in (4.4).
Bibliography


Hajek, B. (1980). Expected number of slots needed for the Capetanakis collision-resolution algorithm. unpublished manuscript, Coordinated Sci. Lab., Univ. of Illinois, Urbana IL.


