Doctoral Thesis

ERROR-CORRECTING CODES IN APPLICATION TO DIGITAL MULTIMEDIA TRANSMITTING AND STORAGE

Eng. Konstantin Shinkarenko

Supervisor: Professor Ing. Karel Vlček, CSc.

TOMAS BATA UNIVERSITY IN ZLIN
Faculty of Applied Informatics
Department of Computer and Communication Systems

Zlin, 2010

Acknowledgments

I would like to express my gratitude to my supervisor, Professor Ing. Karel Vlček, CSc., whose expertise added considerably to my experience. I appreciate his vast knowledge and skill he shared in professional areas, and his assistance in writing papers and the thesis.

A very special thanks goes out to Tatiana Burykhina, who provided consultancy in English stylistics.

I would also like to thank my wife Maria for the support she provided me: without her love, encouragement, patience and editing assistance, I would not have finished this thesis.

Keywords: forward error correction (FEC), error-correcting codes, digital multimedia, erasure codes, Internet Protocol Television (IPTV), packet losses, multimedia broadcasting.

Abstract

The present research is aimed at the problem of error-free transmitting and storage of digital multimedia data. The branch of science which studies methods of error errors correcting is called coding theory.

The existing methods available in coding theory are investigated. The thesis provides a classification of error-correcting codes, analysis of their abilities to correct certain types of errors. Convolutional, block and erasure (Luby Transform) codes are reviewed. The conclusion is done, that erasure codes are the most suitable error-correcting codes for digital multimedia application.

Author performs critical analysis of existing methodology of erasure coding effectiveness estimation. An attempt to systematize the existing methods and introduce a new approach is done.

The research results available up to the present moment are presented. The results showed that the designed NEC (Non-random Erasure Codes) over performs known codes in sense of packet loss recovery effectiveness and robustness.

It is concluded, that there is a possibility to further improve the erasure coding effectiveness for enriching of perceptual quality of digital multimedia data being transmitted or stored in a error-prone environment.

List of Tables

List of Figures

List of Abbreviations

IPTV - Internet Protocol TeleVision

RF - Radio Frequency

TCP - Transport Control Protocol

ARQ - Automatic Repeat reQuest

UDP - User Datagram Protocol

RTP - Real Time Protocol

DVB - Digital Video Broadcasting

BCH codes - Bose-Chaudhuri-Hocquengham Codes

MAP - Maximum a Posterior Algorithm

RTCP - Real-time Control Protocol

LT - Luby Transform codes

MPEG - Motion Picture Expert Group

MP4 - Multimedia Container Format

MXF - Material Exchange Format

ATSC - Advanced Television Systems Committee set of standards

AAC - Advanced Audio Coding

TS - Transport Stream

MPTS – Multi Program Transport Stream

SPTS – Single Program Transport Stream

VOD - Video On Demand

RFC - Request for Comments standard

GOP - Group of Picture

CABAC - Context-Adaptive Binary Arithmetic Coding

CAVLC - Context-Adaptive Variable-Length Coding

NAL - Network Abstraction Level Determination

SPS - Sequence Parameter Sets

PPS - Picture Parameter Sets

FMO - Flexible Macro-block Ordering

MDCT - Modified Discrete Cosine Transformation

PID - Packet Identifier

PCR - Program Clock Reference

PAT - Program Association Table

PMT - Program Map Table

UDP - User Datagram Protocol

PLR - Packets Loss Ratio

SEC - Systematic Erasure Codes

NEC - Non-random Erasure Codes

MTU - Maximum Transfer Unit

PES - Packetized Elementary Stream

PCR - Program Clock Reference

T-STD - Transport System Target Decoder

PAT - Program Association Table

PMT - Program Map Table

XOR - Exclusive OR Operation

Table of Contents

Goals of the Research

The research is undertaken for purposes of providing forward error correction for transmitting of digital multimedia data in live TV systems. It seeks to find existing or to design a new error-correcting code, or concatenation of codes, suitable for solving of actual tasks being faced in up-to-date multimedia broadcasting systems, including broadcasting over IP networks.

The final goal can be formulated as reducing of errors and losses in the digital multimedia streams being transmitted.

Subjects of the research:

- digital multimedia compression standards and container formats, being used in professional broadcasting systems and for multimedia data storage;
- digital multimedia broadcasting technologies and storage devices;
- multimedia data transmitting channels, specificity of errors typical for transmitting channels;
- error-correcting codes which could be utilized in the multimedia broadcasting systems;
- exploring of the requirements and limits on the maximum end-to-end delay, bit-error rates, overhead caused by error-coding (encoding rates), allowed in the concrete multimedia systems;
- error-correcting codes effectiveness criteria: in general and in application to digital multimedia data;

Research objectives:

- theoretical design of a new error-correcting code or enhancing of existing code, capable to provide the protection of digital multimedia data from the errors typical for actually used transmitting channels and storage devices;

- software implementation of the designed error-correcting code and the analogous approaches;
- establishing of a framework for effectiveness estimation of the implemented coding approaches;
- design and development of a technology for adopting and integrating of the error-correcting codes matching the target demands and requirements of multimedia systems in the existing practical applications and systems;
- performance estimation of the practical implementations.

1 Theoretical review of the Research Subject

1.1 Calls for the research

Television is one of the most important inventions of XX-th century. It plays a great role in a life of almost every people on the Earth. Nowadays, we are witnesses of replacing of the traditional analogous television by digital television. The 'digital television' is not only the systems of straight delivery of TV signal to the customer's receiver devices (TV sets) from a satellite, or by cable, or by terrestrial radio channel. Watching of multimedia written on CD, DVD, BluRay disks is also a 'television'.

One of the most intensively developing branches of digital television is delivering of TV signal over IP (IPTV, Internet Protocol TeleVision) networks or Internet (Internet Television). IP networks are used as a medium for signal delivery instead of “traditional” broadcasting of TV signal in Radio Frequency (RF) networks.

Currently, the digital TV is a huge high-tech industry. Thousands of companies, universities, research groups and individuals are working in this industry, being involved in the process of designing of new devices, technologies, consumer and professional hardware and software, compression standards, multimedia data processing algorithms, offering of video services, content and advertisement creation, and so on and so on.

One of the most important tasks of telecommunication engineers is to provide as high perceptual quality of the digital programs as possible, and to make watching of digital TV the best comfortable and convenient for the audience.

Each of the mentioned above mediums of TV signal delivery is error-prone. Radio channels could be affected by atmospheric noise, electromagnetic interference, cable connections are affected by signal bounces, wave reflections, additional resistance at lays, and other physical affects. Laser disks could be scratched or get dirty. Storage of data on hard-drives is also risky – because of mechanical damages or electromagnetic coupling bad blocks could occur, or the hard drive could become completely unreadable. Such effects lead to distortions or losses of the information being transmitted (stored).

It calls to the designing of certain methods of data protection against errors and losses.

The discipline aimed at error detection and correction is called *coding theory* [6, 25]. It is a cross-border discipline of mathematics, computer science, telecommunication and information theory.

Since 1950s, starting from the famous Shannon's work [25], many different error-correcting codes and approaches were introduced. Error-correcting codes are successfully used in digital TV systems, providing data protection in satellite, terrestrial, cable TV systems [30, 37, 49, 51, 52]. However, TV over IP technology, which uses Internet protocol connections for digital multimedia data delivery to the audience, emerged the new calls to the coding theory. Namely, the main problem is complete packet losses, having a place in IP networks. One could object, that TCP connection guarantees the delivery. But TV systems are very sensible to any delays in a channel, thus ARQ (Automatic Repeat reQuest) scheme being in use for TCP protocol is not suitable for multimedia transmitting. Thus the low-delay (but non-guarantee delivery) protocols UDP [70] (User Datagram Protocol) and RTP [65] (Real Time Protocol) are utilized in TV over IP systems.

The present research is aimed at finding of the error-correcting approaches suitable resolving of actual problems of digital multimedia data transmitting and storage.

1.2 Problems of error-correcting coding for digital television systems

Considerable volumes of data in present day data transmission and storage systems, most of which are quite sensitive to errors, make error correction a necessity. The aim of error-correcting codes is to error-proof digital multimedia data throughout the process of their transmission along communication networks. The primary means to provide a high level of error-immunity in a complex system is to impart to it some redundancy required to detect and correct the errors that occur in its operation. Introducing redundancy has to follow a certain code. For the codes to be highly effective they have to be lengthy, for in this case the noise impact gets averaged over a large number of symbols [25]. The task of code designing, as a rule, is resolved by means of a code of a specific mathematical structure allowing for error correction. Knowledge of the mathematical code structure is needed to make the encoding and decoding operations realizable by electronic equipment. Thus, the three basic aspects to the coding problem are:

- designing the codes able to provide proper error correction;
- developing a practically implementable coding method;
- finding an implementable method for decision making at the receiver (decoding).

The use of error-correcting coding is necessitated by the fact that virtually all physical data transmission media have noises causing data loss and distortion.

Basically, the following types of data storage and transmission media used in modern day digital communication systems can be referred to:

- cable connections;
- fiber-optic connections;
- atmosphere;
- outer space;
- laser disks;
- disk storage devices.

Along with it, quite frequently complex communication systems utilize different types of media at different stages.

Every type of medium has its specific kind of noise of its own nature and origin. In terrestrial or satellite radio communication systems these may be lightning and high voltage power transmission lines generating burst interference. Atmospheric interference and cosmic electromagnetic radiation generate Gaussian interference that leads to single bit errors. In cable communication lines the errors are signal attenuation due to resistance of medium, end reflection, network congestion. For laser disks and disk storage devices mechanical damage, scratches are common.

Data packet losses and end-to-end delays in computer networks occur due to switching congestion, throughput channels slump caused by overly large number of simultaneous users, single bit distortion in a packet and checksum discrepancy, network input buffer overflow, etc. In real-time data transmission systems and network applications packet delays equal to losses, as the data processing cannot be suspended waiting for a packet.

The problem of packet losses is considered to be one of the most urgent issues in TV over IP systems. Data losses may lead to degraded TV program perceptual quality: loss of sound, image scattering into irritating „tiny squares", desynchronized sound and image. Thus, for live TV broadcasting non-guarantee delivery protocols have to be utilized, such as UDP and RTP protocols, since retransmitting of packet lost becomes impossible due to long delays for the operation.

Packet delays are intolerable in multimedia stream broadcasting systems, for the delays will cause freezing of video image, and, as a result, real astronomical time lagging. Furthemore, retransmission requests, particularly in one-to many topology, will lead to additional communication channel overloading, which will result in degraded sevice. Classical error-correcting codes are unable to resolve the problem stated. Convolutional codes are capable of correcting single-bit errors. Block codes, Reed-Solomon codes including, can correct bursts of errors in a packet, however, when a complete packet is lost, block codes are helpless.

The demand for multimedia content services is overwhelming, thus, the TV over IP systems development and implementation is of highest commercial value. This explains an extremely high competition in the TV over IP system providers market.

Digital TV broadcasting presents in itself quite a complex technical task, with different fields of science being involved: information theory, mathematics, optics, radio communication theory.

Error-correcting encoding and decoding implementation in TV over IP systems is closely connected with audio and video data compression and multiplexing. Thus, representing a complex approach to TV over IP systems development, error-correcting coding pursues several aims:

- To provide high error-correcting effectiveness that would enhance the TV programs perceptual quality;

- economical transmission band usage. This allows for a larger number of TV programs broadcasted through a single physical channel. The result can make launching a new costly satellite for adding new broadcasting channels unnecessary;

- decoding algorithms optimization aimed at reduced costs of computational resources at the receiver.

At the moment dozens of various codes and decoding algorithms are available, however, their application details for TV over IP systems are not sufficiently dealt with. Because of the high economic value of this research area the findings are rarely get published.

The majority of research effort in this field is dedicated to searching for some "optimal" code that would resolve the basic task of error-correcting coding.

An optimal in this sense code, however, can hardly be considered optimal for TV over IP applications. For TV over IP an error-correcting code has to be comparatively short, and its encoding should not introduce any lengthy delays, while its decoding should not require a large memory capacity and costly processor resources.

1.3 Key terminology

The terminology used should require some clarification. The fundamental classical papers on error-correcting theory adopted the term of “information symbol” to denote an atomic information block that is subjected to encoding. Correspondingly, the symbol obtained as a result of encoding is termed “a coding symbol”. The engineers and researchers involved in DTV systems and IP networks data transmission apply, as a rule, the term of a “packet” to denote this “information atom”. The present paper is using both terms depending on the context: the term of a “symbol” is being used in the context common both for error-correcting coding and data transmission systems, while the term of a “packet” means that IP and/or IPTV networks are mentioned.

As the term “IPTV” is primarily used for indicating of quite specific way of multimedia delivery to consumers, and it does not include broadcasting of multimedia data over Internet, in the present work author uses more generic term “TV over IP”. The term IPTV is used explicitly when it's necessary to emphasize the particular context.

The terms of “TV over IP”, “DTV” (digital television), “DVB” (digital video broadcasting) refer to different notions. TV over IP refers to a particular notion within a more general term of DTV. Thus, when the features of an object mentioned are shared by all the digital broadcasting systems, the term of “DTV” is applied. When there is any specificity attached to the TV over IP systems only, then only this term has to be used. Term “DVB” should be used only to mean Digital Video Broadcasting systems: satellite, terrestrial, cable, as defined by ETSI specifications [51, 52].

The author has also observed a slight discrepancy between the terms of “a convolutional code”, “ a tree code”. In the publications [6, 25, 27] tree codes are positioned to be the most particular instances of semi-infinite codes. Convolutional codes are described as a particular case of tree codes. However, the terminology seems to have later acquired somewhat different meaning. The term of "tree codes" is quite rarely used nowadays, while the term of "convolutional codes" is applied to denote a general class of those codes. That might be connected with the fact, that convolutional codes as a subclass of tree codes used to have the most practical application, and due to this the tree codes research and the terminology, respectively, have been sidelined.

1.4 The history of error-correcting coding

In 1948 Shannon published his classical fundamental theorem for a noisy channel [25]. The theorem sates that encoded information can be transmitted through a communication channel with an arbitrarily small specified error probability at the receiver's end, but the data transmission speed cannot exceed a certain limit value, defined by:

$$M = H \cdot \log_2(1 + S/N), \tag{1.1}$$

where M is a maximum channel capacity,

H is a channel transmission width

S/N is a signal power to channel noise power ratio.

The most significant result of Shannon theorem is that coding allows to achieve data transmission at "an arbitrarily small specified error probability. Designing a code permitting to obtain the result postulated in the theorem has been made possible in principle.

Advance in aerospace field was the factor that stimulated research in this area to no small degree. Satellite communication systems are forced to operate under the conditions of high transmission medium noise level along with the narrow transmission bandwidth and low signal drive. This called for highly effective (even if costly and complicated to implement) methods for error-correcting encoding and decoding.

At the outset of error-correcting theory development a comparatively simple linear algebra apparatus was applied to describe the codes. The codes designed with the help of this apparatus became known as linear codes. The first linear error-correcting code was put forward by Hamming [15]. The code allowed for single-bit error correction..

Further research in this field lead to a booming growth. The mathematical foundation of the theory was considerably expanded, having allowed for more effective codes to be developed.

In 1955 Elias [10] introduced convolutional codes that could be described by a coding device (coder), a finite state machine with specified transition. A soviet scientist Zigangirov K. Sh. made a significant contribution into convolutional codes development.

In 1959 Hegelbeger developed recurrent codes, which were in fact prototypes of Turbo codes [25]. For their definition, similar to convolutional codes, a description of a coding device was used.

In 1960 Bose, Chaudhuri and Hocquengham (Bose-Chaudhuri-Hocquengham Codes) [6, 25] described a broad class of cyclical codes. Cyclical codes apply propositions of Galois fields. A cyclic code is set as a polynomial over a finite field.

An important special case of Bose-Chaudhuri-Hocquengham Codes are the Reed-Solomon codes. Reed-Solomon Codes are widely used in the multimedia data storage and transmission systems.

Invention of Turbo-codes became a significant event in the last decade of the 20-th century [5]. Turbo-codes are a combination of parallel concatenation of convolutional encoders and a pseudo-random interleaver. Quite soon Turbo-codes found application for digital multimedia broadcasting [39].

Thus, by the beginning of the 21st century a mathematical apparatus to describe error-correcting codes was shaped, and the three fundamental code classes were formed. The error-correcting coding theory rests upon the following mathematical base:

- linear algebra;
- Galois finite fields theory; (cyclical codes: Bose-Chaudhuri-Hocquengham Codes, Reed-Solomon Codes, Fire codes, etc.)
- Finite state automatons, or finite state machines (convolutional, recurrent, Turbo-codes).

There are codes that may be described by means of various mathematical apparatuses. For example, Hamming codes belong both to the class of linear codes and cyclical ones. Both cyclical and convolutional codes may be described by logic schemes using memory elements and summators.

In general error-correcting codes can be divided into two classes according to the principle of encoded data conversion: block codes and convolutional codes (Turbo-codes including). Block codes, cyclical and linear codes including, juxtapose to each input data symbol of a specified length an encoded symbol, whose length is also fixed. Each symbol is encoded separately.

Convolutional, as well as Turbo and recurrent codes belong to a class of semi-infinite codes, as the input data sequence is split into blocks of a fixed length, being continuously encoded. The meaning of encoded symbols in the output sequence depends both on the meaning of current input symbols and the state of the encoder, i.e. the meanings of previous symbols. Convolutional codes are designed to correct single-bit errors in a data sequence.

It is worth mentioning the issue of developing decoding algorithms for error-correcting codes. Decoding error-correcting codes presents quite a complicated task. The effectiveness of its solution is of great importance, specifically in digital multimedia broadcasting, one of the areas where error-correcting coding is applied. It is connected with the requirement for the simplicity and use of low power processors for digital TV signal receiver devices incorporating a decoder. Otherwise, the cost of the receiving devices may become unaffordable for an average customer.

The first effective decoding algorithm for BCH codes was offered by W. Peterson [25].

In the late 1950-s the algorithms for sequential decoding of convolutional codes were developed. The authorship of the original paper published in 1957 belonged John Wozencraft [47].

Interestingly, it was not until 1967 that the simplest decoding algorithm for these codes was discovered by Viterbi [41, 42]. However, the original Viterbi algorithm has its imperfection connected with the difficulty of its practical implementation for powerful convolutional codes. This suggested an idea to researchers to optimize Viterbi algorithm, and to further the search for decoding algorithms.

Thus, in 1974 a MAP (*maximum a posteriori)* algorithm was offered by Bahl L. R. [4]. As it turned out later, though, MAP algorithm features the performance inferior to that of Viterbi algorithm when decoding convolutional codes. In spite of this, MAP algorithm has become a key one for Turbo-codes decoding.

A decoder has to be able to take into consideration the specifics of the signal transmitted. For example, a decoding algorithm for modem communication does not suit for voice communication systems. For this reason the search for novel decoding algorithms and improvement of the currently used ones is still going on.

However, due to Internet overall penetration and an ever increasing role of data transmission, digital multimedia broadcasting using protocols with non-guaranteed delivery (*UDP* / *RTP* / *RTCP),* the error-correcting coding theory reached a dead end – the existing error-correcting codes failed to manage the problem of complete packet losses – so-called packet erasure [28, 34].

It was not until 2002 when the scientist M. Luby pioneered a new class of codes that were given the name of erasure codes [19]. The codes were called in inventor's honor "Luby Transform" (LT) codes. He was also the one to develop an LT codes decoding algorithm. Erasure codes got an additional name of "fountain codes" [20, 23, 24].

Erasure codes are designed to be applied in IP packet networks to recover packet loss during transmission. The erasure code theory relies in its basis on the probability theory principles, bringing into play linear algebra apparatus.

1.5 Digital television: history and current state

Technical task of digital multimedia data broadcasting presents an extremely complicated job and is far from finding solutions for the problems it faces, leaving a vast field for further research. Concerted efforts of scientists and engineers, theoretical as well as practical, are required to implement digital multimedia broadcasting systems providing a high level of data transmission reliability and excellent perceptual quality at every stage: audio and video signal capture and digitization, compression, multiplexing, error-correcting coding, data transmission, decoding and playback.

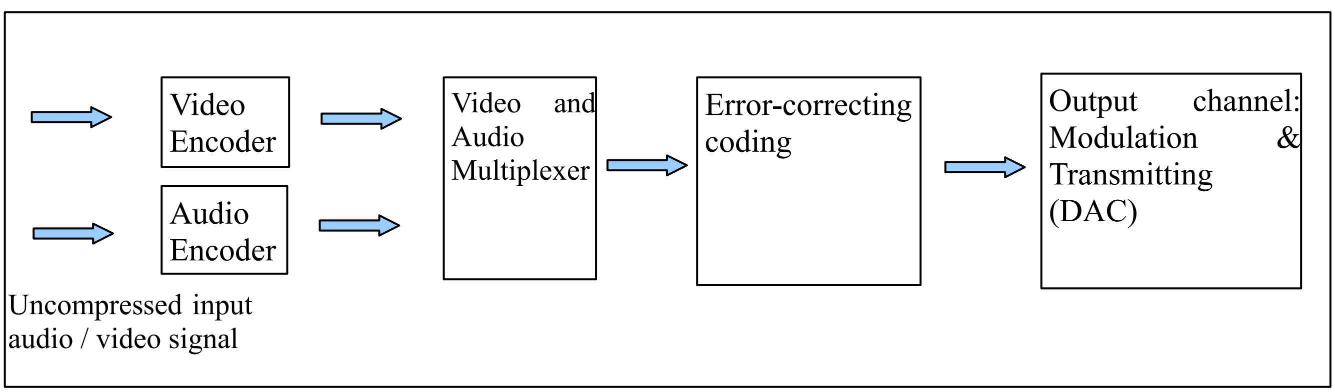

Fig. 1.1. Generic architecture of a nowadays DTV system

The history of digital TV development has numbered no more than two decades. In the early 1990-s a group of MPEG developers adopted the basic standards for streaming

TV broadcasting. It has to be mentioned that in the present research only formats and systems applicable to broadcasting television (streaming formats) will be discussed. File formats, such as MP4, MXF and others, designed for multimedia data file backup are outside of the present research subject. Fig. 1.1. shows a generic architecture of present day DVB system.

The first streaming standard is MPEG-1 developed by Motion Picture Expert Group, a group of developers [52, 53]. MPEG-1 multiplexing standard included guidelines on BCH codes application [8].

In November 1994 International Organization for Standardization adopted ISO 13818-1 [55], 13818-2 [56] and 13818-3 [57] standards that regulated the MPEG-2 transport stream format for digital video and prorated audio transmission, and video MPEG-2 and audio MPEG Layer II, correspondingly. The same year Digital Video Broadcasting (DVB) consortium adopted the first standards for digital video broadcasting based on the ISO 13818 standard - DVB-Satellite, DVB-S for satellite broadcasting and DVB-Cable, DVB-C for cable broadcasting. Three years later DVB-Terrestrial, DVB-T standard was adopted for terrestrial digital video broadcasting.

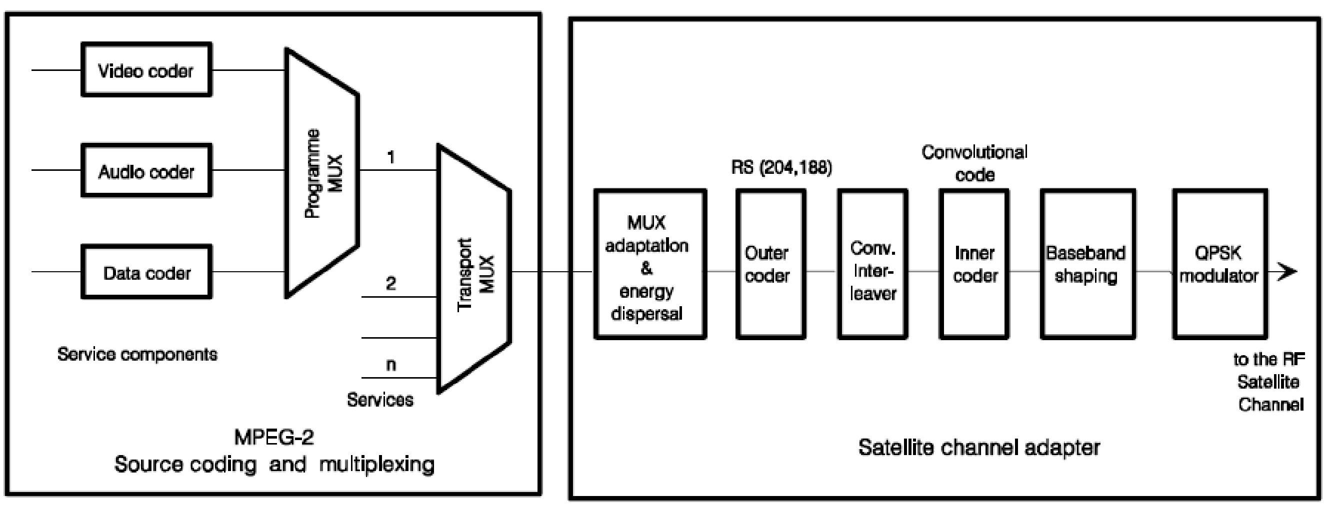

Fig 1.2. DVB-S standard digital satellite broadcasting system architecture [51]

Some countries developed their domestic DVB standards: ISDB – Japan, DMB-T

– Korea, ATSC – the USA, which, however, are similar to European DVB standards, being based on ISO 13818-1 standards.

The architecture is typical for satellite, terrestrial and cable television broadcasting. The Fig. 1.2. demonstrates that DVB system consists of the components resident in digital communication systems on the whole. The key components of the system are the modules of the designated source coding aimed at eliminating information redundancy (audio and video compressors), and an error-correcting block. The system always includes an audio and video signal multiplexer to provide for synchronous playback. DVB-T and DVB-S standards use Reed – Solomon code as an outer code, and convolutional code as an inner code [51].

The developed compression and transport ISO standards have become widely applicable. The flexibility of transport stream standard supported, beside audio and video MPEG-2, the transportation of other formats of data compression. Standardization involved adding supplements to the basic transport stream standard ISO 13818-1, maintaining the compatibility of earlier receiver equipment with new transmitters.

Introduction of a H.264 video compression standard has become a crucial point in DVB development. The standard allows achieving a significantly higher degree of video data compression maintaining the same quality than other analogues. This lead to the current gradual replacement of DVB systems using MPEG-2 video by a more effective H.264 compression standard.

It must be mentioned that H.264 has the video sequence structure that ideologically is similar to the one of MPEG-2. It will be discussed in more detail in p.1.4 of this chapter.

As a cutting edge audio format applied in DVB the AAC (Advanced Audio Coding) format must be recognized [58, 60]. Before AAC standard was introduced, the MPEG-Layer II Audio standard had been mainly used in DVB.

Speaking of the formats and modes of audio and video transport and synchronization, virtually all of the above mentioned DVB standards utilize MPEG-2 Transport Stream (TS, MPEG-2 TS) or its extension [55]. MPEG-2 TS is applied for satellite, terrestrial and cable digital multimedia broadcasting.

Finally, the newest and fastest developing technology are Internet broadcasting and IPTV, generally speaking TV over IP. While as recently as a decade ago as "a TV receiver" "a TV set" was unambiguously understood, nowadays the role more and more frequently belongs to a personal computer, whose source of a digital TV signal comes from IP networks. TV over IP advance has become possible due to a couple of factors: a significant improvement in communication line technologies, and as a consequence, expanded transmission band available to customers [18], and development of highly effective multimedia data compression standards, especially video, such as MPEG-2 and H.264.

Currently Internet broadcasting is implemented by the majority of TV channels around the world. The websites like **www.webtvlist.com** give links to hundreds of channels in different countries of the world. A lot of channels nowadays broadcast in test mode, or limit the program options available for watching on line. For a reliable Internet TV program screening a connection with transmission band of at least 500-1500 kilobit per second is required. Different channels require various transmission bands depending on the broadcasting quality (resolution and bit rate), compression codec effectiveness, and so on. Unfortunately, the problems in service quality when watching IP TV programs are not infrequent, one of the most common being packet losses.

IPTV service market offers to its customers the services making watching TV programs more convenient, such as Video On Demand (VOD), an option of pausing the program, and the like. There exist various technical and conceptual approaches to IPTV system implementation. Quite common is the situation when a DTV system is being developed by a single large company “from A to Z”: starting from data compression and transport protocol (head-end) and ending with software for TV programs reception and playback (front-end). As a rule, these types of technology are closed. The *Real Media* technology can be cited as an example. Nevertheless, presently it may be stated that a general concept of IPTV system design can be traced. In IPTV systems *MPEG-2* transport stream plays another important role, being used to transport and synchronize multimedia data in a large number of professional systems. For MPEG-2 transport stream IP networking UDP protocol is utilized. [46, 63, 64].

It should be noted that, as an alternative to transport stream, in TV over IP systems there was developed a RFC broadcast technology: RTP (Real-time Transport Protocol) / RTCP (Real-time Control Protocol). [63, 65, 66] The technology allows to perform sound and video synchronization by internal facilities without using MPEG-2 transport stream. In addition, RTP protocol supports MPEG-2 transport stream as a payload format.

A summarized architecture of TV over IP system translating server utilizing MPEG-2 transport stream is given on Fig. 1.3.

Among the most recent advances and tendencies in telecommunication technology one has to mention broadcasting to mobile devices.

Microprocessor-based technology development enabled to equip the mobile communication devices with powerful microprocessors - cellular phones, communicators, which, along with the highly effective digital multimedia data compression, made watching TV programs on the mobile phone screens a reality.

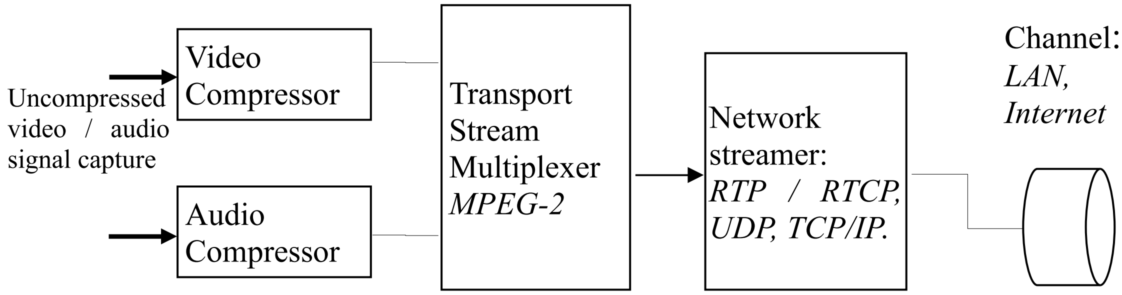

Fig.1.3. A summarized architecture of IPTV broadcasting server utilizing MPEG-2 transport stream

In Europe and Japan several years ago digital TV broadcasting systems for mobile telephones have been put in operation. Czech Television's (CT) multiplex now covers 99% of the population whilst the second and third multiplexes cover 89.5% and 84.4% respectively. The fourth multiplex operated by Telefónica O2 reaches 22% [74].

From August 2010 B PLUS TV company within the framework of EU project "Program of international cooperation" is providing free-to-air TV broadcasting in DVB-H/T standard as first in Czech Republic. Trial operation is launched from 4.August 2010 on 46. TV channel from VSB-TU Ostrava (Technical University of Ostrava). Full operation will be launched from 1. October 2010 on 46. TV channel from VSB-TU Ostrava [75].

For mobile broadcasting AAC and H.264 are also used, beside MPEG-2 transport stream and RTP technology. As an error-correcting code for mobile broadcasting LT code based Raptor codes are applied [36].

1.6 Survey of literature on the subject

The first systematized survey of error-correcting coding theory was offered by W. Peterson in the "Error-Correcting Codes" publication of 1961 [25]. The book offers a complete and profound coverage of theoretical foundation for error-correcting coding.

W. Peterson examined the mathematical principles of designing linear and cyclical codes and their characteristics. For cyclical codes both the method of defining a code as a polynomial over Galois field and the code notation is described.

The scope of elements given from linear algebra and Galois field arithmetic is sufficient for application within error-correcting coding theory. In addition to coding, the book contains the description for a number of linear and cyclical codes decoding algorithms. A decoding algorithm for BCH codes is of major interest. There is offered a systematized classification of error-correcting codes known at the moment of publication, such as Hamming code, Bose-Chaudhuri-Hocquenghem Codes (BCH codes) , Reed-Solomon codes, Fire code and others. The effectiveness of error correcting with the help of various codes is analyzed.

Among more recent publications the book by Richard E. Blahut “Error-correcting codes theory and practice” is of wide popularity [6]. Its original edition was published in 1983. The book also includes theoretical foundations of linear algebra and Galois fields arithmetic. Linear codes and the basic cyclical codes are discussed (Hamming code, BCH, Reed-Solomon). The author emphasizes the practical aspects of code implementation. Decoding algorithms are given a significant attention. The spectral theory of cyclical codes is the main focus of the book. The work presents an attempt to systematize the theory of convolutional codes. The book is a valuable resource for the purpose of error-correcting codes practical implementation.

The book by A. Dholakia may be used as an excellent resource on the theory of convolutional and Turbo-codes [9].

The major sources of up-to-the minute information are publications in journals and scientific periodicals. The paper [8] undoubtedly, the greatest authorities among those are IEEE publications. Costello D. J., J. Hagenauer offer a review of the major achievements in error-correcting coding in the last 25 years. The work considers implementation of error-correcting codes for a wide range of applications, in particular, for digital multimedia transmission.

Concerning erasure codes, in the first place one has to mention the original article by M. Luby [19], the first publication on LT codes. One of the founders of erasure codes theory was MacKay D.J.C. [20] His on-line book on error-correcting codes offering interpretation of erasure codes is available in the Internet.

The results of an attempt to implement erasure codes in high-definition television systems are covered in [2].

As for the sources on digital television systems, audio and video data compression, transport modes, etc., the backbone of the subject are *MPEG*, *RFC*, *DVB* standards [51-71], as they provide the most reliable and up-to-date information.

Useful information on the latest advances in DVB field can be found in Telemultimedia Internet journal [73].

A valuable source of information on compression algorithms, codecs comparison, video and audio quality criteria is the on-line resource of Compressia.ru [72].

Scrupulous attention was also given to certain studies dedicated to the design of adequate packet losses models and IP networks traffic models, and urgent IPTV issues. [28], [29], [30].

1.7 Compression principles and digital multimedia data stream structure

1.7.1 Video compression principles

Digital video in its essence refers to three-dimensional array of pixels.

When digitizing a video signal every pixel is set as three values: brightness magnitude, and two chromaticity values, the so named *YCbCr chart.*

The main of the most efficient concepts for reducing the video data volume are the following (the resources used are [1, 31, 32, 50, 54, 61, 72]):

- video frame color component decimation (“thinning”). A human eye is known to be more sensitive to change of brightness than that of chromaticity. Therefore part of the pixels responsible for chromaticity may be erased without any considerable picture quality loss. As a rule, the formats of 4:2:0 (three fourths of chromaticity pixels erased) or 4:2:2 (a half of chromaticity pixels erased) are applied. Another possibility is using the 4:4:4 format without chromaticity pixels erasure;
- elimination of temporal and spatial redundancy inherent to video. As a rule, two neighbor frames in a video sequence are closely correlated, unless there was a scene change. Therefore, complete information of one key frame may be saved, while for another frame only the difference between it and the key frame be transmitted. Individual image areas within one frame are frequently highly correlated (for example, single-color background). Thus, intraframe elimination of spatial redundancy is performed.

The spatial and temporal redundancy in video is illustrated in Fig. 1.4. by two neighbor video frames.

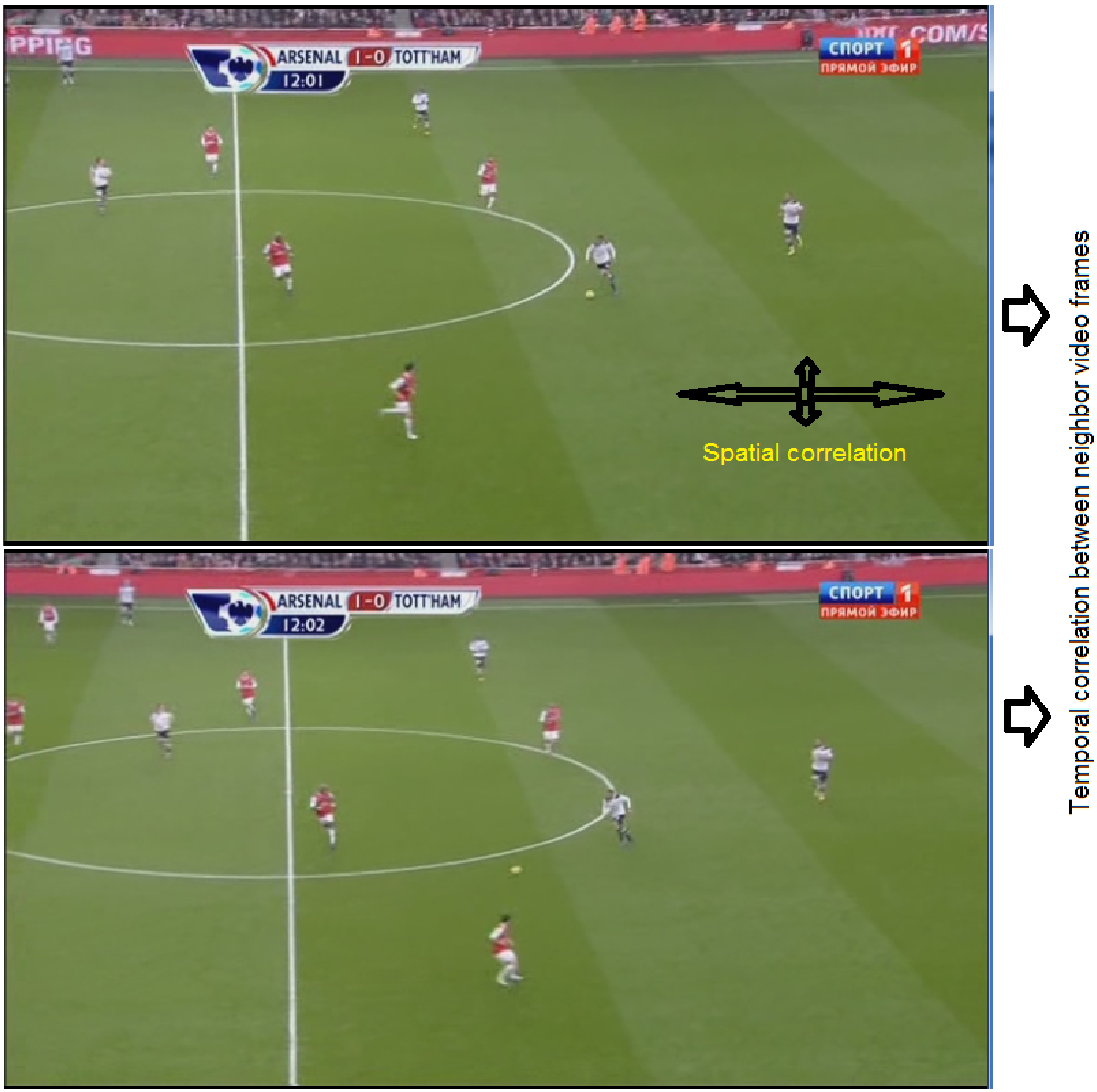

Fig. 1.4. Temporal and spatial correlation in video

To implement the concept of temporal redundancy elimination current compression standards require video to incorporate the sequence of three frame types: intra-coded frames (I-frames), Predictive-coded frames (P-frames), and Bidirectionally-predictive-coded frames (B-frames) (fig.1.5). An I-frame is a compressed version of one initial video frame. It does not depend on other frames.

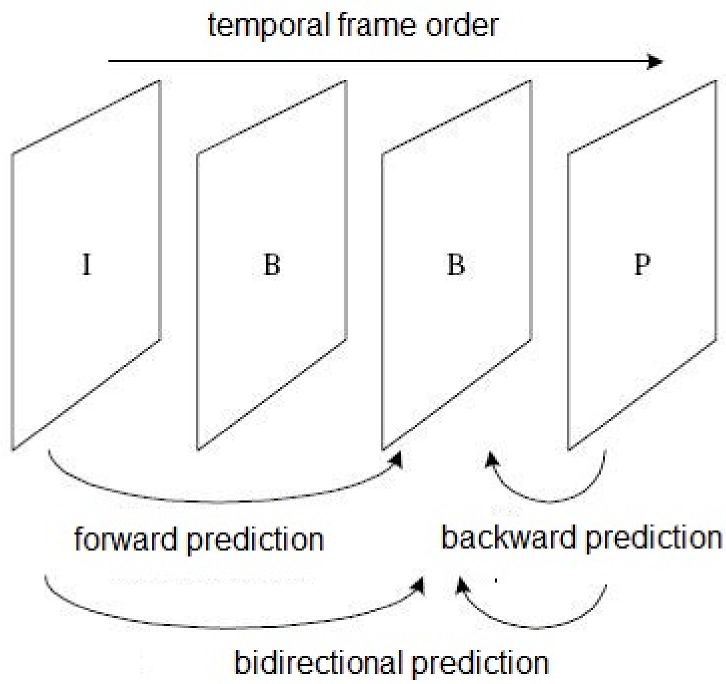

Fig. 1.5. I, B, and P-frames encoding: prediction types

Shortly, compression is performed by a following procedure:

- a frame is split into 8x8 pixel areas, named macroblocks;
- a discrete cosine transform is performed over each macroblock. It results in a 8x8 dimension coefficient matrix. The transform enables to switch from a spatial block presentation into a frequency presentation. At this stage the initial macroblock can be recovered by means of reverse cosine transform;
- the resulting coefficient matrix is quantized by division of the matrix coefficients by a certain number (qauntizer). As a result many of the coefficients will go to zero, which serves to simplify the image and clip a part of the information, the information mainly relating to high brightness and chromaticity frequencies. This corresponds to psycho-visual model of human perception, since the loss of high frequencies affects the overall image perceptual quality to a very little degree;

- quantized matrix coefficients are rolled out into a line by means of zigzag scanning, starting at the top left matrix angle, as in the right bottom area of the matrix most coefficients, as a rule, have zero values;
- the resulting line is encoded by means of Huffman codes, allowing to represent the resulting array in more compact compressed form. Decoding is performed in reverse order.

I-frames are also referred to as *reference* frames, since they serve as a base to encode other P- or B- frames.

To encode a P-frame, the previous reference frame has to be decoded, that is, the frame to be used for encoding the next frame. Current frame is split into 16x16 size macroblocks. An attempt is made to find for each macroblock a partner-macroblock of maximum similarity in a reference frame. To estimate macroblocks' similarity special measures are taken. The process of searching for a similar macroblock is referred to as *motion estimation*. The closest in similarity to the current macroblock may turn out to be in the reference frame with the same coordinates, but in case of motion the coordinates can change. The offset found is encoded as a *motion vector*. Since the closest in similarity macroblock does not have to be identical to the current one, it is necessary to calculate misalignment between the macroblocks, which is to be used for further encoding. If a similar macroblock is not found in a reference frame, then the macroblock is encoded as an intra-frame macroblock. The sum total of all these measures allows to achieve a considerably larger compression degree for *P*-frames.

B-frames encoding is performed in the same way as for P-frames, the only difference being the use of two frames as reference frames for motion estimation: a previous frame and a subsequent one. Interframe correlation when using this encoding method is shown in fig 1.5. The encoding gives the resulting frame sequence in the form of *IBBPBBPBBPBB*...(*I*), named *Group of Picture*, *GOP*.

1.7.2 H.264 video compression standard

H.264 standard [61] offers a range of improvements allowing for considerably higher video compression effectiveness in comparison to previous standards (such as ASP), providing at the same time better flexibility of application in various network media. The main of them are the following:

- Multi-frame prediction. The use of previously compressed frames as reference frames (which means using part of the information from them) allows for much better flexibility than with previous standards. The use of as many as 32 references to other frames is permitted, while with ASP and earlier standards the number of references is limited to one, or in the case of B-frames, two frames. This enhances encoding effectiveness as a coder is able to select reference macro-blocks for motion compensation from a large number of images.
- Entropy coding of quantized transformation coefficients:

  - *Context-adaptive binary arithmetic coding* (*CABAC*) — a lossless compression algorithm for video stream syntactical elements based on occurrence probability, supported only by Main Profile and higher. It provides for compression of larger effectiveness than *CAVLC,* requiring, however, more time for decoding.

  - *Context-adaptive variable-length coding* (*CAVLC*) — a less complex alternative to CABAC. Nevertheless, it is more complex and effective than the algorithms applied in earlier video compression technologies (as a rule, Huffman algorithm [17]).

  - *Golomba coding* - widely applied, simple and highly structured variable-length word encoding of many syntax elements not encoded in CABAC or CAVLC, known as Golomba coding [31].

- Error robustness function:

  - *Network abstraction level (NAL) determination*, allowing to apply the same video syntax for various network environments, including *sequence parameter sets (SPS)*, and *picture parameter sets (PPS),* which provides for increased reliability and flexibility as compared to previous technologies.

  - *Flexible macroblock ordering (FMO)*, also known as ASO – the methods for image macroblocks representation order restructuring. When used effectively, flexible macro-block ordering may considerably increase data loss robustness. Owing to ASO, where each part of image can be decoded independently (with certain encoding limitations), the new standard enables sending and receiving the parts in random order. This may decrease delays in real time applications, especially when used in out-of –order delivery networks. The functions may be also used for a great number of other purposes in addition to error recovery.

- Quantization – a function allowing to separate the data of different significance (for example, motion vectors and other prediction information is of great significance for video content representation) into separate data packets with different error protection levels.
- Redundant parts. An option of a coder to send redundant representation of image areas, which enables to reveal image areas (normally, at the cost of somewhat lower quality) whose data had been lost during transmission (not supported by some profiles).
- Frame numbering, allowing to create subsequences (including temporal scaling by including additional frames between other frames), detection (and recovery) of complete frame losses due to channel failure or packet losses.

1.7.3 Audio compression principles

There are two fundamentally different digital audio signal compression methods: lossy and lossless as to the information of audio stream. In the present study the author is interested in the lossy compression method, for streaming broadcasting, as a rule, applies lossy audio compression standards.

The audio signal compression method is based on psycho-acoustic model of human sound perception [1, 60]. A human ear is capable of hearing far short of all sounds, and it is eliminating those that allows to achieve audio compression of a considerable degree while maintaining acceptable quality. As an example of the sounds one can refer to high frequency sounds, the sounds played back simultaneously with louder sounds. Since eliminating the information redundant in terms of perception may be not sufficient to achieve the desired audio stream bit rate, additional compression methods are also applied, such as MDCT – modified discrete cosine transformation, which allows to shift from temporal domain into frequency domain. Further a part of data with frequencies below a specified for the compression model masking limit is discarded.

The generic audio stream structure is given in the Fig. 1.6.

In terms of television broadcasting objectives, of interest is the question of what a digital audio stream really is. Digital Audio Stream is data sequence of audio frames. For the transport issue a significantly complicating factor is the fact that audio start code is not unique. This applies to MPEG Layer 2, and AAC compression standards as well.

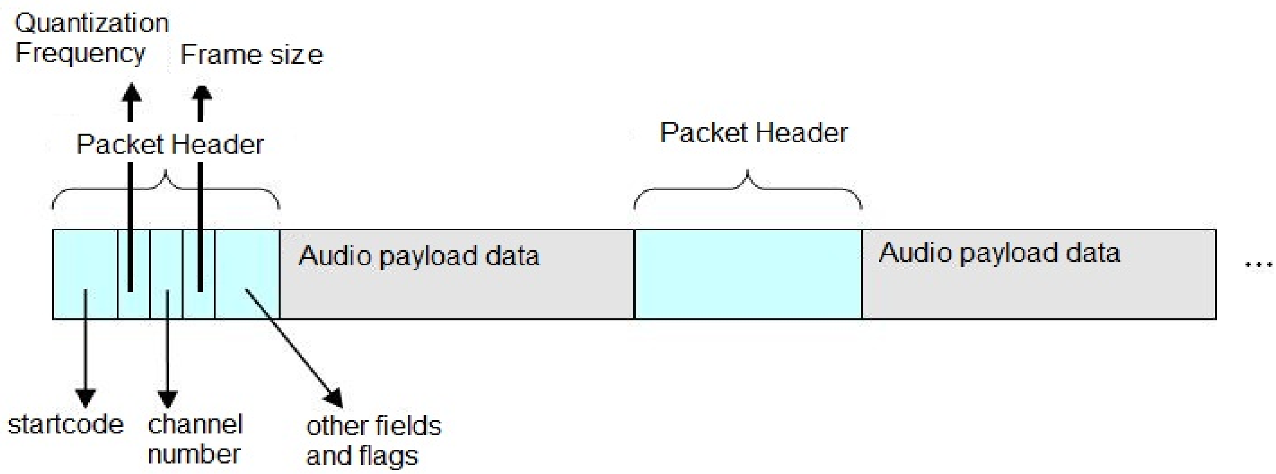

Fig. 1.6. Audio stream structure

Each audio frame header contains most important information needed to decode the audio stream. Having received a frame an audio decoder is able to decode it. The most important audio frame header fields are:

- start code
- audio frame length
- quantization frequency
- audio channels quantity
- audio frame sounding duration

The loss of audio frame header is critical for the frame decoding. Since the start codes in the audio stream are not unique, the loss of a single start code may lead to incorrect decoding of the successive frames during some length of the stream.

Alternatively to video stream, a compressed audio stream does not have any inter frame dependency. However, this does not allow to assume that the "significance" of audio data in multimedia stream is inferior to that of video data, since any sound problem, whatever small the data loss might be, results in considerably reduced perception quality.

1.7.4 MPEG-2 transport stream for digital multimedia data transmission

MPEG-2 transport stream [55] is designed for digital multimedia data streaming broadcasting. MPEG-2 transport stream supports several important functions:

- sound and video image synchronization;
- logical separation of elementary streams into programs within one transport stream;
- providing service information for elementary stream type detection;
- Multimedia data stream error robustness, at the expense of small packet length and others.

MPEG-2 transport stream is an adopted standard for satellite, terrestrial television broadcasting in the vast majority of the countries.

One of the features owing to which MPEG-2 transport stream found its high demand, is its flexibility in supporting various multimedia data formats, as well as ancillary data. Transport stream allows to transmit data in H.264 video, MPEG-2 video, VC-1 video, AAC, MPEG Audio, AC-3 and other formats. Transport stream can transmit television programs, information on a channel, broadcasting language and so on.

MPEG-2 transport stream header syntax is shown in Fig 1.7.

Transport stream consists of a transport packet sequence. The length of each packet is 188 bytes. The number of packets in a transport stream is not limited. Each header is at least four byte long. The first byte is used as a transport stream start code and always has the value of 0x47. They are followed by error flags and others, the most important field after them being *Packet Identifier* (*PID*). It defines a packet tenancy to this or that elementary stream within the transport stream – video, audio, etc. They are followed by 2 bits of scrambling control, adaptation field control bits and a four bit counter. If adaptation field control bits have the value of 01 or 10, then immediately after the transport stream header an adaptation field follows.

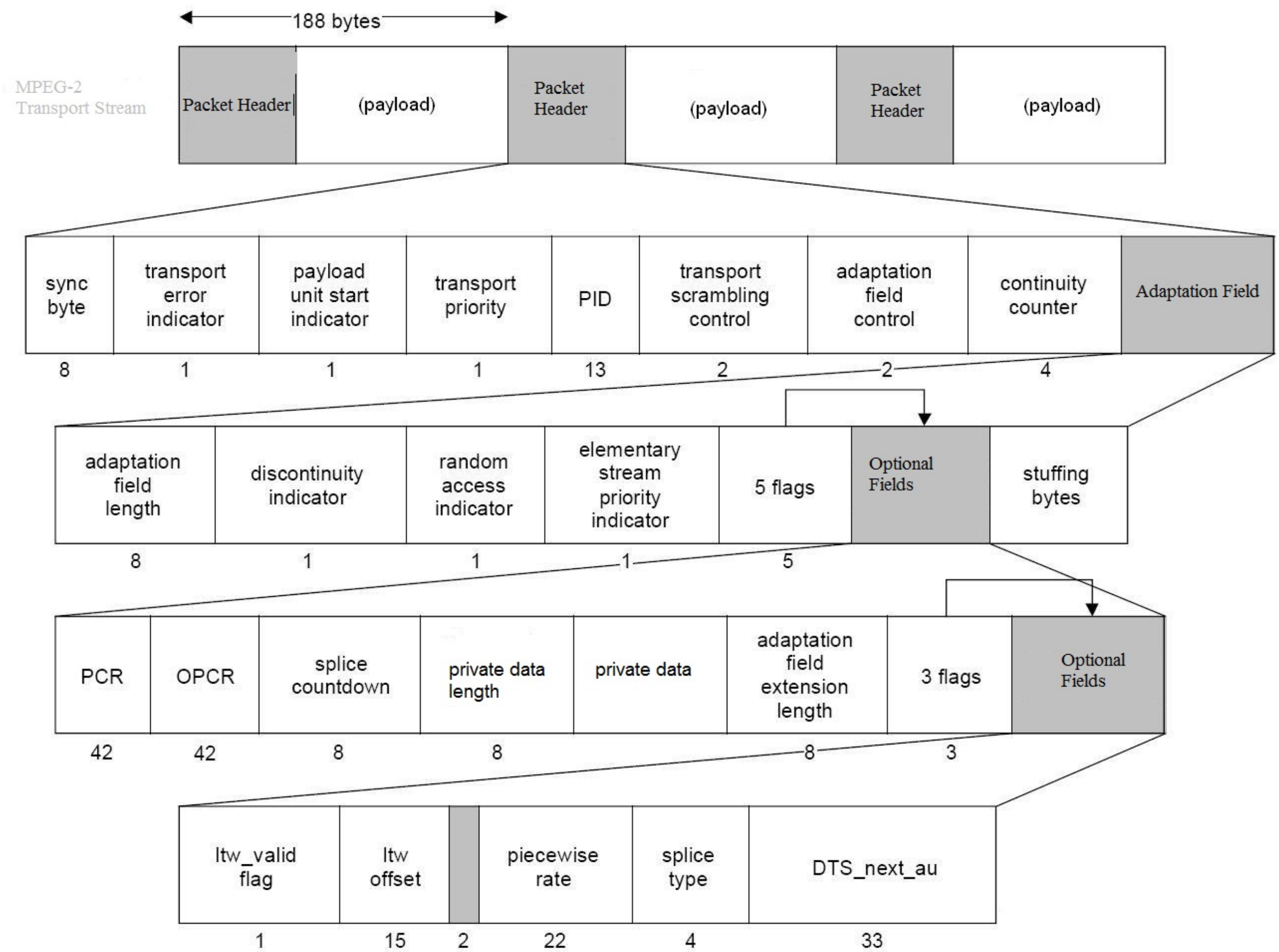

Fig. 1.7. MPEG-2 transport stream header structure

The first byte of the adaptation field indicates a general adaptation field length, starting at the next bit. The byte following the adaptation field length contains the flags based on whose values it is defined if the adaptation field contains any subsequent optional fields. Of much importance is *PCR* (*Program Clock Reference*) field, system time mark of the whole transport stream with the resolution of 27 MHz being written down in it, and a private data field that allows to write down any information in transport packet adaptation field. PCR marks are responsible for transmitter and receiver clock synchronization, and in accordance with the standard are to follow the transport stream at the frequency of at least 100ms. MPEG-2 transport stream allows to transmit several programs in one stream. To logically organize the programs in a stream a mechanism of service tables is supported.

To start transport stream demultiplexing a demultiplexer is to find a packet with *PID* = 0x0 in the stream.

According to the standard the PID is reserved to transport PAT (*Program Association Table)* service table. No any other transport stream data can be transmitted in packets with *PID*=0x0.

PAT table contains references to PID of PMT (*Program Map Table*) packets, and, correspondingly, determines the number of programs in a transport stream. By PAT references demultiplexer finds PMT tables of the transport stream in packets with corresponding PIDs. The PMT tables contain PIDs, whose packets contain audio and video elementary stream data forming the program. It must be noted, that PID intersections in different PMT is permitted, for example, MPEG-2 TS contains only a single video stream, but several audio streams to the video in several different languages. Transmissions in different languages represent logically different programs, a separate PMT table being transmitted for each program in a steam. For each stream in the program, beside PID, PMT table specifies the stream type in accordance with the type chart regulated by ISO 13818-1 [55]. H.264, MPEG-2 Video, AAC audio and others may be one of the types.

MPEG-2 TS provides streaming broadcasting, since the entire information needed for demultiplexing and decoding (PAT, PMT service tables, PCR marks) will repeat in the stream at a specified frequency. By the standard duplication frequency cannot exceed 100 ms. This enables a user to get connected to streaming broadcasting and start watching a desired program after an interval not exceeding 100 ms.

The transport packet length as small as 188 bytes allows to make multimedia stream less sensitive to packet losses. Should a single transport packet get lost, a maximum of 184 bytes of useful information will be lost, which normally does not seriously affect perception quality. When using satellite broadcasting one can frequently observe the image “scattering” into little “squares” for a fraction of a second or a couple of seconds. As a rule, this is the result of separate packet losses and distortions.

1.8 IP protocols and basic principles of TV over IP systems

1.8.1 UDP Protocol

UDP (User Datagram Protocol) is a transport layer protocol in IP protocol stack.

This protocol plays a key role for live digital TV streaming over IP. One of the main advantages of UDP versus TCP is low and constant end-to-end delay (latency). This enables to build consistent buffering model of multimedia data in live streaming mode. The major actual problems of TV streaming over UDP is packet losses due to the fact that UDP does not guarantee packet delivery. This is so-called *unreliability* of UDP protocol. Unreliability should be understand as that in case of packet losses caused by external impacts, the protocol itself does not define any mechanism for packet recovery [70].

UDP is used a transport layer for RTP streaming, when RTP is acting as application layer protocol. Another frequently used scenario is to transfer Transport Stream directly over UDP (TS over UDP, or TS over IP). In that scenario Transport Stream represents application layer protocol for data delivery. An illustration of a principal scheme of DTV system employing TS over UDP transmitting is given in the Fig. 1.8.

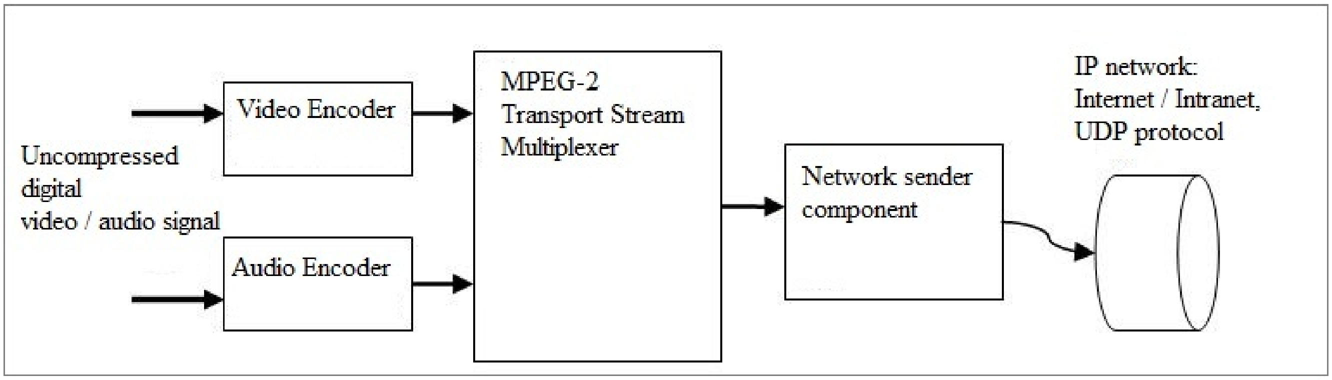

Fig. 1.8. A principal architecture of a DTV system using TS over UDP delivery

The structure of UDP packet header is given in Table 1.1. [70].

Table 1.1. Syntax of UDP packer header

+	Bits 0 - 15	Bits 16 - 31
0	Source port	Destination port
32	Packet length	Control sum
64 …	Payload Data	

The UDP packet header contains 4 fields. Source and destination ports define which port is used for communication between sender and receiver. Control sum allows to check the packet's consistency. In case of control sum mismatch the packet is discarded.

Nowadays UDP is a core protocol for IPTV and TV over IP systems.

1.8.2 Unicast, Multicast and Broadcast streaming modes

For the purpose of IP network broadcasting the following modes of information transmission are utilized: *unicast*, *broadcast* and *multicast*.

Unicast is applied to render customized services. The method allows to transmit information from the source to a particular IP address. A subscriber orders personalized content intended exclusively for him or her, and gets the service ordered personally.

When several users are viewing their orders simultaneously, their traffic is summed up at the area from the source where the needed programs are located to the subscriber line. This mode has great importance and frequently utilized in point-to-point delivery, news gathering system, modern content propagation systems among different cities etc.

Broadcast mode is applied to broadcast from one source to all recipients in a specified subnetwork. The information is delivered to every subscriber without exception. For the mode the addresses ending in 255 are utilized, for example, 192.168.1.255. If a television program is broadcasted in *broadcast* mode, all users in one subnetwork will have to view only this channel. Thus, the mode is utilized to transmit only service messages.

Multicast is the most important broadcast mode in IPTV. It is intended for data delivery to a group of subscribers, and applied when arranging television translation and other mass media services. For channel groups identification a specially reserved for the purpose at protocols development address range is used - from 224.0.0.0 to 239.255.255.255 (D class). Multicast mode allows for multimedia streams transmission from the source to the subscriber commutator in one stream, transmitting it further only to the ports that have ordered the information. Multicast serves to considerably save on transmission band in the network, as it does not require to use a separate stream of every channel to every viewer.

1.8.3 RTP / RTCP Protocols

An alternative to Transport Stream over IP streaming is streaming over Real-Time Protocol (RTP). RTP is an application-level protocol used for multimedia data delivery in live mode. The standard is defined by *RFC 3550* document [63]. Initially, RTP has been designed for multicast streaming, however it can be used for unicast as well.

RTP has no specific UDP or TCP port assigned for connection. As a rule, RTP is configured to work by the ports in a range 16384-32767. The only requirement by the standard is that for each even port being used for data delivery the next odd port is dedicated for Real-Time Control Protocol (*RTCP*). A general architecture of a TV system on a basis of RTP protocol is given at Fig, 1.9.

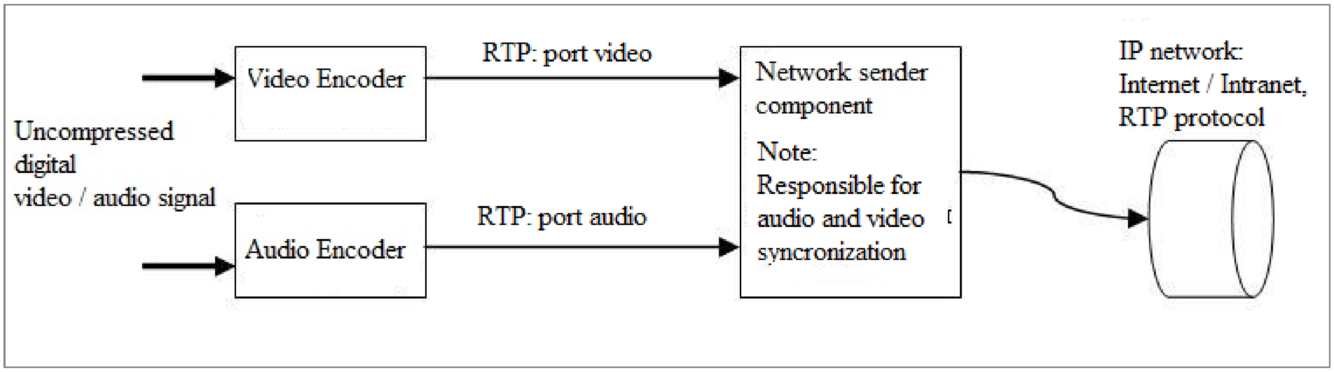

Fig. 1.9. General architecture of TV system on a basis of RTP protocol

RTCP is responsible for service information, session opening and closing, defines media stream types, checks incoming packets' consistency, provides content resource description, monitors available network bandwidth status.

It should be mentioned that RTP could be used for transmitting of Transport Stream packets as well as elementary audio / video streams [65].

The fields *Sequence Number* in the RTP packet header allows to detect lost of a packet [84]. The *Timestamp field* is used for synchronization of video and audio streams when streaming elementary streams over UDP.

RTP and RTCP protocols are application layer protocols on the top of UDP transport protocol.

1.9 Conclusions for Chapter 1

Nowadays, highly efficient audio and video compression algorithms are designed. Most effective compression standards are AAC for audio and H.264 for video.

The high efficiency of video compression is achieved thanks to temporal prediction between frames. Removing of redundancy leads to establishing of internal dependencies between neighbor frames. This makes compressed multimedia streams very sensitive to data losses and distortions during transmitting, since losses of even relatively small pieces of data might cause avalanche effect and lead to incorrect decoding and playback for a significantly long time interval, which affects video quality perception.

Compressed audio bitstream has no temporal dependencies, but audio data losses gets critical for overall perception of TV programs, because human psychological audio-visual model of perception is more sensitive to troubles with sound than video distortions.

Data losses during transmitting networks is an actual scientific and technical issue for present Digital TV systems. In most of the cases, particularly for live broadcasting TV, the only one type of error-correcting codes is suitable – Forward-Error Correcting codes, as there are no opportunity for data retransmitting because of channel nature and architectural specificity.

There are reliable FEC schemes designed for satellite, terrestrial television systems. These systems employ multi-stage concatenated coding by block and convolutional codes, e.g. concatenated coding by Reed-Solomon and Turbo codes.

More and more popular gets Internet TV, IPTV. It is a general trend for TV to be distributed over IP networks. It is caused by several reasons – convenience for content owners and ease of access for TV auditory are the major reasons of that trend.

For broadcast (in sense of live) television IP networks primarily employ UDP protocol as a transport layer, to guarantee constant latency of signal delivery to all receivers. UDP protocol is unreliable in sense of packet delivery, because it does not provide any mechanism. This is an open actual problem, as the traditional FEC codes are confirmed to be unsuitable for protection of data in packet-based IP networks.

A promising approach has been invented in the beginning of 20^{th} century by M. Luby. The class of novel codes he suggested is called Erasure Codes. The second chapter is aimed at survew of existing erasure codes and critical review of their applicability to real-time TV broadcasting.

2 Erasure codes and their applications for TV over IP systems

2.1 The concept of erasure. Erasure channel models.

2.1.1 Binary erasure channel model

According to Peterson [25] “erasure refers to an error whose position is known, and its value may be (or may be not) equal to zero”. This definition can be reformulated as follows: “Erasure is an error, whose position is known, but its value is not defined”.

On the whole, packet losses taking place in IP networks fall under the definition, since IP networks utilize end-to-end packet numbering (refer to p.1.7), and there exists an opportunity to locate the lost packet. The value of erased packet is not known to a receiver (decoder), while the position may be known, but may be not.

Let us consider an idealized model of binary erasure channel (Fig. 2.1)

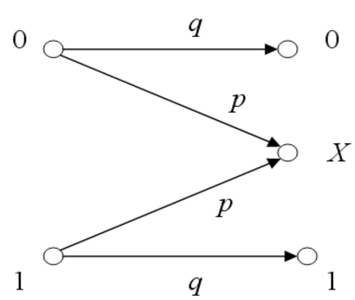

Fig. 2.1 Binary erasure channel model

For the channel a probability q of a successful packet delivery is specified (the value of the packet received is identical to that of the initial packet), and the probability $p = 1 - q$ that the packet forwarded will be erased. The erased packet value is designated as X. The packet value is assumed to have no influence upon its erasure probability. Generalizations of erasure channel include non-binary erasure channel and a channel with erasures and errors. This erasure channel model represents an idealized data transmission system, where demodulator issues a symbol corresponding to erasure and something other than (different from) 0 or 1.

However, this model is an idealized one, and is only applicable for conceptual erasure channel description. The models based on Markov's processes are more realistic.

2.1.2 Gilbert Model

Packet erasures in transmission networks may be sequential, which requires a model allowing to imitate sequential packet losses. One of the most common erasure channel models serving to describe sequential erasures is Gilbert model (Fig.2.2) [35, 45]. Gilbert model represents a Markov's process with two states.

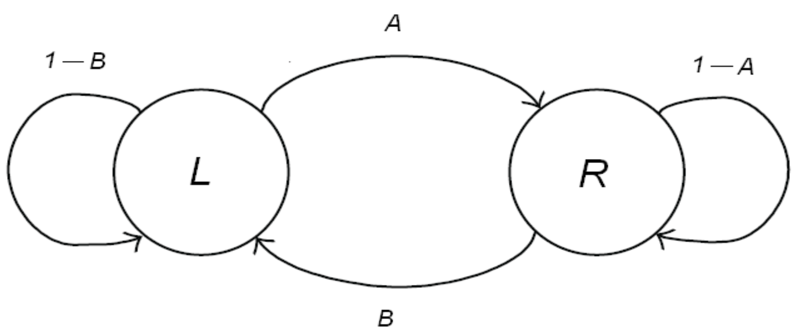

Fig. 2.2. Gilbert Model serving to describe an erasure channel

L (Loss) state corresponds to packet erasure, and *R (Receiving)* – to its delivery. The probability of transition from one state into another is defined by *A* and *B* values. *A* value defines the packet delivery probability if the previous packet was lost, and *1- A* defines the probability of delivering two packets in succession. Correspondingly, *B* is the packet erasure probability when the previous packet was delivered, 1 – *B* is the probability of two packet losses in succession.

Packet loss correlation for this model is represented by

$$PLR = \frac{A}{A+B}. \tag{2.1}$$

Nevertheless, a successive erasure model may be simplified for the cases, when packet interleaving is applied for message transfer. Naturally, interleaving does not affect the channel operation in any way, while allowing to prevent successive packet erasures in source messages.

Thus, at a sufficient interleaving rate, it becomes possible to transfer from Gilbert model to a model with independent packet erasures that happen randomly at some equal probability, which simplifies the analysis of erasure probability and erasure recovery for erasure channels and error- correcting codes designed to fight erasures.

As mentioned above, the packet interleaving method permits to consider the packet transfer process as a process of repeated independent trials with chance outcomes. A trial represents one packet transferred by a transmitter to a receiver. Each trial has two possible outcomes: event *A* (packet erasure) and event *B* (successful packet delivery). The events *A* and *B* are incompatible. Let us designate, as above, a *B* event probability as q, and an *A* event probability as p .

According to probability multiplication theorem [12], the probability of successive packet erasure *k* is equal to

$$P_k = p^k \cdot q . \tag{2.2}$$

The probability of *k* out of *n* packets erasure according to Bayes' formula amounts to [8]

$$P_{k,n} = C_n^k p^k q^{n-k} = \frac{n!}{k!(n-k)!} \cdot p^k q^{n-k} . \tag{2.3}$$

The erasure probability not exceeding *k* out of *n* packets amounts to

$$R_{k,n} = \sum_{i=0}^{k} P_{k,n} . \tag{2.4}$$

2.2 Luby Transform erasure codes (LT). Coding and decoding algorithm

Let us proceed to considering a new class of error-correcting codes for erasure channel – erasure codes. The major characteristic of erasure codes is the possibility to recover the packets erased during transfer process. Erasure codes are essentially different from the above-mentioned classical block codes and convolutional codes, since they allow encoding of a source message of a finite length as a potentially unlimited stream of coding symbols. When needed, an erasure encoder keeps generating coding symbols until the source message will have been recovered by the receiver.

For this reason, in English literature erasure codes are named “*rateless”*, which can be interpreted as “of non-fixed transmission rate”, while the classical block or convolutional codes refer to a class of codes with a fixed transmission rate. Erasure codes may possess a *fountain* quality. The fountain quality is that with erasure code the *order* of coding symbols transmission is irrelevant, and, correspondingly, the order of receiving them within one block of coding symbols is not essential.

The issues of feedback availability, and the applicability of these properties in IPTV systems will be further discussed below, this chapter being dedicated to the base of erasure codes theory. The main focus is on the LT codes, as they represent the most theoretically orderly erasure codes.

Let us consider LT encoding process. The code was invented by M. Luby in 1998, but the first publication about it appeared in 2002. [19]. The key to understanding LT code is the description of encoding process. Here below author cites [19] in order to provide basic background on Luby encoding and decoding procedures, as it is a principal basis of erasure coding.

The encoding process:

- “Randomly choose the degree *d* of the encoding symbol from a degree distribution.

- Choose uniformly at random *d* distinct input symbols as neighbors of the encoding symbol.
- The value of the encoding symbol is the exclusive-or of the *d* neighbors.”.

Luby Transform code can be graphically represented by a generating graph. Generating graph defines which original symbols are taken as neighbors for each particular coded symbol. An example of a generating graph is given in Fig. 2.3.

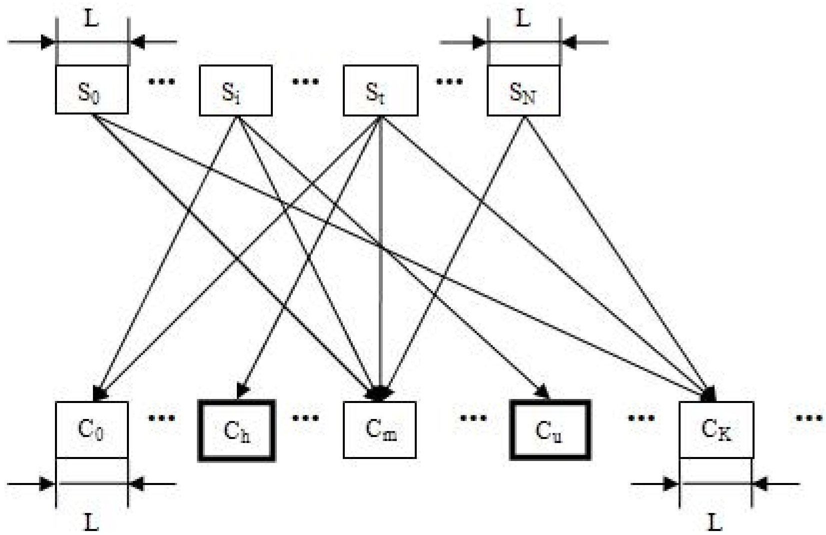

Fig. 2.3. An example of a generating graph for LT code

The decoding process is performed in the following way (citing [19]):

“All input symbols are initially uncovered. At the first step all encoding symbols with one neighbor are released to cover their unique neighbor. The set of covered input symbols that have not yet been processed is called the ripple, and thus at this point all covered input symbols are in the ripple. At each subsequent step one input symbol in the ripple is processed: it is removed as a neighbor from all encoding symbols which have it as a neighbor and all such encoding symbols that subsequently have exactly one remaining neighbor are released to cover their remaining neighbor. Some of these neighbors may have been previously uncovered, causing the ripple to grow, while others of these neighbors may have already been in the ripple, causing no growth in the ripple. The process ends when the ripple is empty at the end of some step. The process fails if there is at least one uncovered input symbol at the end. The process succeeds if all input symbols are covered by the end."

One of the most important issues for LT codes is the design of degree distribution. The issues of degree distribution development are considered in [19, 21, 22, 38].

2.3 Probability estimation of packet recovery with LT codes

One of tasks performed by author during the theoretical research and analysis of applicability of erasure codes to real-time DTV systems was to analyze probability of initial packet recovery when using LT codes.

When analyzing recovery probability with LT codes, one should take into account that it is a sequence of codes that is transmitted through a communication channel, not the original packets themselves. Therefore, the correlations (2.1) – (2.4) are applied for coding packets, and then, based on the coding packets decoding probability, the probability of original packets recovery is defined.

Since LT coding algorithm offers the fountain characteristic of a code chain, it is reasonable to consider a code chain as a packet sequence over which interleaving has been performed. This allows to use the erasure channel model with independent accidental erasures (p.2.1).

Let us assume that a coding packet erasure probability during transmission is equal to P. Correspondingly, successful transfer probability is $q = 1 - p$. The probability of successful original packet recovery will be a function of the number and level of coding packets that have the original packet as a neighbor.

Let us assume that an original packet s_f is a generating packet of a level 1 coding packet. Then the probability of successful recovery of s_f is

$$Q(s_f) = q . \qquad (2.11)$$

This formula is correct only if an original packet is not not involved in any coding packets as a neighbor. Otherwise, (2.11) is transformed into a more complex relation (2.12) Since with LT encoding a packet recovery may be possible to be performed by several methods, it is necessary to take into consideration the whole range of options available.

Let us denote as A_t an event of original packet s_f recovery by using some coding packet c_t. Recovery of s_f with the help of c_t is also stochastic, being defined by two conditions:

a) coding packet c_t was received by a decoder and b) all original packets s_i involved with coding packet c_t as neighbors, except s_f, can be recovered by means of packets c_j, $j \in 0 \ldots N, j \neq t$.

Thus, due to the above, recovery probability of s_f by means of one of the coding packets s_f c_t by the probability product formula is

$$Q(A_t) = \left[\prod_i^D Q''(s_i) \right] \cdot q ,$$

Where $Q''(s_i)$ - total recovery probability of original packet s_i;

s_i - all neighbors in coding packet c_t, except s_f;

D – level c_t.

The events A_t are collateral, since original packet recoveries by means of different coding packets are not mutually exclusive.

Thus, a packet s_f total recovery is calculated by summation formula of collateral events probability:

$$Q''(s_f) = \sum_i Q(A_i) - \sum_{i,j} Q(A_i A_j) + \sum_{i,j,k} Q(A_i A_j A_k) - \ldots + (-1)^{n-1} Q(A_1 A_2 \ldots A_n) \quad (2.13)$$

By substituting (2.12) into (2.13) it is possible to calculate the total recovery probability for original packet s_f.

Let us consider a particular case, when for s_f there is a single occurrence of involvement with level 2 coding packet together with packet s_r as a neighbor. The s_f packet recovery probability according to (2.13) will be

$$Q''(s_k) = q \cdot Q''(s_r), \quad (2.14),$$

which means that it is equal to the probability of coding packet delivery (probability q), AND the neighbor s_r will be successfully decoded, but without using the packet s_k (otherwise, we are going to get infinite recursion).

Let us assume that s_r has a singular decoding variant. Let us consider two cases. In the first case s_r is involved in the level 1 packet as a neighbor, and $Q''(s_r) = q$. In the second hypothesized case let us assume that s_r may be recovered by a certain level 2 coding packet s_r decoding, in which the second neighbor has total recovery probability q. Then probability $Q''(s_r) = q^2$.

Then for the first case $Q''(s_k) = q^2$, and for the second case $Q''(s_k) = q^3$. Since $q<1$, in the second case the recovery probability will be smaller.

This example serves to illustrate the fact, that the recovery probability of initial packet encoded within a packet whose level exceeds 1, essentially depends on which packet was taken as a neighbor to encode a given code character.

This is also confirmed by the formula (2.13) for total initial packet recovery.

The result shows, that using initial packets with high recovery probability as neighbors to packets with smaller recovery probability is more effective, than that of weakly protected initial packets as neighbors to high level coding packets.

For LT codes utilizing randomly selected neighbors it means that packet recovery probability may be considerably (and uncontrollably) variable. Some packets, in the case of “unsuccessful” selection, may turn out unrecoverable even at the total absence of packet loss. For larger K values due to a great number of a packet entries into coding characters, recovery probability gets averaged and reaches a certain value that allows a considerably high probability of recovery. However, with smaller K values, in the range of 100 - 500 packets, the phenomenon may lead to a considerably degraded LT codes effectiveness.

From the mentioned above, the developed robust distribution is based on certain probability assumptions and statistical relations which comply for "sufficiently large numbers." In addition, the encryption algorithm inherently is a random selection of neighbors and a random selection of the degree of code symbols. It also makes necessary to use a certain large value of K for statistical regularities.

2.4 The Analysis of LT codes applicability in TV over IP systems

2.4.1 Formulating of requirements to FEC for TV over IP systems

Considering the perspectives of LT codes applicability for multimedia streaming, the most important, in the author's view, are the following three aspects:

- what marginal threshold value of initial K characters number LT code will be able to provide for effective loss recovery? It would be desirable to achieve effective recovery with the least K number, since the larger K number, the longer the total time of data delay in TV system (“*end-to-end delay”);*
- *PLR* packets loss ratio critical for successful decoding at a certain threshold K value. The term of “successful decoding” needs a more precise definition. It is common to consider total decoding of all initial packets as successful decoding.

However, to achieve this LT codes offer a possibility of reforwarding coding packets until all the initial packets have been recovered. For real time streaming broadcasting this option is not possible. Each video frame has to be displayed at a precisely defined time, and an arrival of an additional packet for the frame data recovery cannot be waited for. Thus, we are going to assume as successful decoding in this context the recovery of some number of packets allowing to achieve an acceptable visual perception quality of TV programs;

- the third critical aspect – at threshold K value, to what extent will the N/K ration increase, i.e., how large will the overhead contributed by LT encoder be. This is the most important characteristic, as an additional bandwidth loading, keeping in mind that it is already operating at its limits due to packet losses, may lead to increased *PLR* and degraded transmission and receiving.

To be able to answer these questions, an experimental *Erasure TestBench* program was developed for LT simulation.

2.4.2 Simulation experimental results for LT codes

The *Erasure TestBench* program is designed to simulate erasure codes error-correcting coding. The program allows to estimate erasure codes effectiveness at various values of code character level distribution. As a criterion for erasure codes effectiveness estimation the percentage of recovered packets to the total number of initial packets is applied (ref. p.1.9). To estimate the codes effectiveness the program simulates a computer network where packet losses occur at a specified loss rate. Initial information sequence composed of K packets is encodes by means of LT codes. In addition, there were implemented systematic erasure codes, which will be discussed further. At the erasure coder output there is N coding packets ($N > K$). The program simulates coding packets transmission in a computer network with unwarranted packet delivery, emulating the loss of some number of packets. The number of packets to be lost is determined by an input PLR program value (*Packet Loss Ratio)* as $PLR \cdot N/100$.

The packets subject to loss are randomly selected from the transferred packet sequence as values of random quantity with uniform probability distribution. Since packet losses occur randomly, the result of the experiment may vary from run to run. In this connection, statistical simulation is implemented in the program – the results of the experiment are averaged for encoded *R* runs, loss simulation and decoding for each of the codes implemented.

The following parameters are the program inputs:

- the number of packets in initial information sequence, *K*;

- the required number of coding packets *N (N > K),* generated by erasure coder;

- the values *c* and δ - Robust distribution density of code characters levels *p(d)* for LT codes.

The robust distribution density is set according to (2.8).

The value of c is set as some positive number; it is recommended to set $0<c<1$; δ – value, determining the probability of successful decoding; the decoding process will be successfully completed at the probability of $1–\delta$.

Based on these considerations, it is recommended to set δ as small values in the range of 0 – 0,2. Otherwise, the decoder may fail to decode the coded message, and the recovered packets ratio may turn out to be extremely low, up to zero;

- *PLR* – ratio of the packets subject to loss during transmission (packet loss ratio);

- the number of simulation runs for statistical averaging of the results *R*;

The LT coding and decoding algorithms were implemented in strict conformity with the description given above in this chapter. Robust distribution was applied for the implementation. The program was implemented in C++ programming language.

The following data structure was utilized as a packet:

```
struct packet_s {
        int     number;
        int     d;
        int     neighbors[MAX_REFS];
        int     payload;
};
```

where *number* is packet serial number,

d - packet level,

the *neighbors* - serial numbers array (references to initial packets numbers),

MAX_REFS - a constant for the maximum possible coding packet level. In the implementation the value of 100 was applied.

payload – “useful” information in the packet; It is the field over which the XOR operation is performed.

Further the description of robust distribution program implementation is given. The distribution $\mu(d)$ is specified as tabular data according to (2.8), applying the values of c, δ given above, for each K value selected for the simulation.

The degree d is incremented by 1 starting at $d = 1$, until the probability sum total reaches $\sum_{i=1}^{d} \mu(i) < 1$.

The value of d at which the total probability reaches 1 determines the maximum degree d_{max}.

According to the LT coding algorithm determination of coding packet degree is performed randomly. For this purpose the program applies a standard C++ language *rand* function which returns random values from 0 to 1 .

Using the distribution table as a table of correspondence between a packet degree and selection probability is not entirely convenient, since then one has to iterate the table in search for the element's index in the table, whose probability is less or equal to the dropped random value.

That would lead to additional computational expenses. Thus, the following method was implemented: instead of probability table an array I of P elements was created, where the elements are arranged as follows:

$$I_j = 1, \forall I_j : j = 0,...,ceil(\mu(1) \cdot P),$$
$$....$$
$$I_j = i+1, \forall I_j : j = ceil(\mu(i) \cdot P),..,ceil(\mu(i+1) \cdot P)$$
$$I_j = d_{\max}, \forall I_j : j = ceil(\mu(d_{\max}-1) \cdot P),...,ceil(\mu(d_{\max}) \cdot P)$$
,

At this point and further *ceil(x)* represents an operation of rounding-off fractional value *x* to upper integer value.

For example, $\mu(1)$=0.1, $\mu(1)$=0.378 P=1000. Then array *I* will comprise $P \cdot \mu(1)$ =100 of the first elements having the value of 1,and following them $P \cdot \mu(2)$=378 elements having the value of 2. Now to select the coding packet degree is enough to perform the following operation (C++):

d=I[rand()(P-1)]/RAND_MAX;*

that is, a simple index referencing, the index being determined pseudo-randomly.

For robust distribution values K = 10000, c = 0.2, δ = 0.05, there were calculated the values S = 244, K / S = 41, $Z \approx 1.3$. These values agree with the calculation results [21], which confirms the validity of the model. The illustration of the robust distribution being built against varied values of c is illustrated in the Fig. 2.4. The graphic is created by author using Microsoft Excel by values obtained using *Erasure TestBench* program.

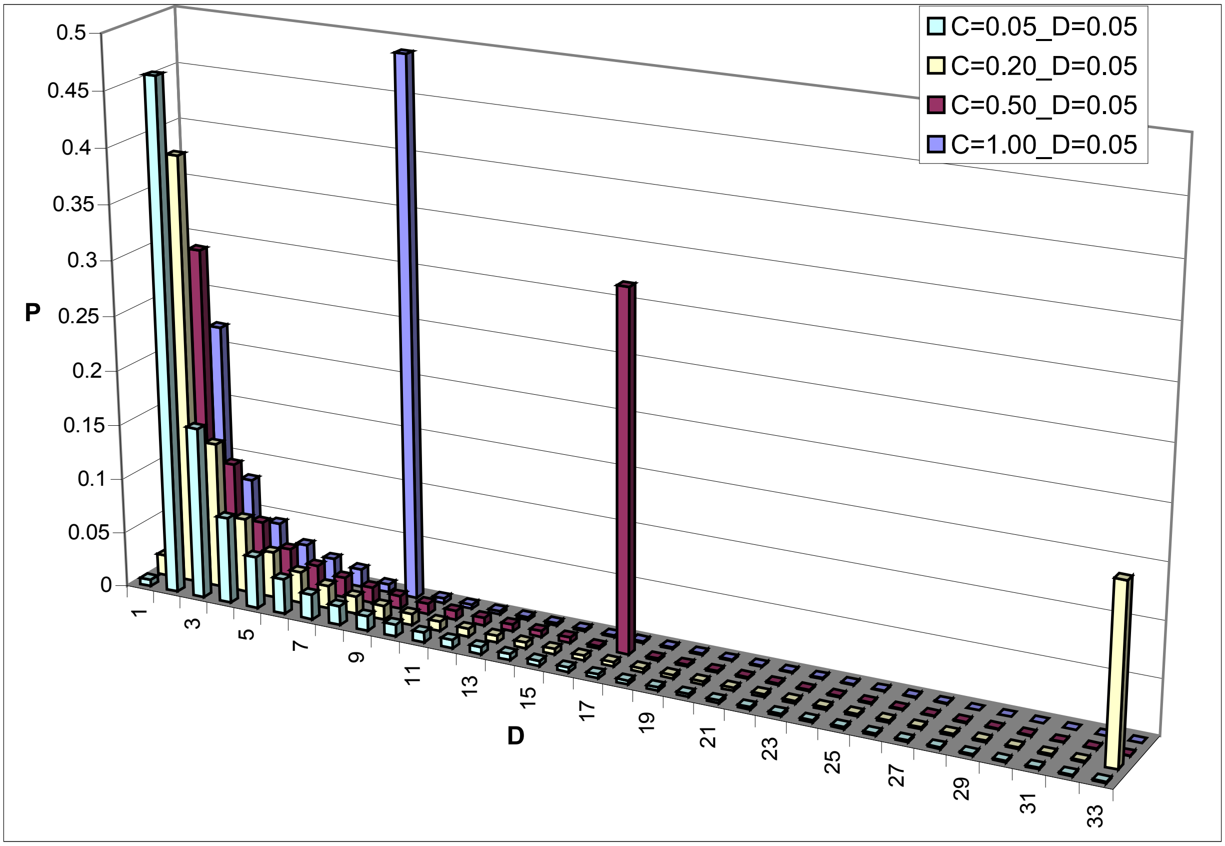

Fig. 2.4. Robust distribution of coded symbol's degrees (*D*) probability

The illustrated diagram is built using parameters *c* and *δ* both equal to *0.05*. The *D* axis: degree of a coded symbol, the *P* axis: probability of a coded symbol to get degree *D*. The 4 colored curves are corresponding to 4 values of the parameter *c*. The parameter *K*=1000.

The results of the program operation are:

- the ratio of packets successfully decoded by receiver, averaged over R runs of simulation for each erasure code simulated;
- mean square deviation σ from the average ratio of packets successfully decoded by receiver.

The program outputs the results into the text files, which could be viewed using any text editor program (e.g. Notepad.exe), or opened by table processors like Microsoft

Excel for further processing and building of graphs.

The program allows to vary parameters K, c and δ. The parameters were varied within the intervals:

$K = \{100, 500, 1000, 5000, 10000\}$,

$c = \{0.05, 0.1, 0.2, 0.5\}$,

$\delta = \{0.05, 0.1, 0.2, 0.5\}$.

Parameter N was also varied, taking values $N=\{1.05 \cdot K,\ 1.1 \cdot K,\ 1.2 \cdot K,\ 1.4 \cdot K\}$, for each K.

For each combination of the parameters was performed the experiment, which is done due to the following procedure:

–the robust distribution is built using the certain set of parameters;

– LT coding is performed using the appropriate robust distribution;

– packet loss emulation is done; *PLR* was taking values {0, 3, 5, 8, 15} for each combination of the mentioned above control parameters;

– decoding is performed;

– decoded packet values are compared against original array, and number of successful / lost packets is evaluated.

Every experiment has been performed 1000 times for every set of parameters. Value 1000 has been considered high enough to ensure statistically correct experiment.

The results of the experiment are provided in the Appendix A, graphical representation of table data is given in Appendix B.

The best result in sense of recovery effectiveness is achieved for c=0.05, δ=0.05. For other combinations of the parameters the result is significantly worse, therefore here below we will analyze only the results obtained for the most successful parameter set.

As can be seen from the diagrams, the recovery gets reliable starting from K= 500, and at overhead equal to 40% ($N=1.4 \cdot K$). Practical applications require to get as low overhead as possible in order to use the expensive channel resources as effectively as possible. By the results of the experiment, the overhead value = 20% achieves reliable recovery at

K=5000. Under these conditions the code resists up to 8% of packet losses. If packet loss ratio further increases, an abrupt splash of effectiveness is observed.

To conclude, the discovered characteristic of LT codes matching the anticipated theoretical assumptions mentioned above. The study shows, that LT codes are suitable for data storage applications, but do not fulfill the requirements to FEC in live TV broadcasting systems over packet-loss networks.

2.5 Code Of Practice (COP) #3 codes

A FEC approach which has to be mentioned within the scope of digital TV transmitting is COP# 3 code, developed by Pro-MPEG group. The COP# 3 code are designed particularly for multimedia data transmitting over IP.

The encoding is performed in the following way. Original information sequence consists of packets having equal length. Typically, RTP packets are meant. The packets are organized into the matrix of D x L. COP# 3 coding adds $D + L$ (or less) redundant, or coded packets to the sequence, where D packets are generated as XOR over rows, L packets are generated as XOR over columns. The overhead is determined as

$$(L+D) / (L\text{x}D).$$

The illustration of the coding process is given in Fig. 2.5.

The coded packets are transmitted by dedicated RTP ports. The values of L and D should be within the interval of {4..20}, and product of L x D should not exceed 100.

The effectiveness estimation and latency consideration could be found in [76, 77].

Benefit of COP# 3 code is that they have systematic nature (original packets transmitted as they are), and relatively simple in implementation. Next positive point for

COP#3 is significantly lower latency comparing to e.g. LT codes, determined recovery possibilities.

Still, COP# 3 code do not provide opportunities for ranking of input packets by priority, and need quite large overhead to effectively protect the data.

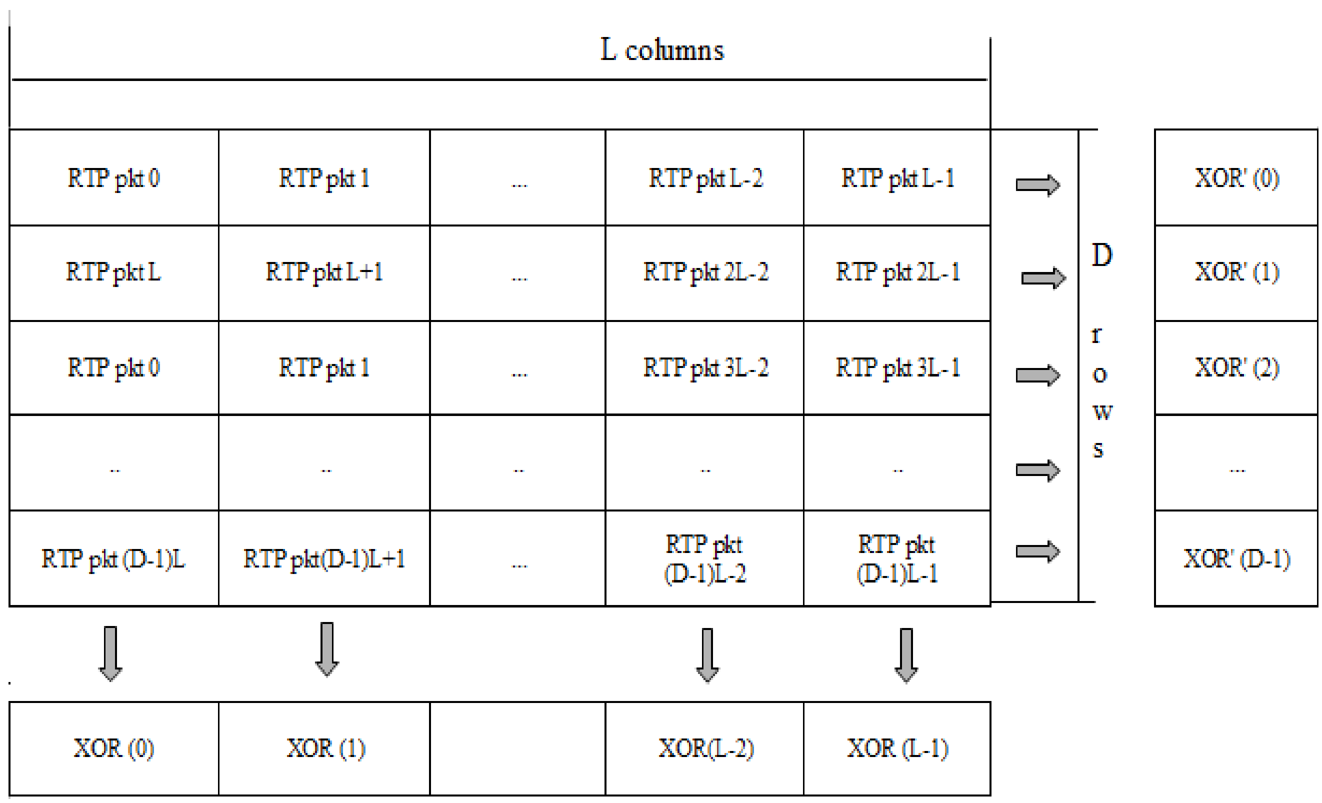

Fig. 2.5. COP#3 coding scheme

2.6 Systematic Erasure Codes (SEC)

Let us consider the specific class of erasure codes: so-called "systematic erasure codes" (SEC). The term 'systematic' has the same meaning for erasure codes as for blocking codes. Due to Peterson [25], systematic code is such a code, for which first k components of the coding vector can be randomly selected information symbols, and any of $(n - k)$ components is a linear combination of the first k components. Initial k components are called information symbols, rest of $(n - k)$ symbols are called redundant or checking symbols.

This means, that original symbols are not affected by coding process.

SEC codes belong to the class of forward error-correcting codes.

SEC coding process is performed by the following procedure. In the same way as for LT codes, the original message is separated onto K original packets each of equal length L. In addition to original K packets V redundant packets are added. The V / K ratio is called *overhead* of coded sequence. Original K packets are transferred as they are without modification, each of V redundant packets is generated as a result of XOR operation over D original packets, where D is an integer value from the interval $0 < D < K$.

To construct SEC, the same question as for LT codes arises – how to choose degrees of redundant packets. Analysis of available resources, incl. Internet resources, didn't give an answer on this question, therefor author performed own experiment aimed at investigation of dependency between redundant packet degrees and SEC recovery effectiveness.

The experiment was aimed to confirm or deny the following hypotheses formulated by author.

2.7 Hypotheses on recovery effectiveness functional dependency

Hypothesis 1. For a given amount of redundant packets V and given packet loss ratio such a value of D exists, at which SEC provides highest recovery for any possible combination of erasure positions.

Hypothesis 2. To achieve most efficient packet recovery by SEC, the degree D of redundant packets should be specified as a function of present PLR. Most effective recovery could be achieved only statistically, because each particular coded sequence might recover all erasures or not recover any erasure, depending on the position of erasures.

Axioma. Using given parameters, SEC cannot guarantee recovery of particular erased packets. The only lowest and highest limits can be specified. The theoretical lowest limit is number of erasures, since total amount of lost packets cannot exceed number of erasures.

It is worth to notice, that for LT codes lowest limit of recovery is limited to loss of all K original packets. The lowest limit for SEC is explained by systematic property.

Let author to prove hypothesis 1. The recovery probability of a source symbol Q by an erasure code is determined due to equation (2.13). The full probability of source packet recovery is a function of probability delivery q of a coded packet. Value of q is a function of *PLR* in the channel. The value of Q is determined as a sum of neighbor's recovery probability. Neighbor's recovery probability, in turn, is a sum of multiplications of polynomials of degree 1 and higher from q (2.12). Number of summands is determined as number of coded packets into which the given packet is involved as a neighbor, and number of neighbors in these coded packets. Consequently, the recovery probability of a source packet Q is a function of three arguments: V, *PLR* and D.

This proves functional dependency of recovery effectiveness as an argument of D

and *PLR*, however it doesn't prove that the function has one or more maximums.

To prove that, enough and necessary to prove that the function is smooth and has no breaks.

This is proved by the fact that this function is defined as a sum of polynomials of degree equal or higher than 1 (2.13).

It can be proven that this function has minimums at border values of D=0 and D=K.

Indeed, if SEC code is built using 0 neighbors for redundant packets, recovery effectiveness is equal to 0, as such a code is not able to recover any erasure. Opposite, if D=K, then in case of 2 and more erasures the code is not able to recover any packet, too. Redundant packets with D from the interval $0 < D < K$ are able to recover lost packets more effectively.

As soon as the recovery effectiveness function is smooth, has no breaks and gets minimal values at the border values of the argument D, such a value of D exists at which recovery effectiveness achieved by SEC has a maximum at given V and *PLR*.

2.7.1 Experimental evaluation of SEC codes effectiveness

To confirm the hypothesis 1 experimentally author performed the following experiment. The parameters K = 10, V = 2 were taken for the experiment. The values were taken not too high in order to make total number of possible combinations of erasures' positions relatively small, as the exhaustive search was used for determining of the number of recovered packets for each combination. Over K original packets were constructed SEC-coded sequences for values of D = {1, 2, 3, 4, …, K–1}. Original packets' indexes were taken as {1,2,..,D} for the first redundant packet and {K–D, K–D+1,..,K} for the second redundant packet.

During the experiment were emulated erasures of 2 packets among total N=12 packets. Positions of erasures were varied $R = C_{12}^{2} = 66$ times to get recovery results for any possible combination of 2 erasures' positions from 12 transmitted packets (*PLR* = 16.67%).

The following experimental results of packet loss recovery were obtained (Table 2.1.). Modeling was performed using Erasure TestBench program.

Table 2.1. Results of statistical experiments for recovery effectiveness estimation by SEC. The parameters of SEC are: K=10, V=2, PLR=16.67%, D={1, 2, 3, 4, …, K–1}.

D	*avg_prob*	*avg_rec*	*F*
1	0.832	8.636364	4
2	0.8512	8.878788	9
3	0.86144	9.060606	16
4	0.865536	9.181818	25
5	0.865536	9.242424	36
6	0.865843	9.484848	45
7	0.867909	9.545455	48
8	0.854062	9.424242	45
9	0.847058	9.121212	36

The columns in the Table 2.1 have the next meanings:

avg_prob (*average probability*) – theoretical recovery probability calculated due to (2.13);

avg_rec (*average recovered*) – average number of recovered packets over total number of runs $R = C_{12}^{2}$. Number of runs corresponds to total number of possible combinations of positions of 2 erasures among 12 transmitted packets;

F – number of combinations of erasures' positions, at which SEC recovered 100% of source packets.

It could be seen from the Table 2.1 the highest effectiveness of SEC is achieved at

D=7. All three criterion indicating recovery effectiveness shows the maximum at this value of *D*=7.

Important result is that at D=7 the full recovery was achieved for 48 combinations of losses' positions from 66 possible, which is more than 70% from the total possible outcomes. It is natural that the worst result is achieved for *D*=1. This result was considered and argued above.

The results of modeling fit the theoretical prepositions performed above and statistically confirm the hypothesis 1 for the private case of *K*=10, *V*=2, *PLR*=16.67%.

For practical applications it is required to use quite high values of K, in order to reveal statistical properties of the erasure codes. However, at high K it is getting troublesome to perform exhaustive search for all values of *D* / *PLR* / combinations of losses. To argue, the exhaustive search for 5 erasures from 100 packets would require C_{100}^{5} =75287520 runs. To avoid that but still be able to study the characteristics of erasure codes, author performed simplified statistical experiment.

A key point of the experiment is that positions of erasures are random, but the number of runs for each set of parameters of interest is done multiple times, and effectiveness results got for each run are averaged over total number of runs. The correctness of the experiment is assured by confidence interval. The number of runs in the present experiment is chosen as *R*=1000. Coding by SEC is performed for varied parameters *D*, *PLR*, and fixed *K* = 100, which is considered high enough to reveal statistical properties of erasure codes and satisfy practical limitations for end-to-end latency dictated by DTV systems.

The results of the experiment are given in the Tables 2.2 – 2.3.

In the Tables 2.2 – 2.3 the percentage of of the recovered packets is presented by *M*, and the standard deviation of *M* is shown by σ [40]. *S*ince the experiment is statistical, *M* and σ is an estimate of the mean value of the packages recovery effectiveness. In connection with the statistical nature of the experiment, the issue of results accuracy and reliability is arising. For relevance of the results a confidence interval is calculated for certain "sufficiently large" value of confidence probability. "Sufficiently large" value of confidence probability, according to [40] can be considered as the probability of greater than 0.9. this value is defined as $\beta = 0.95$. To solve this issue author uses the rule, that the value of M is the sum of R independent and identically distributed random variables, and according to the central limit theorem, for sufficiently large N, its distribution law is close to normal.

Table 2.2. Percent of packets recovered by SEC. K=100, R=1000, D=10..40.

	D=10		***D*=20**		***D=30***		***D*=40**	
Overhead: 5%								
PLR, %	*M, %*	*σ*	*M, %*	*σ*	*M, %*	*σ*	*M, %*	σ
1	*99.45*	*0.05*	99.72	0.05	*99.85*	*0.04*	99.87	0.03
3	*98.2*	*0.09*	98.81	0.09	*99.27*	*0.09*	99.41	0.09
5	*96.81*	*0.11*	97.55	0.12	*97.75*	*0.14*	97.53	0.17
8	*94.73*	*0.14*	94.87	0.15	*94.25*	*0.17*	93.65	0.17
Overhead: 7%								
1	*99.58*	*0.05*	99.83	0.04	*99.91*	*0.02*	99.97	0.02
3	*98.64*	*0.08*	99.29	0.08	*99.67*	*0.06*	99.81	0.05
5	*97.34*	*0.11*	98.42	0.12	*98.93*	*0.13*	98.78	0.17
8	*95.59*	*0.15*	96.27	0.18	*95.54*	*0.22*	94.47	0.24
Overhead: 10%								
1	*99.64*	*0.05*	99.92	0.03	*99.99*	*0.01*	99.99	0
3	*98.96*	*0.08*	99.63	0.06	*99.89*	*0.04*	99.94	0.02
5	*98.08*	*0.11*	99.24	0.1	*99.62*	*0.08*	99.63	0.11
8	*96.64*	*0.15*	97.88	0.17	*97.79*	*0.26*	96.03	0.31
Overhead: 15%								
1	*99.78*	*0.04*	99.95	0.02	*100*	*0.01*	99.99	0
3	*99.3*	*0.07*	99.89	0.03	*100*	*0.02*	100	0
5	*98.84*	*0.1*	99.9	0.05	*99.95*	*0.03*	99.95	0.05
8	97.82	0.14	99.32	0.12	*99.08*	*0.24*	96.55	0.37

Table 2.3. Percent of packets recovered by SEC. K=100, R=1000, D=50..90

	D=50		***D=60***		***D=70***		***D=80***		***D=90***	
PLR, %	*M, %*	*σ*	*M, %*	*σ*	*M, %*	*σ*	*M, %*	σ	*M, %*	*σ*
					Overhead: 5%					
1	*99.95*	*0.01*	99.99	0.01	*99.99*	*0*	100	0	*100*	*0*
3	*99.52*	*0.09*	99.31	0.12	*99.07*	*0.14*	98.25	0.14	*97.67*	*0.11*
5	*96.86*	*0.18*	96.29	0.16	*95.76*	*0.14*	95.59	0.12	*95.51*	*0.1*
8	*93.31*	*0.18*	93.14	0.16	*93.22*	*0.16*	93.11	0.17	*93.11*	*0.16*
					Overhead: 7%					
1	*99.97*	*0*	100	0.01	*100*	*0*	100	0	*100*	*0*
3	*99.87*	*0.05*	99.72	0.08	*99.39*	*0.11*	98.67	0.14	*97.88*	*0.13*
5	*98.15*	*0.21*	97	0.21	*96.24*	*0.18*	95.72	0.13	*95.61*	*0.13*
8	*93.71*	*0.22*	93.56	0.21	*93.55*	*0.22*	93.44	0.2	*93.44*	*0.19*
					Overhead: 10%					
1	*100*	*0*	100	0	*100*	*0*	100	0	*100*	*0*
3	*99.97*	*0.03*	99.93	0.04	*99.8*	*0.08*	99.17	0.13	*98.28*	*0.14*
5	*99.07*	*0.18*	97.99	0.23	*96.72*	*0.21*	95.96	0.16	*95.79*	*0.13*
8	*94.56*	*0.29*	94	0.26	*93.92*	*0.23*	93.58	0.21	*93.76*	*0.21*
					Overhead: 15%					
1	*100*	*0*	100	0	*100*	*0*	100	0	*100*	*0*
3	*100*	*0*	99.98	0.02	*99.94*	*0.05*	99.58	0.1	*98.66*	*0.15*
5	*99.68*	*0.11*	98.9	0.2	*97.66*	*0.23*	96.57	0.2	*96.04*	*0.15*
8	*94.64*	*0.34*	93.93	0.3	*93.53*	*0.28*	93.33	0.28	*93.35*	*0.25*

In practice, even with a relatively small number of terms (10-20), the statistical law of the sum can be approximately regarded as normal [12].

Since the calculation of confidence intervals for each value of M is a very laborious task, the author calculates the confidence interval for the most "unreliable" values of M, which corresponds to the largest value of standard deviation σ. This value is the result obtained for D=40, PLR=8%, and the magnitude of the excess of 15% (see Table 2.2.). The value of σ=0.37. From the Table 14.3.1 of [40] let us find the desired value t_β , critical to the normal law of quadratic deviation, which must be deferred to the right and left of the dispersion center so as to the probability of the hit in the area selected was equal to β. The value of 0.95 corresponds to the value t_β =1.960.

Then, according to [40], the confidence limits are calculated as:

$$m_1 = M - t_\beta \cdot \sigma \ ,$$

$$m_2 = M + t_\beta \cdot \sigma \ .$$

The primary interest is the $t_\beta \cdot \sigma$, the product, which determines the confidence intervals. In the example presented, the product is equal to 0.7252, which represents 0.75% of the value of M=96.55, corresponding to σ=0.37. Therefore, as far as in the calculations above the largest value of σ among the all derived from the experiment is used, it is arguable that with probability β=0.95, the value of M hits in the range, distant from the average value measured, not more than 0.75%.

The data from the Tables 2.2 – 2.3 are presented in a graphical view below in the Fig. 2.6 – 2.8, for PLR=3%, PLR=5% and PLR=8% (colored bars: green, violet and dark-blue respectively). Each diagram is built for a different overhead value. The approximating curves are polynomials of 5th degree (added automatically by MS Excel).

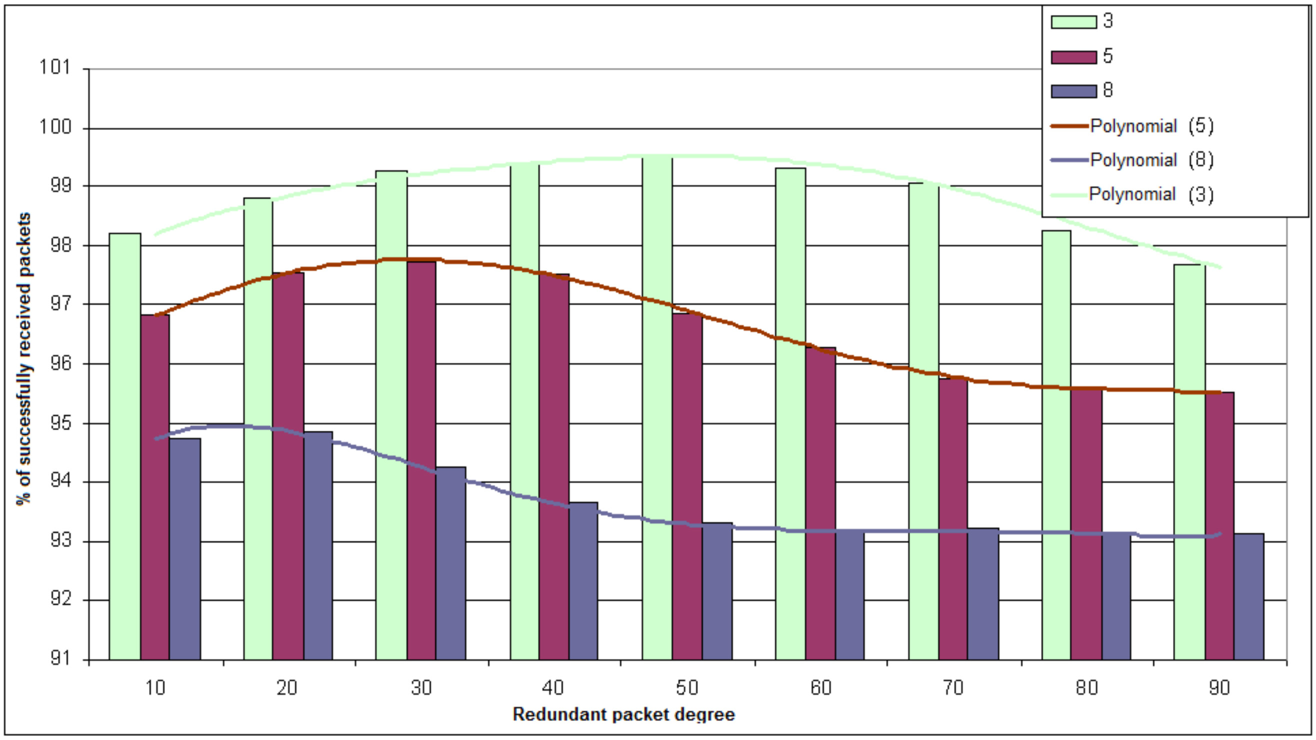

Fig. 2.6. Results of the statistical experiment of SEC effectiveness. Overhead=5%

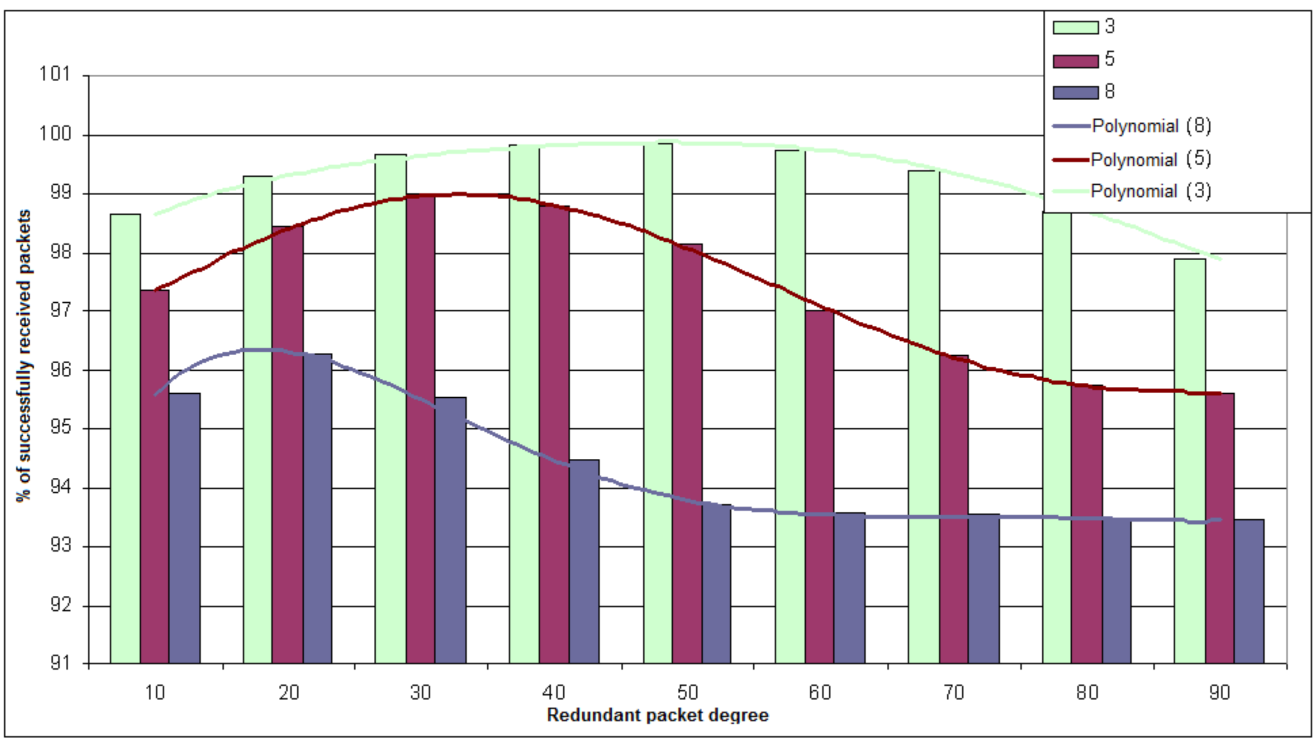

Fig. 2.7. Results of the statistical experiment of SEC effectiveness. Overhead=7%

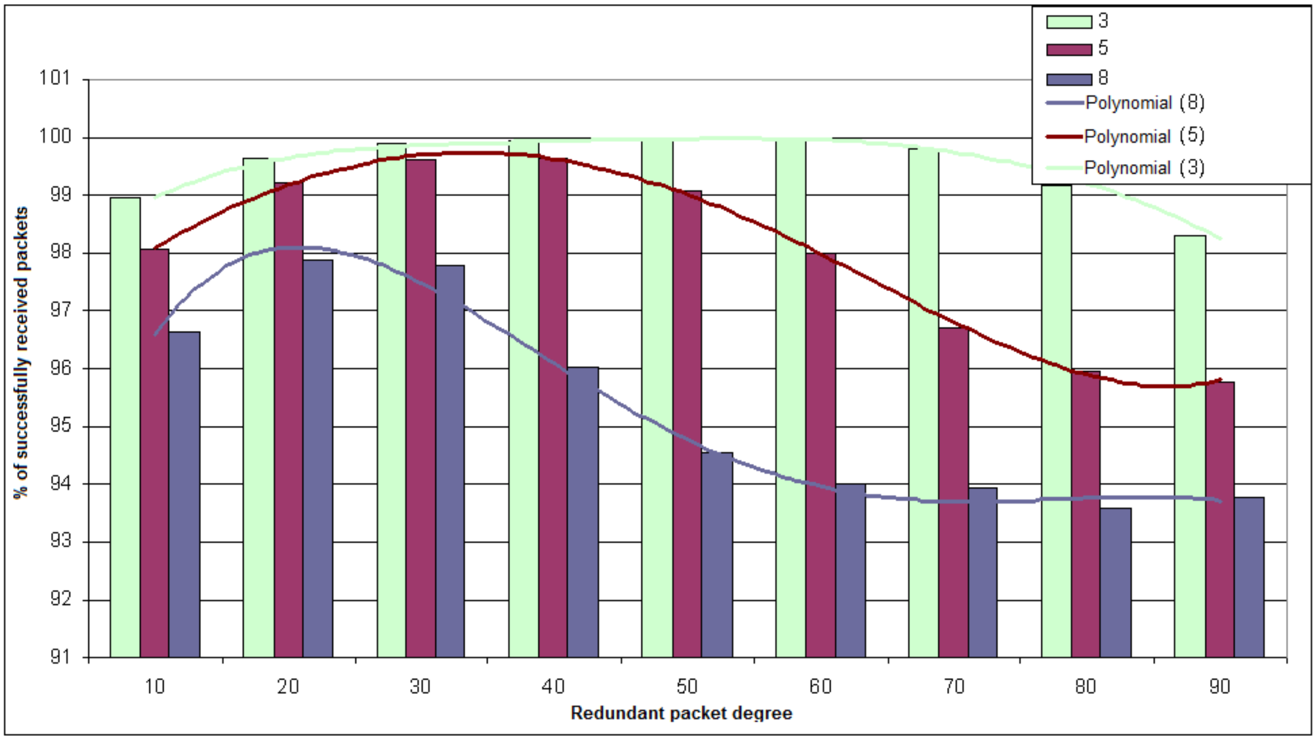

Fig. 2.8. Results of the statistical experiment of SEC effectiveness. Overhead=10%

The major result of the experiment is that the SEC effectiveness significantly depends on the selection of redundant packet degrees *D*. Next factor, is that actual PLR must be taken into account, as maximum of SEC effectiveness is achieved at different values of D, depending on the PLR condition. In case of high PLR the peak of the effectiveness function shifts to the lower value of *D*.

Next important result is that at just 10% of overhead the SEC is able to recover up to 4.62% (with a standard deviation of 0.08%) lost packets from 5%, which could be treated as a very effective recovery from severe loss ratio at a tolerable overhead.

Important to notice, that with increase of overhead the distance between curves for each PLR gets higher, which again confirms the necessity to take the factors assumed in the hypotheses above to be involved into the code construction to get more effective results.

The presented experimental results match the theoretical calculations performed by author above, and confirm the formulated hypotheses.

By the results of the performed experiment author created a table of approximating polynomials for SEC effectiveness *F* as a function of redundant packet degree *D* for given transmitting conditions *PLR* and application-specific limits on overhead *H*. The polynomials are presented in the Table 2.4.

Table 2.4. Approximating polynomials for the SEC effectiveness as a function of redundant packet degree *D* and overhead *H*.

Overhead *H*, in % to *K*	PLR	Approximating polynomials
$0<H\leq 5$	$0<PLR\leq 5$	$F(D) = 0.0007D^5 - 0.0182D^4 + 0.1603D^3 - 0.7267D^2 + 1.969D + 96.805$
	$5<PLR<8$	$F(D) = -0.0009D^5 + 0.0209D^4 - 0.151D^3 + 0.2113D^2 + 0.8691D + 95.862$
	$PLR\geq 8$	$F(D) = 0.0013D^5 - 0.039D^4 + 0.4255D^3 - 2.0691D^2 + 3.9091D + 92.503$
5<H<10	0<PLR≤5	$F(D) = 0.0006D^5 - 0.0154D^4 + 0.1468D^3 - 0.7184D^2 + 2.0146D + 97.208$
	5<PLR<8	$F(D) = -0.0013D^5 + 0.0339D^4 - 0.2919D^3 + 0.7738D^2 + 0.3058D + 96.525$
	PLR≥8	$F(D) = 0.002D^5 - 0.0623D^4 + 0.7131D^3 - 3.6809D^2 + 7.6532D + 90.955$
H≥10	0<PLR≤5	$F(D) = 0.0006D^5 - 0.0186D^4 + 0.2032D^3 - 1.0507D^2 + 2.6861D + 97.13$
	5<PLR<8	$F(D) = 0.0002D^5 + 0.0005D^4 - 0.0219D^3 - 0.1783D^2 + 1.7589D + 96.535$
	PLR≥8	$F(D) = 0.0007D^5 - 0.0313D^4 + 0.4986D^3 - 3.3373D^2 + 8.4829D + 90.976$

2.8 Conclusions for Chapter 2

The erasure codes are introduced in Chapter 2. Inventor of erasure codes is Luby, who was the first to suggest erasure codes called in his honor “Luby Transform” (LT) codes.

The LT codes has the following general characteristics of interest within the scope of the present work:

– coded symbols are generated independently from each other; this is so-called “fountain” property of LT codes. It means that receiver can use any coded symbol for recovery of the original message;
– original symbols have equal priority;
– length of original symbols could be any, but must be equal for all symbols involved into the coding process;
– LT codes are effective at significantly high number of original symbols involved into coding process.

The mentioned above properties are of a high value, and enables LT codes to be used for any input data format. Fountain property of LT codes has high significance for “many - to - many” distribution task [20, 38].

However, these characteristics are not completely suitable for multimedia data transmitting, particularly for live TV broadcasting. The major reason of this is a severe end-to-end delay which is introduced by LT coding, as coding is efficient only over large amount of source symbols. This specificity is blocking for utilization of LT codes in multimedia transmitting systems.

Next code considered in the present chapter is COP# 3. Benefit of COP# 3 code is that they have systematic nature (original packets transmitted as they are), and relatively simple in implementation. Next positive point for COP#3 is significantly lower latency comparing to e.g. LT codes, determined recovery possibilities.

Still, COP# 3 code do not provide opportunities for ranking of input packets by priority, and need quite large overhead to effectively protect the data.

In the present chapter, author suggested and investigated systematic erasure codes. The hypothesis that coding symbols' degrees should be determined on the basis of packet loss ratio and maximum allowed amount of redundant packets has been formulated. The statistical experiment performed by author confirmed the hypothesis and illustrated the dependency of losses recovery from coding symbols' degrees being used.

On a basis of the obtained experimental results for systematic erasure codes, author compiled the table of approximating polynomials for packet recovery effectiveness as a function $F(D)$ of coding packets' degrees D.

Considering theoretical prepositions, performed statistical analysis and obtained experimental results, let us summarize the problems of existing erasure codes in application to live DTV systems:

- Unacceptably high latency of TV signal, caused by the necessity to buffer of significant amount of packets, either by sender or receiver, to perform coding and decoding operations, respectively;
- Lack of mechanisms to account unequal priority of original symbols, which is typically a case for digital multimedia streams compressed by currently used formats;
- Fountain property looses it's significance for live TV broadcasting, because it's not an option for receiver to wait extra “drops” of data in order to recover a stream. This option is relevant for offline distribution systems, e.g. peer-to-peer networks, torrent systems.

Author's conclusion is, that erasure codes have perspectives to be applied in DTV systems, but their certain characteristics must be improved to fit the requirements of live transmitting.

3 Design of new erasure codes for TV over IP: Non-random erasure codes (NEC)

3.1 Introduction to NEC. Coding Algorithm

To solve the problems of streaming digital broadcasting over IP networks novel erasure codes were offered by the author. They were termed as “nonrandom erasure codes” (NEC). The development of nonrandom erasure codes was motivated by the fact that at low K values the statistical features at the basis of LT code theory fail to function. Therefore, it is necessary to control the quantity of initial characters’ entries into coding packets to provide for increased robustness and higher code effectiveness. Based on probability analysis for successful packet recoveries with various methods of generating graph construction cited in p. 2.1.4, there was developed an algorithmic NEC construction procedure.

The number of coding packets for NEC may be potentially infinite, but for practical applications a concrete limited value *N* should be applied, therefore the simulations were done using fixed values of *N* based on the maximum allowed coding packets redundancy in relation to the number *K* of initial characters as $N = K \cdot \beta$, where β is slightly more than 1 and should be taken on the basis of the practical limitations.

The encoding process for NEC is initiated in the same way as for LT codes: the input message is divided into *K* blocks of equal lengths. The degree distribution is then constructed. To determine the quantity of packets of a certain degree NEC code applies robust distribution $\mu(d)$ according to the formula (2.8). However, the distribution is modified by introducing additional checking packets for reference packets. Let us designate the modified distribution as $\mu'(d)$. To determine the degree of additional checking packets is applied the result for systematic erasure codes (ref. p. 2.6). Additional packets are generated separately for the highest priority data encoded as degree 1 coding packets, and for lower priority packets involved as neighbors for degree 2, 3 and so on encoding. To determine the checking packets degrees the value of D is applied, at which the approximating polynomial $F(D)$ (ref. table 2.4) has its maximum, which means that the value of D is determined at which the approximating polynomial derivative goes to zero. The resulting D value is scaled for the number of coding packets with the degree of $n[d_i]$, for which the checking packets are generated. Let us designate this D value as D_i. The number of packets of this degree is determined by means of permitted redundant packets quantity $V=N-K$ as $Z_i = \frac{V \cdot n[d_i]}{K}$.

In the conformance with the $\mu'(d)$ degree distribution probability, the array D is filled with values in such a way, that D_0 represents the amount of coding symbols with one neighbor, D_1 – with two neighbors, and so on. The input and output packet sequences are designated as *IN* and *OUT*, respectively. Square bracket indexes are used to address a particular packet in either *IN* or *OUT*. Number of times an input packet is involved into the coding process as a neighbor is termed "*protection degree*" of an input packet.

The NEC encoding process is described below.

Step 1.

Determine the amount of packets subject to encode with degrees $d_{0,} {}_{\ldots,}\ d_{max}$ as $n(d_i) = \mu'(d_i) \cdot N$.

Step 2.

Select $n[d_0]$ original packets with the *highest priority*. Encode them as packets with degree 1. This could be written in pseudo-C language as:

```
for(i = 0; i < n[d_0]; i++)
{
  OUT[i] = IN[f_i];
},
```

where f_i means the index of original packet in the *IN*.

Step 3. Perform the following procedure over the highest priority packets selected in Step 1:

```
for(i =0; i <Z_0 ;i++) {
        r= rand(R_min); R[r]++; OUT[i + n[d_0]] = IN[r];
        for(h=0;h< D'_0 ;h++) {
                r = rand(R_min); R[r]++;
                OUT[i + n[d_0]] = OUT[i + n[d_0]] XOR IN[r];
        }
}
```

Step 4.

Let's denote by $SUBSET_d$ a subset of input symbols with the number of references equal to *d*. Subset of symbols with minimal and maximal *d* are designated as $SUBSET_{MIN}$ and $SUBSET_{MAX}$, respectively.

i = 2; y = 0;

Step 4.1. Do while(y < n[d_i]) {

Step 4.1.1. Generate *k* -th coded packet as a result of *XOR* over one packet from $W_{MAX,}$ one packet from $W_{MAX-1,}$…, one from W_{L-1}, and $W_{MIN,}$ where $L=MAX — d_{i:}$

$OUT[k] = W_{MAX}[j_0]$ *XOR* $W_{MAX-1}[j_1]$ *XOR… XOR* $W_{L-1}[j_{L-1}]$ *XOR* $W_{MIN}[j_L]$;

k = k + 1;

Step 4.1.2. For each $(j_t, t=0…L)$ *do* $\{R[j_t] = R[j_t]+1\}$ (here j_t are indexes of original packets in *IN*).

y = y + 1;

}

Step 4.2.

Update subsets $W_{R,}$ update minimum and maximum values of R.

$i = i+1$.

If i reached $d_{max}+1$ or total amount of coded symbols reached N, EXIT.

If all original packets are involved into the coding process - goto Step 5, else - goto Step 4.1.

Step 5.

```
m=1;
while(k<N) {
        for(v=0;v <Z_m ;v++)  {
                r= rand(n[d_m]); R[r]++; OUT[k] = IN[r];
                for(h=0;h< D'_m ;h++) {
                        r = rand(n[d_m]); R[r]++;
                        OUT[k] = OUT[k] XOR IN[r];
                }
        }
        k++;
        If all packets from n[d_m] are involved into coding process at this step, m++;
}
```

***EXIT*.**

In the algorithm given the higher initial packet priority, selected from $W_{MIN,}$ the lower has to be the coding packet degree at which it will get initially involved into encoding.

The most important is that the rows in initial packet index matrix *J* ideally should be linearly independent, or at least have a minimum of linearly dependent vectors-rows combinations.

$$J = \begin{bmatrix} j_0^1 & \dots & j_{d_2-1}^1 & 0 & \dots & 0 \\ j_0^2 & \dots & \dots & j_{d_3}^2 & 0 & 0 \\ \\ \dots & \dots & \dots & \dots & \dots & \dots \\ j_0^{N-n[d_0]} & \dots & \dots & \dots & \dots & j_{d_{max}}^{N-n[d_0]} \end{bmatrix}.$$

The example of algorithm for selecting "neighbor" indexes among W_R subsets can be found in [34]. As a result of the algorithm operation a generating graph of systematic structure is produced. The example of such a graph is given in the Fig. 3.1.

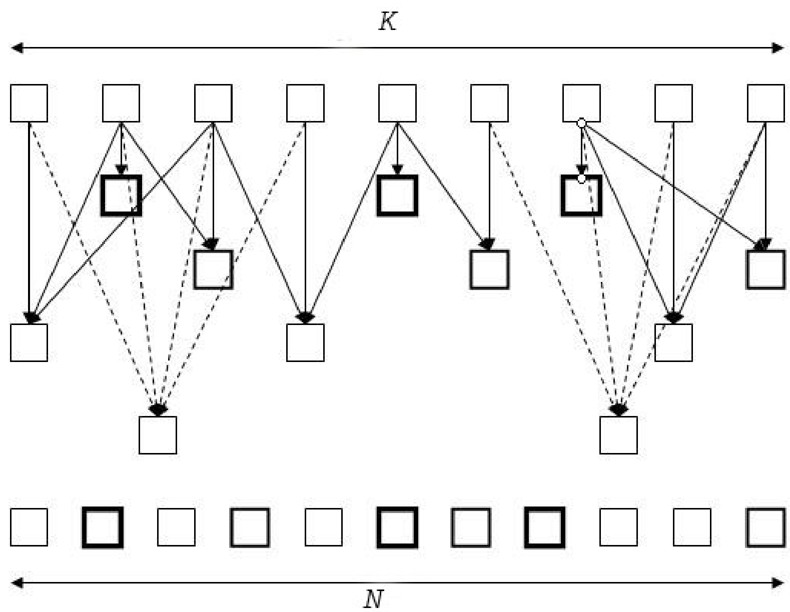

Fig. 3.1. The Example of NEC code generating graph

3.1.1 NEC Code definition by generating matrix

NEC code, in common with all other erasure codes, may be preset as a generating matrix. There are two possible variants to preset a generating matrix:

- presetting a matrix of correspondence between coding characters' indexes and initial characters ("input-output" matrix);

- presetting a matrix of correspondence of initial characters' indexes for coding characters 'arrays ("output-input").

Given below is the example of "output-input" matrix for NEC code for K=100, N= 105. In the right parentheses a total quantity of elements' entries as neighbors into coding characters is given. Initial packets indexes are preset in square brackets.

Example of "Output-input" generating matrix for erasure code presetting:

```
OUT->IN matrix: N = 105
OUT idx, refs, degree
0: [0] (1)
1: [1] (1)
2: [2] (1)
3: [3] (1)
.........
17: [17] (1)
18: [18] (1)
19: [19] (1)
20: [20] (1)
21: [21] (1)
22: [22] (1)
23: [23] (1)
24: [0] [1] [2] [3] [4] [5] [6] [7] (8)
25: [8] [9] [10] [11] [12] [13] [14] [15] [16] (8)
26: [17] [18] [19] [20] [21] [22] [23] (7)
27: [11] [25] (2)
28: [18] [26] (2)
29: [15] [27] (2)
30: [7] [28] (2)
31: [5] [29] (2)
32: [21] [30] (2)
33: [8] [31] (2)
.........
56: [14] [42] (2)
57: [4] [48] [32] (3)
58: [21] [49] [28] (3)
.........
94: [6] [85] [26] (3)
95: [11] [86] [33] (3)
96: [10] [87] [24] (3)
97: [5] [88] [37] (3)
98: [8] [89] [40] (3)
100: [4] [98] [42] (3)
101: [16] [99] [31] (3)
102: [11] [100] [52] (3)
103: [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37]
[38] [39] [40] [41] [42] [43] (21)
104: [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [67]
[68] [69] [70] [71] [72] [73] (21)
105: [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87]
[88] [89] [90] [91] [92] [93](21)
```

3.2 Simulation experiment results for NEC codes

NEC experimental model was implemented within *Erasure TestBench* program in accordance with the above stated coding algorithm.

K, c and δ parameter values varied within the following ranges:

$K = \{100, 500, 1000, 5000\}$,

$c = \{0.05, 0.1, 0.2, 0.5\}$,

$\delta = \{0.05, 0.1, 0.2, 0.5\}$.

For every combination of the parameter triad a modified robust distribution was constructed and NEC coding utilizing the corresponding resulting distribution was performed. Along with it, for every run a quantity of output characters N generated by LT encoder was modified. The values of N were selected as $N=\{1.05 \cdot K, 1.1 \cdot K, 1.2 \cdot K, 1.3 \cdot K\}$, for every K. And, finally, for every code determined by the set of $\{K, c, \delta, N\}$ a packet loss simulation or coding characters loss simulation was performed. The packet loss ratio also varied successively taking the values from *PLR* vector, $PLR = \{0, 3, 5, 8, 15\}$.

The experiment results represent averaging over R=1000 runs for every set of the parameters varied. The average recovered packets ratios resulting from the experiment are given in the Appendix B. The graphical diagrams of the table data representing NEC code effectiveness estimation are shown in the Appendix B, Fig. B.1 – B.3.

The most significant result is that NEC allows the recovery of up to 100% of initial packets even at low K in the range of 100-500 packets. Additionally, in comparison to LT codes NEC provide for higher decoding robustness even at high packet loss ratios.

Another significant factor is that NEC codes require a small coding packets redundancy to achieve a reliable decoding of 97 - 100 %. Even at redundancy value of 5%, and at a small *PLR*, of course, in some cases the codes allow the recovery of 100%

of the initial packets.

That high effectiveness of NEC becomes possible due to the deterministic code structure and defining the degree of checking packets as the values at which the maximum approximating function of systematic erasure code recovery effectiveness is achieved, as defined in the table 2.4.

These features, along with the capability of ranking initial packets and assigning them protection priority, make NEC applicable for TV over IP systems to fight packet erasures during network transmission.

3.3 NEC practical application aspects in TV over IP systems

The major problem of applying error-correcting codes, and erasure codes in particular, in IP networks is that they cannot support IP stack protocols modifications. The reason for this lies in the fact that network devices operating in the frames of IP protocol stack are utilized by millions of users around the world, and forcing them to switch from existing network adapters to a new standard of network devices is apparently impracticable, and even unnecessary. In connection with this, there arises a need to develop an alternative solution that would allow applying an erasure coder-decoder at the cost of the least possible TV head-end station architecture modifications, and without the need of hardware changes, i.e. it could be implemented within IP stack.

As the solution of these issues a technology offering to perform NEC erasure code encoding at the level of MPEG-2 TS.

The general TV over IP system architecture applying NEC encoder is presented in Fig. 3.2. The architecture offered is original and has no any analogues.

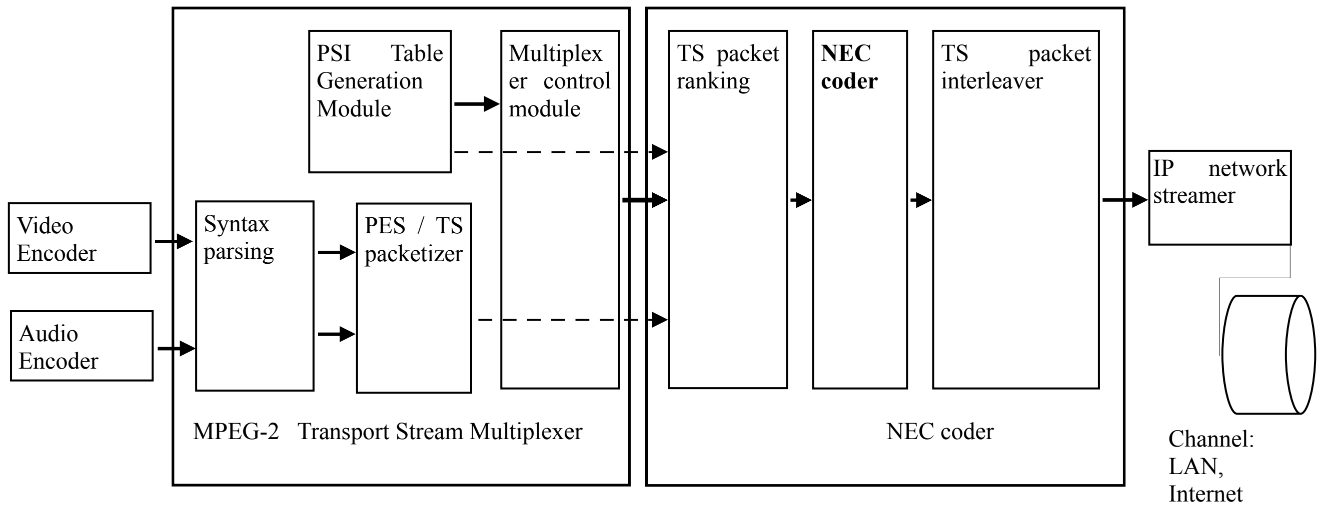

Fig. 3.2. TV over IP system architecture using NEC for reliable data delivery

It has to be mentioned that the logical conceptual architecture offered can have various implementations. The program can be implemented as a component with a corresponding data transfer and control call program interface, or as DirectShow filter, or as a software-hardware implementation with FPGA *(Field Programmable Gate Arrays*).

The author sees his task as designing a logical architecture aimed at supporting the architecture with a required error-correcting coding function meeting the existing standards, rather than considering a particular implementation method.

MPEG-2 transport stream has been selected as a NEC integration level for another reason: MPEG-2 TS is considered de-facto a worldwide standard. MPEG-2 TS-based component can be integrated into majority current DTV systems.

A NEC coding variant at the level of UDP/RTP protocol content was also considered, but was rejected for the following reasons:

- UDP packet size, *MTU* (*Maximum Transfer Unit*), can have one value for a network adapter at one workstation, but a different one at the other, which creates difficulties with selecting a packet size when encoding using erasure codes. But the TS packet size is fixed by the standard and is equal to 188 byte [55].

- a typical MTU value is around 1500 byte. Apparently, for erasure codes the larger the initial packet size, the longer total end-to-end delay and memory buffering burden. It is objectionable to decrease initial packets *K* quantity, since erasure codes are the more effective the larger is *K* value, as it was shown in the experiment performed in chapter 2 of the present work. Artificial packets "cutting" into smaller ones is equally inadvisable, because, as it is going to be explained further, there is a need of coding characters numbering, which means reducing packet payload byte number, and the smaller is a packet, the larger the reduction ratio. The TS packet length presents an adequate balance in-between these two limiting conditions [50];

- UDP/RTP multimedia data broadcasting standard [70, 71] allows for MPEG-2 TS broadcasting as carry-over data. Therefore, the architecture offered does not narrow capabilities and does not break the integrity of TV over IP architecture concept. The use of NEC at the UDP/RTP level would lack generality and bring about the need for parsing modules within network broadcasting component, which is not common. And that would mean the use of parsing modules for two types of streams – MPEG-2 TS, for maintaining generality, and elementary streams in case of direct broadcasting.

Let us consider in more detail a "hybrid" transport stream multiplexer module of MPEG-2 – a NEC coder. A parsing module plays a crucial role in the architecture offered (Fig. 3.2). A parsing module, as a rule, is implemented within the frames of transport stream multiplexers. Its function in the multiplexer is to determine the boundaries of audio and video frames so that the multiplexer was able to perform frames packing into PES packets aligning the packets start. In the suggested architecture the parsing module ought to have additional functionality: to determine packets type (*I*, *P*, *B* for video, or audio frame) and transmit the information to the PES/TS packetizing multiplexer module.

PES/TS packetizing module packs elementary video and audio streams into PES, and then TS packets, after that transmitting the TS packets to the multiplexer control module along with the information on the packets content type.

During an "ordinary" implementation a multiplexer performs audio and video synchronization control, system time control - PCR (*Program Clock Reference*), transport packets forwarding, so that to meet the requirements of T-STD model (*Transport System Target Decoder*), and timely forwarding of PSI tables.

Within the present architecture the multiplexer control module is also to transmit the information on transport packets content to the NEC coder ranking module. The transport stream processing is further performed by NEC coder module comprising the three logical components: TS packets ranking module, the NEC coder itself, and an interleaver (refer to Fig. 3.2).

The NEC coding algorithm is described in p. 3.1.

The transport packets ranking modules and the interleaver functionality will be discussed below.

3.3.1 Packets ranking in MPEG-2 Transport Stream

Transport packets contain different types of data:

- Ordering information required for the stream demultiplexing: *PAT* (*Program Association Table*) и *PMT* (*Program Map Table*) [55]. The packets transmitting this information are duplicated in the stream at the frequency of a t least 100 ms;

- audio data;

- video data;

- Filler-packets to align the general bit rate of the transport stream.

The concept of applying NEC in TV over IP consists in the following.

The initial transport packets are offered to be ranked by their priorities as follows (in decreasing order):

- Audio packets and packets carrying video key frames (I-frames);
- Packets carrying video P-frames;
- Packets carrying video B-frames.

Filler-packets are to be excluded for the purpose of saving the bandwidth. At the receiver's end NEC coder replaces all the packets failed to be recovered by filler-packets.

The ordering information packets are not NEC encoded. Since MPEG-2 TS standard supports transmission of additional data via descriptors, this method is offered to be applied for NEC checking parameters transfer from encoder to decoder. In this case, it will be possible to start decoding when connecting to the broadcast at any given time, with a delay not exceeding 100 ms.

Priority packet sorting is performed by the corresponding module in the NEC encoder (ref. Fig. 3.2). It is also responsible for incoming packets buffering. After accumulating K packets (service packets not included), NEC coding algorithm starts. In this process it's significant that encoding takes into account the packets' priorities. It is this distinctive feature of the system that is designed to increase the error-correcting coding effectiveness and to improve audio-visual perceptual quality.

For degree 1 coding packets the highest priority transport packets are selected, for degree 2 coding packets - priority 2 transport packets, and so on, until all initial packets get involved in the encoding process, according to NEC encoding algorithm, as given above. In each set of K initial packets there may turn out to be a different number of packets having this or that priority, and it may be possible to transfer packets between the groups, both from a higher priority group with downgrading into a lower priority one, and vice versa.

3.4 Performance estimation for LT and NEC encoding and decoding algorithms

The most resource-intensive operation among erasure code encoding and decoding algorithms is the XOR operation, since it has to be performed over large amounts of data. Therefore, the number of XOR operation calls may be considered as a trustworthy complexity (performance) estimations. The total XOR operation calls number at a processor is determined by the processor bit register and the packet length. For the 32 bit processors with x86 architecture the number of XOR calls over four bytes ought to be calculated. However, within the frames of a particular application, at a fixed length of the packet over which the encoding is performed, the measure of XOR calls per packet can be applied.

Decoding operation requires an analogue to encoding XOR operation calls number, thus, for rough estimations it is permissible to limit oneself to encoding computational complexity, assuming encoding and decoding to be equally complex.

Within the frames of the present work the basic aim of LT and NEC algorithms computational complexity estimation is to determine computational resources required for multimedia data streams encoding at a specified bit rate in real time mode.

By means of *Erasure TestBench* program LT and NEC algorithms computational complexity estimation was performed. The estimation was performed for the values of K= 100, 500, 1000, 5000. The values of c=0.05 and δ=0.05 were selected as the values for which the highest recovery effectiveness had been achieved both for LT and NEC codes. The results of LT and NEC algorithms computational complexity estimation are given in the table 3.1.

In the table 3.1 the parameter *C* represents the number of XOR operation calls per packet. The *CPU_time* estimate is the quantity of processing time seconds spent for 1000 encoding runs.

Table 3.1. LT and NEC algorithms computational complexity estimation results

	LT		NEC	
K	*C*	*CPU Time*, sec	*C*	*CPU Time*,sec
100	857,198	0	550,000	1
500	5873,002	4	4234,952	5
1000	12787,626	10	9895,812	14
5000	62920,497	67	38485,822	98

For each of the codes considered, LT and NEC, there was performed 1000 runs. The packet length of 188 byte was specified. The experiment was performed at the computer equipped by Mobile Intel Pentium processor with 2,2 GHz clock frequency.

As may be seen in the table, NEC code requires a smaller number of XOR operation calls than LT code, but takes away a larger processing time. This is due to additional computational complexity of neighbor indexes selection for encoding.

Let us calculate the NEC code capability to encode a multimedia data steam. Based on delay restriction in the "receiver-transmitter" chain the values of *K*=100 – *K*=500 present practical interest. For these *K* values encoding time increases linearly, so any value can be selected.

Let us consider the case of *K*=100. To calculate the input bit rate of multimedia stream the packet length (188 x8 bit = 1504 bit) is to be multiplied by the number of packets *K*=100. The product equals to 150400 bit. Then one has to divide the amount by the value of *CPU_time*. Division by one will result in previous value. Since the result of *CPU_time* = 1 sec represents the sum of encoding lengths of 1000 runs, we have to multiply the stream bit rate by 1000 to get a maximum possible bit rate value that allows real time NEC encoding. The multiplication gives us the result of 150400000 bit, or ~143 megabit per second, which is a very good result, taking into account that encoding was performed by a processor that is far short of the modern processors. In fact, the result achieved allow to state with confidence that using NEC codes in IPTV systems in real time mode is possible.

RAM requirements are quite easy to define: the number of initial packets *K* has to be multiplied by a packet length. For *K*=500, in the case of *MPEG-2 TS* encoding, the value will amount to 94 000 byte. Some additional memory space will be required for temporary storage of the probability density tables, code generating matrix, and others, however, these expenses are the slightest if compared to the RAM memory space available with state-of-the-art computers.

3.5 Conclusions for Chapter 3

NEC error-correcting erasure codes suggested by the author are able to provide for high recovery effectiveness of packet losses. NEC application for TV over IP systems is intended to improve TV programs perceptual quality. To maximize the use of NEC codes it was also necessary to develop a method allowing their application for TV over IP systems. The problem of error-correcting codes application in IP networks is connected with impossibility to modify IP stack protocols and networking equipment. In view of this circumstance a method of NEC codes application in TV over IP systems at the level of transport stream was suggested. The method does not require any essential modifications of the current broadcasting systems. The architecture developed by the author requires no more than a simple modification of transport stream program multiplexer MPEG-2 TS, supporting for additional functionality to transmit ancillary data on the content type of the generated transport streams.

The solution developed features the following advantages:

- based on the structural analysis of multimedia data stream NEC encoder is able to automatically determine which packets exactly are of higher significance for audio and video perception quality on the whole, and provide for the appropriate protection degree of certain packets;

- the system architecture does not violate the principles of TV over IP systems and merely expands the functionality within the available engineering solutions. If desired, NEC modules can be easily removed from the data processing chain, and its ordinary configuration restored.
- the method suggested is implemented within the frames of existing standards regulating multimedia data streams syntax. As for third-party modules modifications, but insignificant MPEG-2 TS multiplexer modification is required to realize the adaptation (again, within ISO standard), and additional implementation of ways to communicate the data types of transport packets to NEC multiplexer encoder;
- abstraction from particular video and audio data compression standards. All the standards, be it MPEG-2 Video, H.264, MPEG Layer II Audio or AAC, or any other, if the format is supported by multiplexer, it can be automatically applied for broadcasting in the system with NEC encoding, without any additional effort for system modifications;
- Abstraction from the level of interaction with network devices. This job, like in any TV over IP system, is taken upon by a web-cast component, independently taking the decision which particular IP stack protocol to utilize, what MTU size to specify, and similar specific tasks.

The NEC computational complexity estimation demonstrated that resource requirements to perform the encoding and decoding operations are not critical for modern computers.

Conclusion

Summary and achievements

The thesis work solved an urgent scientific and technical challenge – development of error-correcting codes allowing to overcome the problem of multimedia data packet loss in digital TV over IP systems.

The principle thesis work results are as follows:

1. Up-to-date approaches to DTV systems implementation have been surveyed: compression standards and digital multimedia data transmission standards, network protocols used for IP network transporting, general issues of TV over IP systems architectural design.
2. The research results revealed the problems of data loss during transmission actual for TV over IP systems, origin of the problems. The problems to be solved were formulated: reducing the amount of packet losses in TV over IP systems. The search for an adequate error-correcting code is an obvious necessity to solve the problem formulated.
3. Taking into consideration the specifics of TV over IP systems, the requirements to error-correcting codes applied in the systems were formulated. It was found, that classical error-correcting codes, Reed-Solomon codes, convolutional codes, Turbo codes included, do not meet the requirements of TV over IP to solving the task of packet erasure correction, neither are they applicable for TV over IP systems.
3. The prospects of using error-correcting codes in TV over IP systems was grounded. As the object of study Luby Transform erasure codes were selected, being the most conceptually holistic known analogues.

4. LT encoder and decoder were software implemented, *Erasure TestBench* program was developed to estimate the effectiveness of erasure codes using the simulation of erasure channel. The analysis of LT code features demonstrated that LT codes have some deficiencies that make their application to solve the problems stated in the study quite complicated, which was proved experimentally.
5. Author performed comprehensive study of statistical properties of systematic erasure codes, proving that the characteristic of systematic erasure codes are common for erasure codes in general.
6. As a significant contribution is evaluated formulating of hypotheses on functional dependency between recovery effectiveness as an argument of redundant packet degrees, present packet loss ratio condition and available overhead. The statistical experiment confirmed the hypotheses and allowed to obtain fundamental polynomial approximations for the effectiveness function, which were used for NEC design.
7. The need for developing new erasure codes able to meet the specifics of TV over IP systems and multimedia streams structure was proved.
8. Author designed new erasure codes, termed NEC codes. Contrary to analogues known, NEC codes provide for higher recovery effectiveness at small lengths of initial data sequences, which makes them a promising candidate to be applied in TV over IP systems. NEC codes apply deterministic algorithm for choosing the code symbols' degrees based on the polynomial functions table developed experimentally. An experimental NEC code implementation was performed during the study. The results of simulation obtained by using the *Erasure TestBench* program proved that the codes developed possess high

erasure corrective effectiveness.

9. Conceptual design of computer networks broadcasting system architecture using NEC codes was performed. The architecture features high versatility in respect to its communication to third-party components, as it is developed to meet the requirements of current streaming broadcast systems.
10. Experimental computational complexity estimation for NEC encoding was performed. The results demonstrated that resource requirements for the implementation of NEC encoding and decoding are not critical for up-to-date computers.
11. The algorithms and methods developed by author allow for considerable reduction of data loss in digital broadcasting systems over IP networks.

Outlook

The future work should be connected to practical implementation of the designed erasure codes in TV over IP systems. It's getting the particularly high interest with introducing of TV over 3G networks broadcasting – the general trend for digital television now is to use IP-based networks as a primary delivery medium.

The present work left opened the question of objective quality comparison for different erasure coding approaches.

Certain perspectives exist for SEC codes application to DTV transmitting and storage.

References

1. Artushenko V., Sheluchin O., Afonin M. Digital compression of video and audio. – Moscow.: Dashkov & Co, 2003. – 426 c.
2. Argyropoulos S., Tan A.S., Thomos N., Arikan E., Strintzis M.G.. Robust Transmission Of Multi-View Video Streams Using Flexible Macroblock Ordering And Systematic LT Codes. Proc. of the 3DTV International Conference: True-Vision, Capture, Transmission, and Display of 3D Video (3DTV-CON 07). Kos Island, May 2007.
3. Byers J., Luby M., and Mitzenmacher M.. A digital foundation approach to asynchronous reliable multicast // IEEE J. Selected Areas Comm., V. 20, pp. 1528–1540, 2002.
4. Bahl L. R., J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate, IEEE Trans. Inform. Theory 20(2): 248–287 (March 1974).
5. Berrou C., Glavieux A., and Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo codes // In Proc. of 1993 IEEE Int. Communications Conf. – Geneva, 1993. – P. 1064–1070.
6. Blahut R. Theory and practice of error-correcting codes: Russian translation – Moscow, “Mir”, 1986. – 576 c.
7. Bolot J.C. End-to-end packet delay and loss behavior in the Internet // In Proc. Of ACM Sigcomm'93. – San Fransisco, CA. – P. 189–199.
8. Costello D. J., J. Hagenauer, et al. Applications of Error-Control Coding. IEEE Transactions on Information Theory, 1998. – V. 44. – № 6 (October). – P.77–102.
9. Dholakia A. Convolutional codes. John Wiley & Sons, England, 2001, P.431.

10. Elias P., Coding for noisy channels, Proc. IRE Conv. Rec., 1955, part 4, P.37–46.
11. Furht B. A Survey of Multimedia Compression Techniques and Standards. Part II: Video Compression // Journal of Real-Time Imaging, Vol. 1, No. 5, November 1995, pp. 319-338.
12. Gnedenko B. Course of probability theory: – Editorial URSS, 2001. – 320 p.
13. Gray R.M. Entropy and Information Theory. Stanford University, Springer-Verlag, New York, 1990, P.284.
14. Gallager R. G. Low Density Parity Check Codes. M.I.T. Press, 1963, Monograph.
15. Hamming R.S. Error detecting and error correcting codes. – Bell Syst. Tech. J., 1950. – V.29. – P. 147–160.
16. Hoffman D., G.Fernando, et al. RTP Payload Format for MPEG1/MPEG2 Video, - RFC 2250, January 1998.
17. Hoffman D., Leonard D. et al. Coding Theory The Essentials: Marcel Dekker Inc, New York, 1992.
18. IABM Broadcast and Media Technology Industry Guide 2010/11 ISBN 978-0-9552420-5-2, The Global Media Alliance, UK, Pensord Press. 2010.
19. Luby M. LT Codes // Proc. of the 43rd Annual IEEE Symp. on Foundations of Computer Science (FOCS). – 2002. – P. 271–282.
20. MacKay D.J.C., Information Theory, Inference, and Learning Algorithms. [WEB-resource]: **http://www.inference.phy.cam.ac.uk/mackay/itila/book.html**. – 14.05.2008.
21. MacKay D.J.C. Fountain codes // IEE Proc.-Commun., 2005. – V. 152. – № 6 (December). – P. 1062–1068.
22. Mathys P.. Theory and Practice of Error Control Coding. [WEB-resource]. –

http://www.colorado.edu/~ecen5682/index.html. – 11.10.2007.

23. Maymounkov M. Online codes. [WEB-resource]. – http://www.csc.cs.nyu.edu/~petar/oncodes.pdf. – 12.10.2007.
24. Mitzenmacher M. Digital Fountains: A Survey and Look Forward. – Information Theory Workshop 2004.
25. Peterson W. W. and Weldon E.J., Jr. Error-Correcting Codes, M.I.T. Press, Cambridge, MA, 1961.
26. Pennebaker, W. B., J. I,. Mitchell, C. Fogg and D. McCall, MPEG Digital Video Compression Standard, Chapman & Hall, 1997.
27. Proakis J. Digital communications. - Russian Edition, Moscow.: Radio I Svyaz, 2000.
28. Puri A. and Chen T. Multimedia Systems, Standards and Networks: Marcel Dekker, 2000.
29. Puducheri S., Kliewer J., Fuja T. E.. Distributed LT Codes. Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA, 2006.
30. Rao, K. R. and 3. 5. Hwang, Techniques and Standards for Image, Video and Audio Coding, Prentice Hall, 1997. P. 382.
31. Richardson I. E. G. H.264 and MPEG-4 Video Compression. John Wiley & Sons, England, 2003. P.281.
32. Richardson I.E.G., Video Codec Design, John Wiley & Sons, 2002.
33. Shinkarenko K., Vlcek K. Architecture of Channel Coding Effectiveness Estimation System // In Proc. of 6th Electronic Circuits and Systems Conference. – Bratislava, 2007. – P.165 – 169.
34. Shinkarenko K., Vlcek K. Design of Erasure Codes for Digital Multimedia Transmitting // In Proc. of 11th IEEE Workshop on Design and Diagnostic of

Electronic Circuits and Systems (DDECS08). – Bratislava, 2008. – P. 30 – 34.
35. Seong-Won Yuk, Min-Gyu Kang, et al. Adaptive Redundancy Control for Systematic Erasure Code Based Real time Data Transmission in Internet // **IEEE Transactions on Multimedia 3**(3): 366-374 (2001). – **http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/y/Yuk:Seong=Won**. – 11.08.2008.
36. Shokrollahi A. Raptor codes // IEEE/ACM Trans. Networking (TON). – 2006. – V. 14. – P. 2551–2567.
37. Taub H., Schilling D.I. Principles of Communication System. 2-nd edition, MC Graw-Hill Book Company, 1996.
38. Vargauzin V. Error-correcting coding in packet networks// Telemultimedia. 2005. – № 3. – pp.10 – 16.
39. Vargauzin V., Protopopov L. Turbo-codes and iterative decoding: principles, properties, application. // Telemultimedia. 2000. – №34. – pp.33 – 38.
40. Ventzel E.Probability theory: – Moscow, «Nauka», 1969. – 576 p.
41. Viterbi J., Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, IEEE Trans. Inform. Theory 13. – 1967, – P. 260 – 269.
42. Viterbi A.J. An Intuitive Justification and a Simplifed Implementation of the MAP Decoder for Convolution Codes, IEEE Journal On Selected Areas In Communications, 2003. – V.16. – №.2.
43. Vlchek K., Advanced Error-Control Coding Methods Enhance Reliability of Transmission and Storage Data Systems, RADIOENGINEERING, 2003. – V.12. – №. 1.
44. Wesel R. D., Convolutional Codes. University of California at Los Angeles, California. Wiley Encyclopedia of Telecommunications, 1993.
45. Wicker S. B., Error control systems for digital communication and storage.

1995.
46. Whatckinson J., The engineers Guide to Compression, Snill and Wilcox Handbook Series, 2003.
47. Wozencraft J. M.. Sequential decoding for reliable communication, Proc. IRE Conv. Rec. part 2, – 1957. – P. 11–25.
48. Yajnik M., Kurose J., Towsley D. Packet Loss Correlation in the MBone Multicast Network: ExperimentalMeasurements and Markov ChainModel //
49. Yeung R. W., A first course in information theory. Kluwer Academic/Plenum Publishers, 2002.
50. Yun Q. Shi, Huifang Sun. Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms and Standards. New York, CRC Press, 1999, 463 c.
51. ETSI EN 300 421 V1.1.2: Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services, (1997-08).
52. ETSI TR 102 033, Digital Video Broadcasting (DVB); Architechtural Framework for the Delivery of DVB services Over IP-based Networks, April 2002.
53. ISO/IEC 11172-1: Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: Systems. – 1993.
54. ISO/IEC 11172-2: Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video. – 1993.
55. ISO/IEC 13818-1: Information technology – Generic coding of moving pictures and associated audio information: Systems. Third Edition, 2007.

56. ISO/IEC 13818-2: Information technology - Generic coding of moving pictures and associated audio information: Video. – 2000.
57. ISO/IEC 13818-3: Information technology - Generic coding of moving pictures and associated audio information: Audio. – 2000.
58. ISO/IEC 13818-7:2003. Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC). – 1997.
59. ISO/IEC 14496-2: Information technology – Coding of audio-visual objects - Part 2: Visual. – 2001.
60. ISO/IEC 14496-4: Information technology – Coding of audio-visual objects - Part 4: Advanced Audio Coding. – 2001.
61. ISO/IEC 14496-10: Information technology – Coding of audio-visual objects -- Part 10: Advanced Video Coding. – 2005.
62. ISO/IEC 15938, Information technology – multimedia content description interface (MPEG-7). – 2002.
63. IETF RFC 3550: RTP: A Transport Protocol for Real-Time Applications. – 2003.
64. IETF RFC 3559: Multicast Address Allocation MIB. – 2003.
65. IETF RFC 3640: RTP Payload Format for Transport of MPEG-4 Elementary Streams. – 2003.
66. IETF RFC 3605: Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP). – 2003.
67. ITU-R Draft new Recommendation ITU-R BS. [10/20], Method for objective measurements of perceived audio quality. – 1998.
68. ITU-T H.263. Video coding for low bit rate communication. – 2005.
69. ITU-T Recommendation J.143. User requirements for objective perceptual

video quality measurements in digital cable television. – 2000.

70. RFC 768. User Datagram Protocol. – 1980.
71. SCTE 42 2002, IP Multicast for Digital MPEG Networks. – 2002.
72. Compression.ru (WEB-resource) – **www.compression.ru**. – 06.11.2010.
73. Telemultimedia.ru (WEB-resource) – **www.telemultimedia.ru**. – 19.07.2009.
74. **http://www.dvb.org/about_dvb/dvb_worldwide/czech_republic/index.xml** – 08.11.2010
75. **http://www.btv.cz/en/DVB-H** – 09.11.2010
76. Professional MPEG forum (WEB-resource) - **http://www.pro-mpeg.org/** - 24.11.2010.
77. Pro-MPEG Code of Practice #3 release 2 (WEB-resource) - **http://www.pleora.com/pdf/Vid-on-IP-CoP3-r2.pdf** - 04.10.2010.

List of author's publications

[1] Shinkarenko K. "Statistical multiplexing of MPEG-2 variable bitrate video streams for effective usage of channel bandwidth", Information Systems, Tomsk, Russia, pp. 123–129, 2004.

[2] Shinkarenko K. "One-pass video bitrate control algorithm," Trans. of Tomsk state University of Control Systems and Radioelectronics, Vol. 11, №3, pp. 110 – 114, 2004.

[3] Shinkarenko K. "Statistical multiplexing of variable bitrate video streams for effective usage of channel bandwidth", in Proc. of XLIII scientific student conf. "Student and scientific and technical progress": Information technologies, Novosibirsk, Russia, p. 52, 2005.

[4] Shinkarenko K. "The algorithm of MPEG transport stream multiplexing", in Proc. of "Sibresurs-2006" conf., Tumen, Russia, p. 90 – 94, 2006.

[5] Shinkarenko K. "Error correction in digital multimedia broadcasting systems", in Proc. of "Scientific session-TUSUR-2007" All-Russian conf., Tomsk, Russia, pp. 189 – 192, 2007.

[6] Shinkarenko K., Korikov A. "Optimization criteria of error-correcting codes for digital television", Computation technologies, Vol. 12, Novosibirsk, Russia, pp. 86 – 95, 2007.

[7] Shinkarenko K., Vlcek K. "Architecture of Channel Coding Effectiveness Estimation", in Proc. of 6th Electronic Circuits and Systems (ECS'07) international conf., Bratislava, Slovak Republic, pp. 165–169, 2007.

[8] Shinkarenko K., Vlcek K. "Design of Erasure Codes for Digital Multimedia Transmitting", in Proc. of 11th IEEE Workshop on Design and Diagnostic of Electronic Circuits and Systems (DDECS08), Bratislava, Slovak Republic, pp. 30 – 34, 2008.

Available at: **http://ieeexplore.ieee.org/Xplore/login.jsp?url=/stamp/stamp.jsp?arnumber=4538751&isnumber=4538735**.

[9] Shinkarenko K. "Application of erasure codes in digital television systems", in Proc. of 15th All-Russian conf. «Microelectronics and informatics-2008», Moscow, Russia, p. 248, 2008.

[10] Shinkarenko K., Korikov A. "Packet-loss recovery in computer networks", Trans. of Tomsk state University of Control Systems and Radioelectronics, Vol. 18, №2, pp. 107–111, 2008.

[11] Shinkarenko K. "Erasure error-correcting codes for digital television systems", in Proc. of Electronic Systems and Control Facilities-2008, Tomsk, Russia, pp. 181–188, 2008.

[12] Shinkarenko K., Korikov A. "Error-correcting coding of multimedia data in computer networks", Trans. of Tomsk Polytechnic University, Tomsk, Russia, , Vol. 313, № 5, pp. 37–41, 2009.

[13] Shinkarenko K., "Application of non-random erasure codes in IP broadcasting", Trans. of Tomsk Polytechnic University, Tomsk, Russia, Vol. 313, № 5, pp. 42–46, 2009.

Curriculum vitae

Eng. Shinkarenko Konstantin **shinkarenko@fai.utb.cz**

Education and training:

October 2006 – present time – doctoral degree programme

Faculty of Applied Informatics, Tomas Bata University, Zlin, Czech Republic,

2000 - 2005 - degree: engineer, diploma with excellence

Tomsk State University of Control Systems and Radioelectronics

Diploma thesis: Statistical Multiplexing of Digital Multimedia Streams

Job experience:

April 2006 – March 2010 : Software Developer

Mainconcept A.G., Aachen, Germany (www.mainconcept.com)

Main activities and responsibilities: digital multimedia software development

June 2004 - April 2006 : Software Developer

Elecard Ltd., Tomsk, Russia. (**www.elecard.com**)

Main activities and responsibilities: Software development

Personal skills and competences

Languages: Russian (native), English, Czech (communicative).

Programming languages: C++, C#.

Scientific grants and awards:

1. Grant of Czech Ministry of Education and Committee of Sports and Youth for doctoral degree programme in Tomas Bata University, Czech Republic.
2. The paper "Application of Erasure codes in digital multimedia systems" is awarded as the best scientific paper at International Conference "Microelectronics and Informatics-2008" held in Moscow Institute of Electronic Technique, 2008, April.

Appendix A

Experimental results and their graphical representation for LT codes

Table A.1. Recovered Packets Ratio for LT, c=0.05, δ =0.05

Overhead N/K, %	PLR, %	Initial Sequence Length K, packets			
		100	*500*	*1000*	*5000*
5	*0*	20	14.4	20	28.94
	3	3.8	9.88	5.47	7.36
	5	5.64	6.68	14.67	12.89
	8	18.28	16.4	20.97	4.95
	15	7.88	9.24	6.8	1.16
10	*0*	32	23.8	8.8	30.68
	3	21.24	18.68	6.06	2.17
	5	9.56	9.74	4.4	12.45
	8	3.68	5.58	6.27	15.4
	15	20.46	2.18	10.83	6.39
20	*0*	99	100	98.2	100
	3	2.88	90.69	99.02	100
	5	1	75.62	53.52	100
	8	42.98	23.96	27.89	100
	15	22.4	12.91	12.11	8.56
40	*0*	100	100	100	100
	3	62.52	100	100	100
	5	28.64	100	99.99	100
	8	59.58	86.78	96.74	100
	15	84.28	98.14	38.06	100

Table A.2. Recovered Packets Ratio for LT, c=0.2, δ =0.05

Overhead N/K, %	PLR, %	Initial Sequence Length K, packets			
		100	*500*	*1000*	*5000*
5	*0*	100	100	100	99.04
	3	97.6	98.08	98.12	98.78
	5	93.66	95.99	96.41	98.67
	8	89.18	89.13	88.07	97.32
	15	77.7	74.53	77.41	83.16
10	*0*	100	100	100	99.36
	3	98.54	98.85	98.8	98.3
	5	97.36	97.72	98.79	98.01
	8	95.1	96.53	98.52	98.16
	15	82.96	82.65	81.06	81.9
20	*0*	100	100	100	99.84
	3	99.46	99.25	99.3	99.14
	5	99.08	98.82	98.96	99.01
	8	98.16	98.62	99.38	98.8
	15	89.74	98.12	98.6	98.41
40	*0*	100	100	100	100
	3	99.78	99.9	99.92	99.61
	5	99.06	99.8	99.83	99.14
	8	97.44	99.64	99.76	98.87
	15	94.76	99.78	99.91	98.46

Table A.3. Recovered Packets Ratio for LT, c=0.2, δ =0.2

Overhead N/K, %	PLR, %	Initial Sequence Length K, packets			
		100	*500*	*1000*	*5000*
5	*0*	100	100	100	100
	3	97.7	97.78	97.65	97.32
	5	95.04	95.69	96.31	96.15
	8	92.32	89.98	90.7	85.34
	15	80.76	80.4	80.97	75.7
10	*0*	100	100	100	100
	3	98.12	98.97	98.78	98.43
	5	97.1	97.28	98.26	97.12
	8	95.76	96.42	96.81	92.92
	15	84.64	78.86	84.56	82.44
20	*0*	100	100	100	99.99
	3	99.22	99.72	99.69	98.13
	5	98.52	99.47	99.29	93.64
	8	97.98	98.9	99.21	97.88
	15	92.7	96.08	98.36	95.82
40	*0*	100	100	100	99.9
	3	99.86	99.61	99.55	85.58
	5	99.5	99.47	99.48	86.77
	8	99.66	99.27	99.34	85.86
	15	98.1	99.64	99.7	90.61

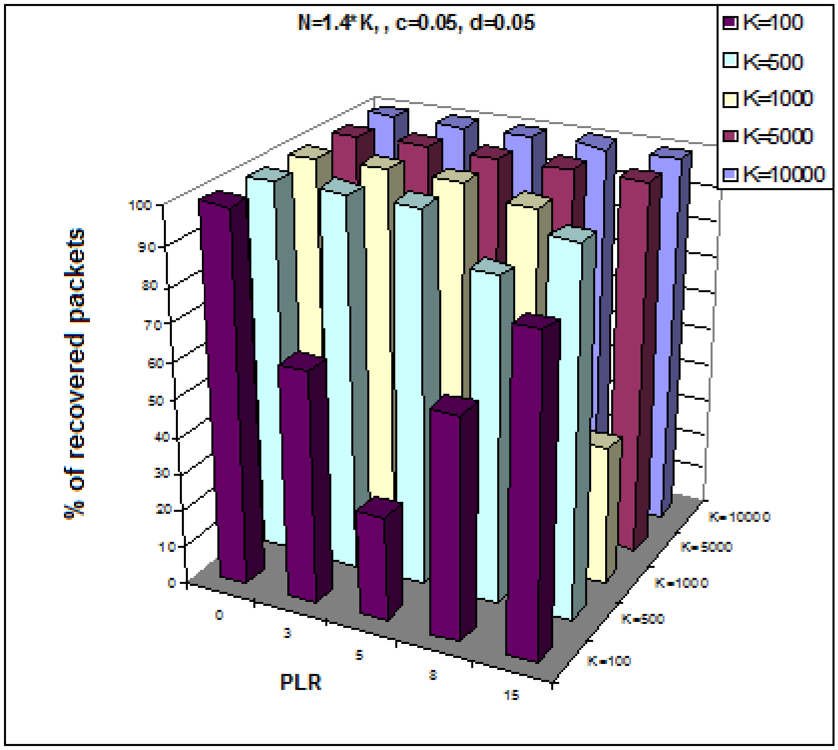

Fig. A.1. Percent of Successfully Recovered Packets for LT.
c=0.05, *δ*=0.05, *K*=100…10000, *PLR*=0…15, $N = K \cdot 1.4$

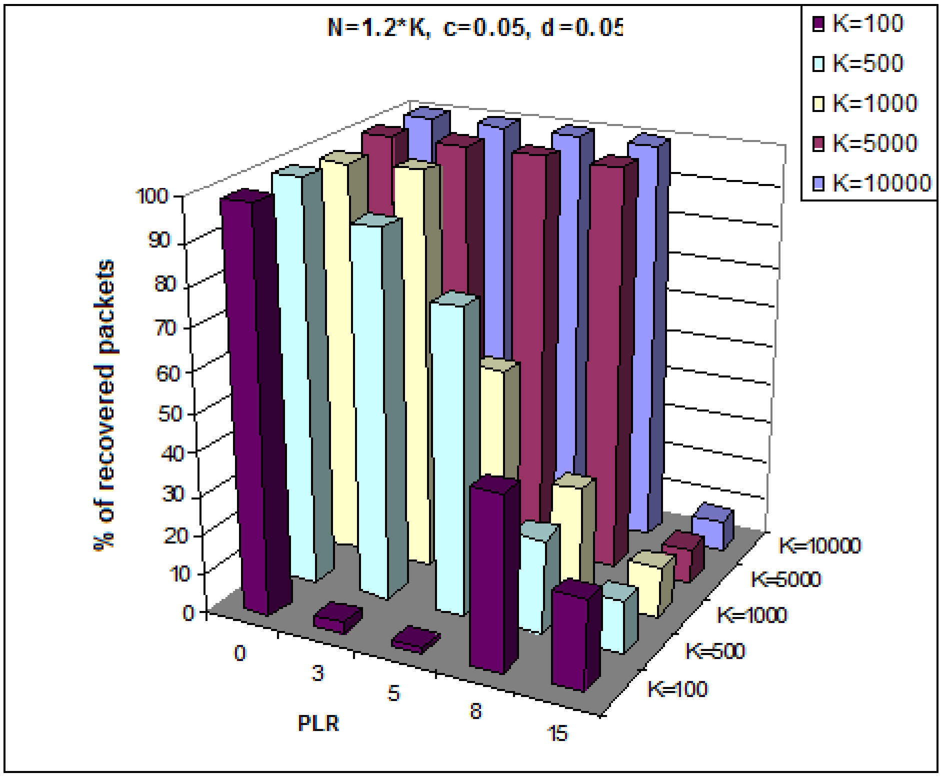

Fig. A.2. Percent of Successfully Recovered Packets for LT.

c=0.05, δ=0.05, K=100…10000, PLR=0…15, $N = K \cdot 1.2$

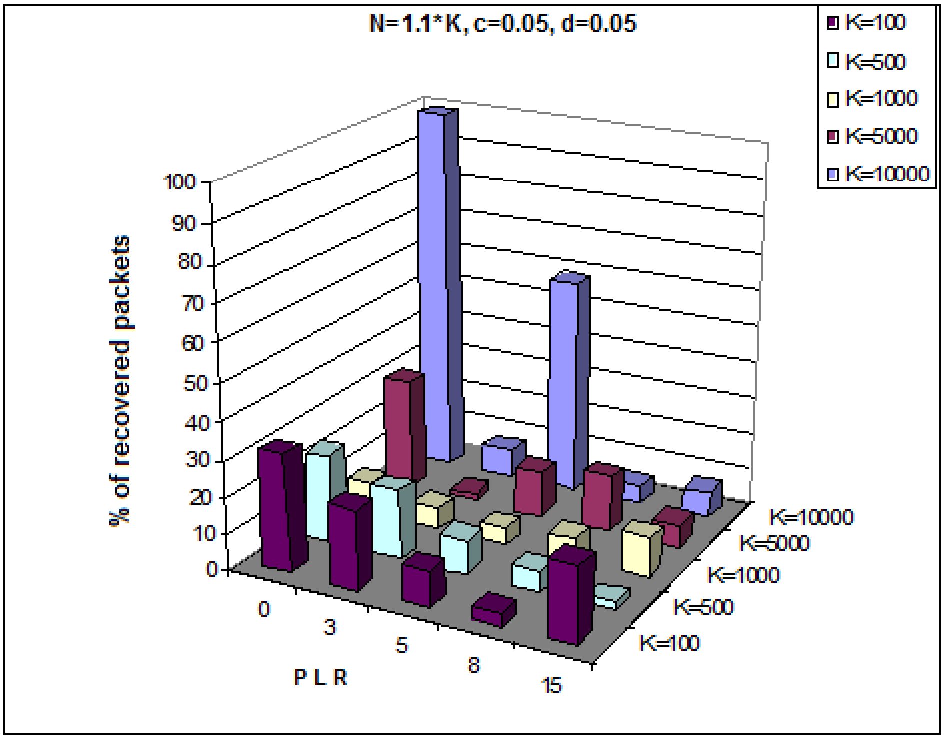

Fig. A.3. Percent of Successfully Recovered Packets for LT.
c=0.05, δ=0.05, K=100…10000, PLR=0…15, $N = K \cdot 1.1$

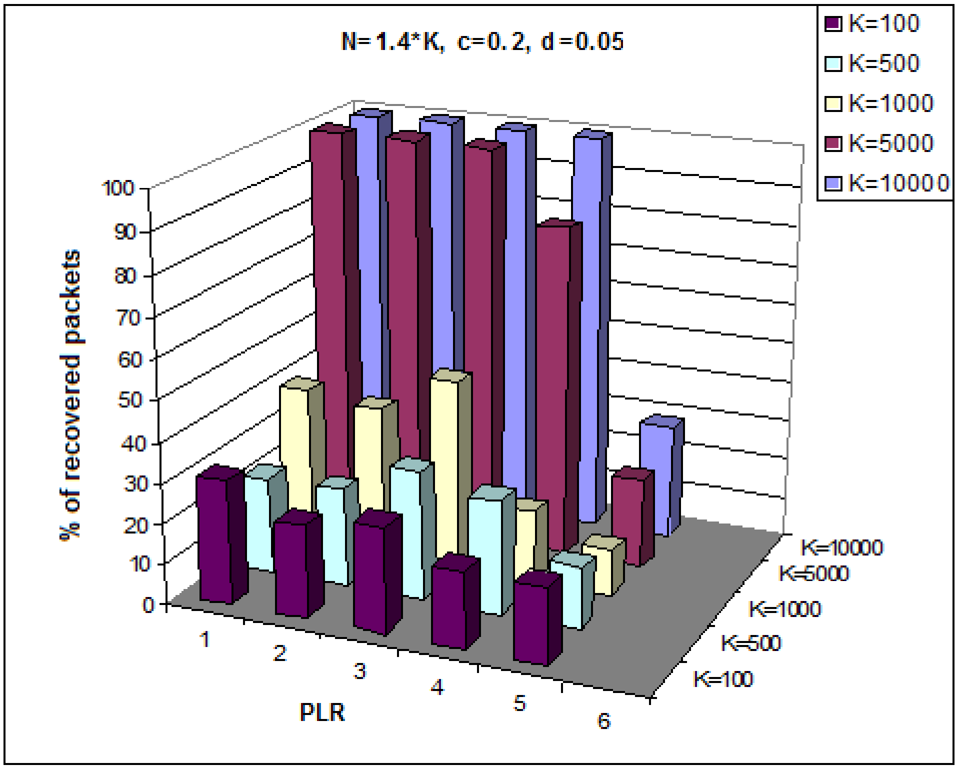

Fig. A.4. Percent of Successfully Recovered Packets for LT.

c=0.2, δ=0.05, K=100…10000, PLR=0…15, $N = K \cdot 1.4$

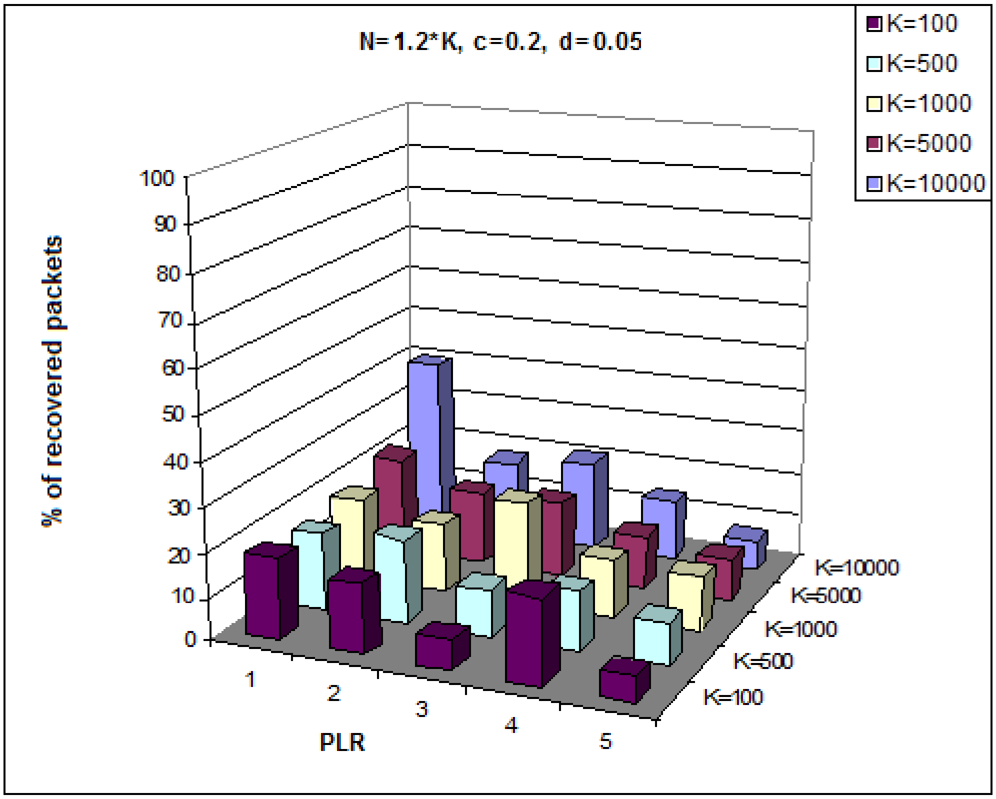

Fig. A.5. Percent of Successfully Recovered Packets for LT.

c=0.2, δ=0.05, K=100…10000, PLR=0…15, $N = K \cdot 1.2$

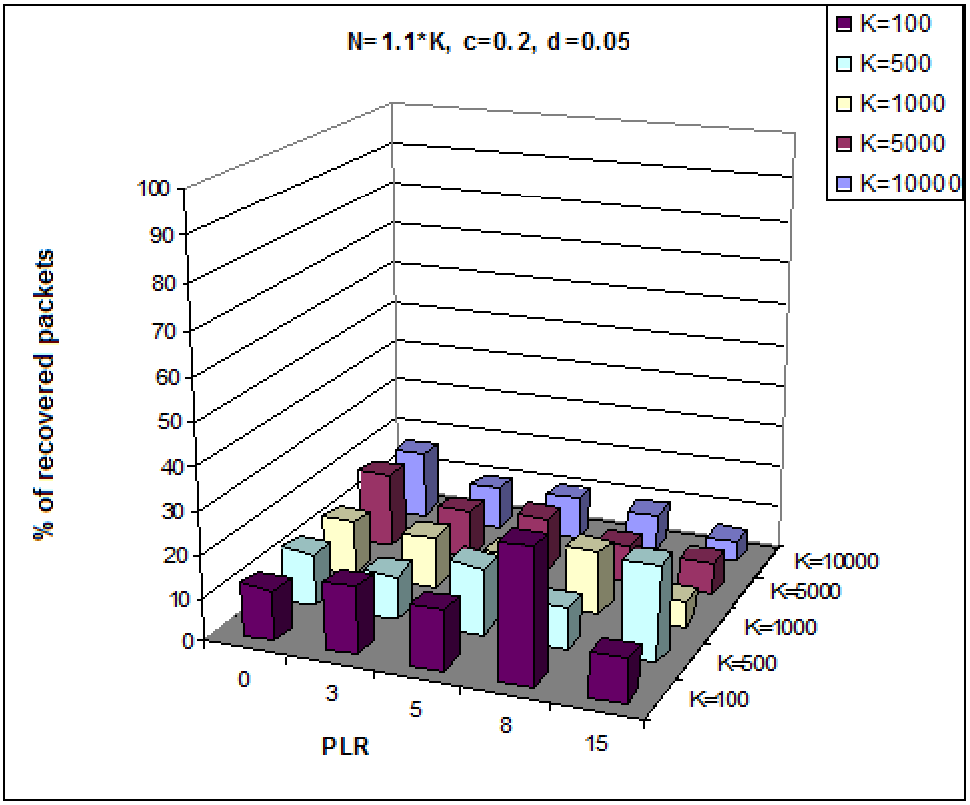

Fig. A.6. Percent of Successfully Recovered Packets for LT.

c=0.2, δ=0.05, K=100…10000, PLR=0…15, $N = K \cdot 1.1$

Appendix B

Experimental results and their graphical representation for NEC

Table B.1. Recovered Packets Ratio for NEC, c=0.05, δ =0.05

Overhead N/K, %	PLR, %	Initial Sequence Length K, packets				
		100	500	1000	5000	10000
5	0	100	100	100	89.27	100
	3	96.34	98.13	98.2	90.38	96.34
	5	95.92	96.66	95.96	88.91	95.92
	8	90.24	89.44	90.23	87.17	90.24
	15	79.18	73.94	80.49	82.45	79.18
10	0	100	100	100	99.72	100
	3	98.82	98.65	98.71	97.21	98.82
	5	98.2	98.07	98.22	96.19	98.2
	8	94.66	96.55	96.88	91.71	94.66
	15	78.1	79.96	83.8	82.01	78.1
20	0	100	100	100	99.04	100
	3	99.7	99.19	99.36	97.04	99.7
	5	98.66	98.84	99.2	99.29	98.66
	8	98.4	98.78	99.23	99.54	98.4
	15	92.34	96.81	99.53	98.1	92.34
30	0	100	100	100	96.86	100
	3	99.86	99.88	99.91	96.04	99.86
	5	99.8	99.81	99.89	79.96	99.8
	8	99.76	99.76	99.82	96.31	99.76
	15	97.12	99.72	99.95	94.23	97.12

Table B.2 Recovered Packets Ratio For NEC, c=0.2, δ =0.05

Overhead N/K, %	PLR, %	Initial Sequence Length K, packets				
		100	500	1000	5000	10000
5	0	20	14.4	20	28.94	16.3
	3	3.8	9.88	5.47	7.36	27.98
	5	5.64	6.68	14.67	12.89	10.7
	8	18.28	16.4	20.97	4.95	5.41
	15	7.88	9.24	6.8	1.16	1.55
10	0	32	23.8	8.8	30.68	99.99
	3	21.24	18.68	6.06	2.17	7.35
	5	9.56	9.74	4.4	12.45	58.74
	8	3.68	5.58	6.27	15.4	5.04
	15	20.46	2.18	10.83	6.39	6.96
20	0	99	100	98.2	100	100
	3	2.88	90.69	99.02	100	100
	5	1	75.62	53.52	100	99.99
	8	42.98	23.96	27.89	100	99.95
	15	22.4	12.91	12.11	8.56	7.35
30	0	100	100	100	100	100
	3	62.52	100	100	100	100
	5	28.64	100	99.99	100	100
	8	59.58	86.78	96.74	100	100
	15	84.28	98.14	38.06	100	99.99

Table B.3. Recovered Packets Ratio For NEC, c=0.2, δ =0.2

Overhead N / K, %	PLR, %	Initial Sequence Length K, packets				
		100	500	1000	5000	10000
5	0	20	14.4	20	28.94	16.3
	3	3.8	9.88	5.47	7.36	27.98
	5	5.64	6.68	14.67	12.89	10.7
	8	18.28	16.4	20.97	4.95	5.41
	15	7.88	9.24	6.8	1.16	1.55
10	0	32	23.8	8.8	30.68	99.99
	3	21.24	18.68	6.06	2.17	7.35
	5	9.56	9.74	4.4	12.45	58.74
	8	3.68	5.58	6.27	15.4	5.04
	15	20.46	2.18	10.83	6.39	6.96
20	0	99	100	98.2	100	100
	3	2.88	90.69	99.02	100	100
	5	1	75.62	53.52	100	99.99
	8	42.98	23.96	27.89	100	99.95
	15	22.4	12.91	12.11	8.56	7.35
30	0	100	100	100	100	100
	3	62.52	100	100	100	100
	5	28.64	100	99.99	100	100
	8	59.58	86.78	96.74	100	100
	15	84.28	98.14	38.06	100	99.99

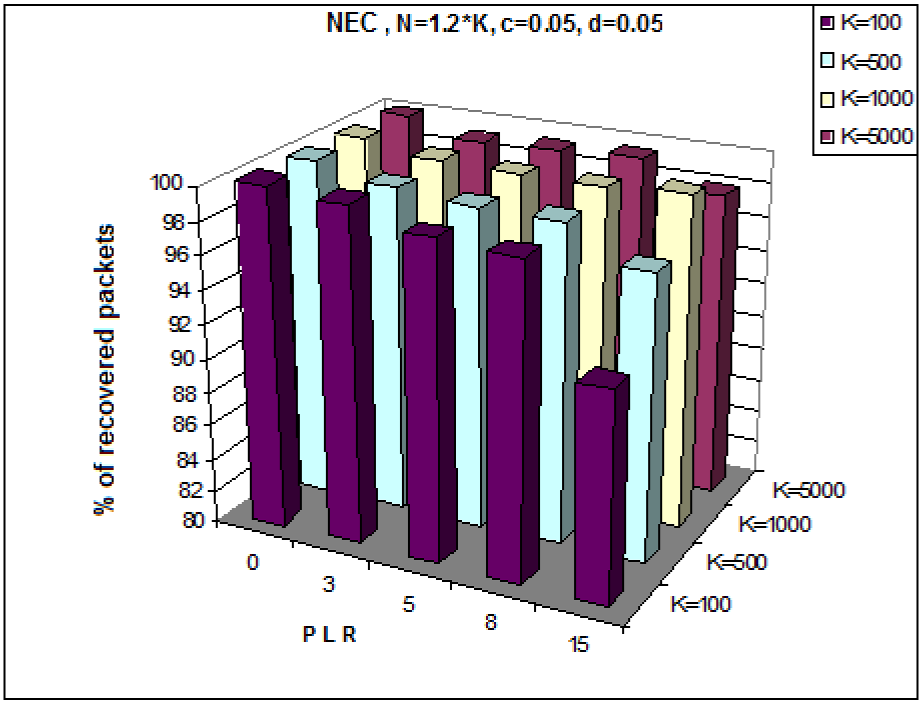

Fig. B.1. Percent of Successfully Recovered Packets for NEC.

K=100…5000, PLR=0…15, c=0.05, δ=0.05, $N = K \cdot 1.2$

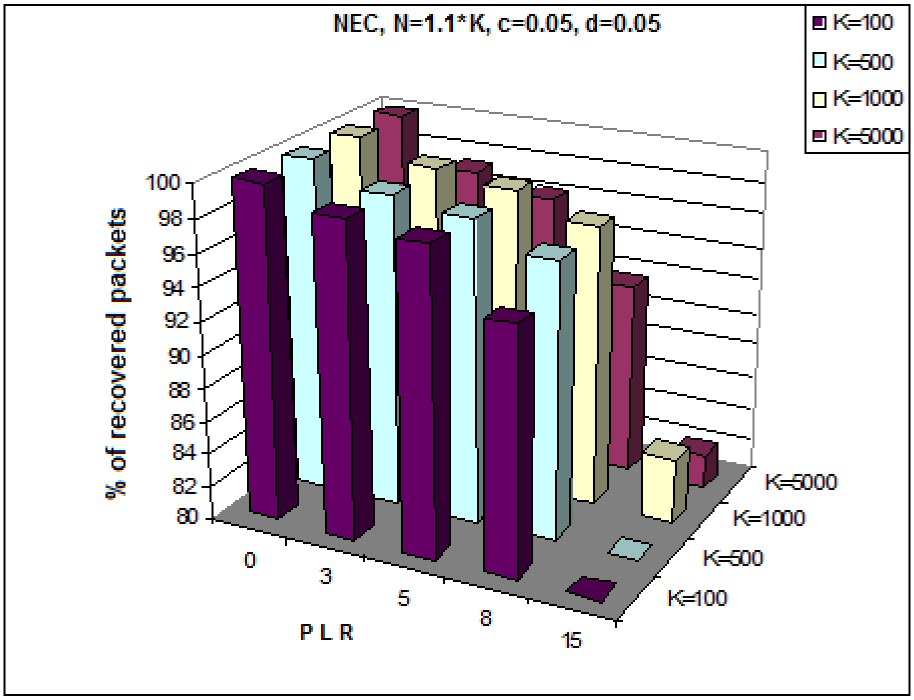

Fig. B.2. Percent of Successfully Recovered Packets (*Y* axis) for NEC.

K=100…5000, PLR=0…15, c=0.05, δ=0.05, $N = K \cdot 1.1$

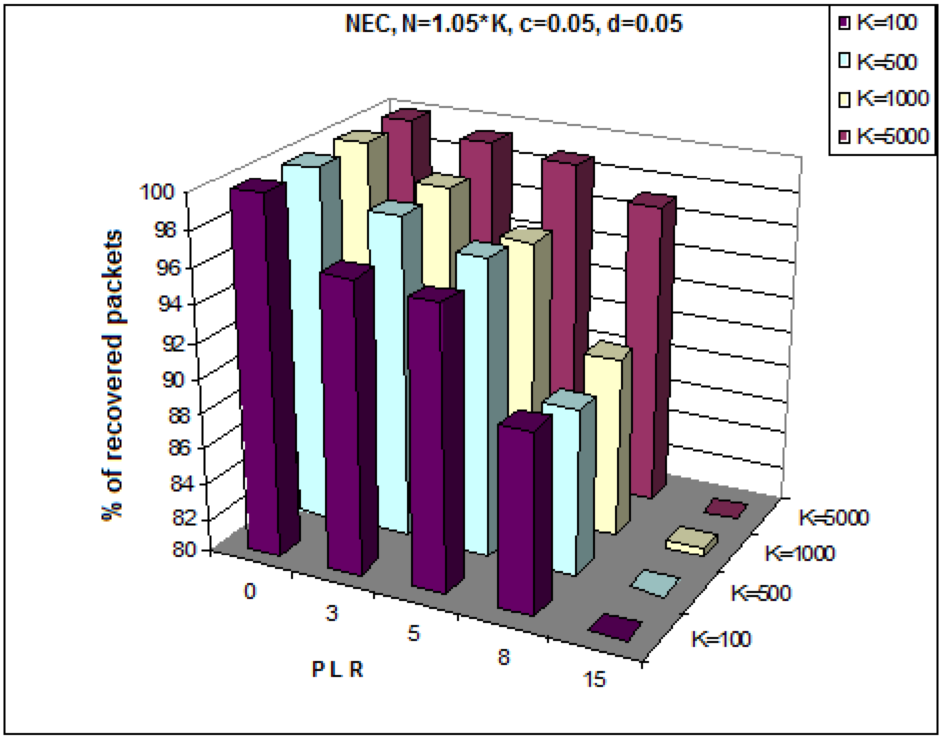

Fig. B.3. Percent of Successfully Recovered Packets for NEC.

K=100…5000, PLR=0…15, c=0.05, δ=0.05, $N = K \cdot 1.05$

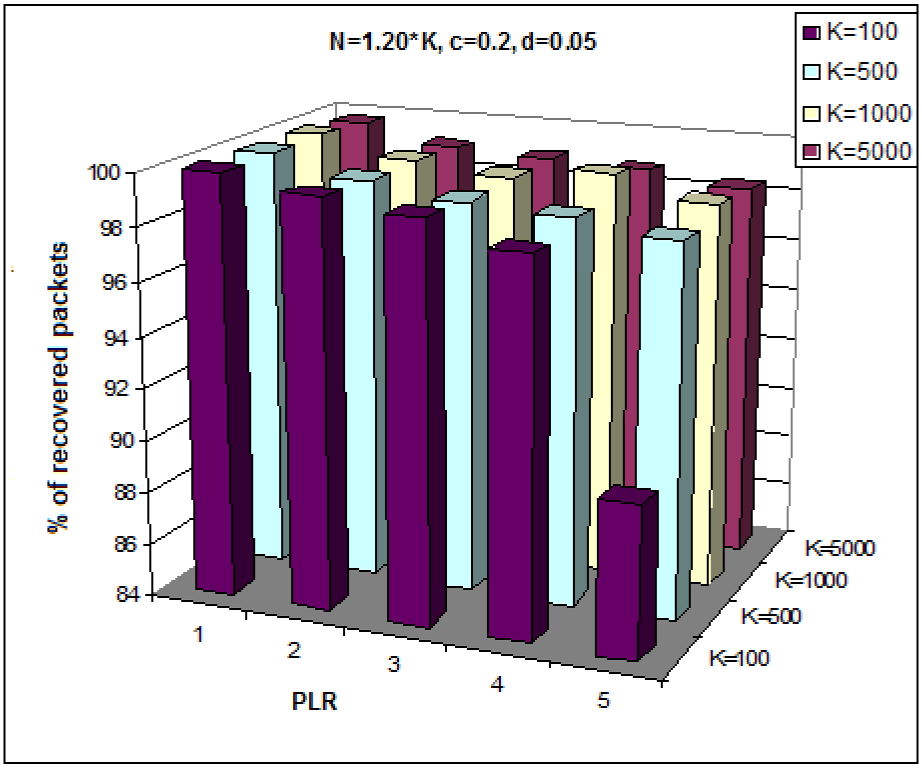

Fig. B.4. Percent of Successfully Recovered Packets for NEC.

K=100…5000, PLR=0…15, c=0.2, δ=0.05, $N = K \cdot 1.2$

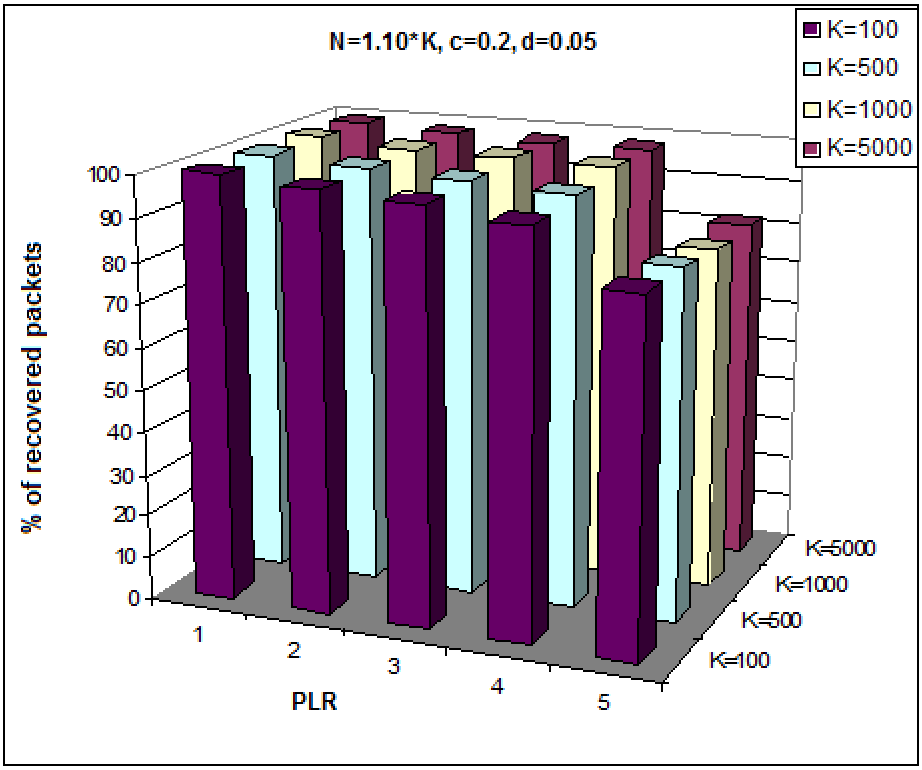

Fig. B.5. Percent of Successfully Recovered Packets for NEC.

K=100…5000, PLR=0…15, c=0.2, δ=0.05, $N = K \cdot 1.1$

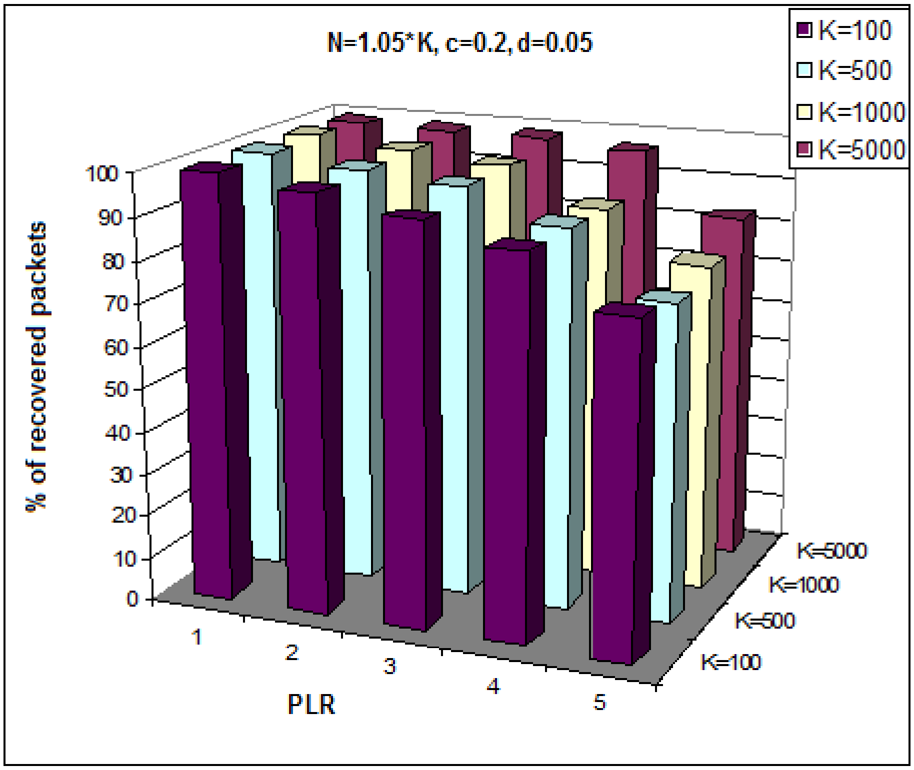

Fig. B.6. Percent of Successfully Recovered Packets for NEC.

K=100…5000, PLR=0…15, c=0.2, δ=0.05, $N = K \cdot 1.05$