MOBILE SPECIFIC ERROR CONTROL (MSEC): A PROPOSED ERROR CONTROL SCHEME FOR SUPPORTING QUALITY OF SERVICE IN WIRELESS ATM NETWORKS

by Panayiotis Toundas

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the School of Engineering Science

SIMON FRASER UNIVERSITY
April 1997

APPROVAL

Name: Panayiotis Toundas

Degree: Master of Applied Science

Title of thesis: Mobile Specific Error Control (MSEC):
A proposed error control scheme for supporting quality of service in wireless ATM networks.

Examining committee: Dr. Shawn Stapleton

Professor, Engineering Science, Chairman

Dr. R.H.S. Hardy

Professor, Engineering Science

Senior Supervisor

Dr. Paul Ho

Associate Professor, Engineering Science

Supervisor

Dr. Jacques Vaisey

Associate Professor, Engineering Science

Internal Examiner

Date Approved: _____

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

MOBILE SPECIFIC ERROR CONTROL (MSEC): A PROPOSED ERROR CONTROL SCHEME FOR SUPPORTING QUALITY OF SERVICE IN WIRELESS ATM NETWORKS

Author: _____
(signature)

_____
(name)

_____
(date)

ABSTRACT

Asynchronous Transfer Mode (ATM) is an important technology for the interconnection of heterogeneous data networks. The importance of ATM lies in its ability to support all kinds of data traffic seamlessly, and with minimum processing from node to node. Possible implementations of ATM can be realized with a variety of physical links, ranging from optical fiber, to radio links. However, the ATM layer (although independent of the choice of the physical link) assumes that at the physical layer the Bit Error Rate (BER) performance is exceptional. In most of the implementations today, the physical layer is the Synchronous Optical NETwork (SONET), which satisfies the BER performance requirements of the ATM layer.

Implementation of ATM in mobile data networks with wireless links is more challenging, and the subject of extensive research today. Some of the problems encountered are the hand-off treatment, mobile host registration, and the poor BER performance of the radio links. This research is dealing with the BER performance of the radio links. We investigate the problem of data packets corrupted with errors during transmission over radio links, and propose a method of correcting and/or detecting the errors and request retransmissions from the transmitter.

The proposed scheme is called Mobile Specific Error Control (MSEC), and its operating requirements is the maintenance of the Quality of Service (QoS) parameters set at the

setup phase of the connection(s). These parameters include the Bit Error Rate, Cell Loss Rate, and Maximum Delay. In addition, the MSEC scheme adapts to the changes of the quality of the radio links, utilizing the available bandwidth efficiently.

A detailed description of the mechanisms of the MSEC is given, and results derived from computer simulation of the scheme are presented and analyzed, to evaluate its performance in a variety of conditions.

To My Wife Dina

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. R.H.S. Hardy for his guidance and support during my research, and my fellow researchers Paraskeva Andrea Polydorou and Pamela Lee for their cooperation. In addition, I would like to thank the Natural Sciences and Engineering Research Council (NSERC), and Motorola of Canada Ltd. for their financial assistance during my research. Finally, I would like to thank Mil3 Inc. for providing access to the OPNET simulator.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Identification of the problem

This thesis presents my research in the area of mobile multimedia communications. The purpose of this research is the development of an error control scheme capable of handling errors in the wireless environment of the next generation Personal Communications Network (PCN). The term next generation PCN, refers to a mobile computing network capable of supporting multimedia services, in the same manner that today's wireless data networks, such as the Cellular Digital Packet Data (CDPD), are handling data services. The next generation PCN will likely consist of an Asynchronous Transfer Mode (ATM)-based backbone network, with wireless extensions at its ends. ATM is enormously successful in transporting multimedia traffic over wireline networks, and it is the transport technology of choice of the Broadband-Integrated Services Digital Network (B-ISDN). Many researchers view the transport of multimedia services to mobile users as an extension of the existing ISDN, thus ATM will

likely be used over wireless links. The extension of ATM to mobile users however, involves additional problems unique to the wireless environment. Such problems are registration of the mobile users, constrained bandwidth, hand-off treatment, and handling of errors introduced during the transport of data over the radio link(s). In this research we investigate the problem of errors in the wireless environment in more detail, and we are proposing an error control scheme capable of handling this problem.

1.2 Previous and current work on wireless ATM

Wireless ATM is a relatively new area of research, with pioneering work dating just a few years ago. As described in the previous section, the deployment of ATM over wireless links involves a number of problems stemming from the mobility of the end users. As of this writing, there is a number of papers in the area of wireless ATM. In this section we give an overview of the existing literature on the subject.

Raychaudhuri and Wilson in a hallmark paper [27] written in 1992, suggest a transport architecture for wireless ATM. Their work encompasses the structure of the Next Generation Personal Communications Network (NGPCN), provides a proposed Protocol Layering to support multimedia services, gives a proposed packet format to be used over wireless links, and describes in detail the underlying problems for implementation.

Acampora and Naghshineh in [1] propose a methodology for the treatment of handoffs in wireless ATM networks. Their work assumes the network architecture described by Raychaudhuri and Wilson in [27].

Ayanoglou et al [4] proposed a complete link layer protocol for wireless networks in 1995. The protocol handles all the problems associated with wireless data transmission. For error control, the protocol utilizes both Forward Error Correction and Automatic Repeat reQuest (ARQ) combined, to handle errors when the time constraints of the connection are stringent. We follow the same approach in the MSEC scheme.

Benneli et al in [6] propose an error recovery method for use in wireless ATM. The method uses segmentation of the ATM cell payload into submultiples of 48 bytes, and addition of error control bits, as a form of "radio packet".

In a recent paper, Bibb and McGregor [8], investigate in detail the needs of error control in wireless ATM networks. Again in this paper, the architecture of the network is the one suggested by Raychaudhuri and Wilson in [27]. The error recovery suggested uses both FEC and ARQ, along with interleaving for randomization of errors.

Lu and Brodersen in [18] also propose and error control utilizing both FEC and ARQ, with an adaptation feature, which selects the code used from a set of available codes.

In addition, Wilson et al in [31] compare dynamic TDMA versus packet CDMA in an effort to address the problem of the Medium Access control in wireless ATM networks, for the support of multimedia services.

As noted previously, most of the papers cited in the references are quite recent, demonstrating the activity in this field.

1.3 The design of the error control scheme

The design of the error control scheme takes under consideration the characteristics of the radio link, as well as the Quality of Service (QoS) requirements of the services to be supported, and the constrained bandwidth available in the wireless environment. With the term Quality of Service (QoS) we refer to a traffic contract, which describes the requirements of the connection in terms of performance parameters. The quality of service contract includes parameters such as maximum acceptable Bit Error Rate (BER), Cell Loss Rate (CLR), and maximum delay. The traffic contract is to be satisfied for the duration of the connection to ensure proper communication between the communicating parties. In the case of the wireline ATM networks, the QoS contracts are constructed at call setup time using the signaling protocol Q.2931 [10], or ATM Forum's User-Network Interface (UNI) specification [3]. According to this convention, the calling party issues a CALL SETUP message, which contains information about the connection to be established, including the QoS parameters. Upon establishment of the connection, the QoS parameters must be maintained.

The central purpose of the error control scheme is to ensure the maintenance of the QoS contract, when the ATM network has wireless links. However, such contracts cover a wide range of values, ranging from services with stringent time delay requirements and relatively high tolerances in cell loss rate, to time-tolerant services requiring very low cell loss rate. Thus, the error control must be able to accommodate a wide range of services and QoS contracts. One approach to solve this problem would be to construct a scheme powerful enough to handle the "worst case scenario" of conflicting QoS requirements.

Such an approach, would inevitably use a powerful error correcting code of low rate, and thus it would be inefficient in terms of bandwidth utilization.

In addition, radio channels exhibit variations over time, subject to fading, co-channel interference, weather conditions, and other parameters which may impair the quality of the received signal.

The error control scheme we are proposing (MSEC: Mobile Specific Error Control) addresses the above considerations by using different error correcting/detecting codes for services with different QoS contracts.

For purposes of easing the design, we first categorize the services to be supported in two types: Time-tolerant and time-sensitive. For the former we have the convenience of retransmitting corrupted packets, while for the latter strict error correction must be used. A large code suite is used including non-binary Reed-Solomon codes (with symbols from the 64-ary alphabet), and Bose-Chaudhuri-Hocquenghem (BCH) codes of length 54 (derived from the codes with length 63). The Reed-Solomon codes are used for error correction for time-sensitive services, while the BCH codes are used for error correction and detection for time-tolerant services. An important component of the design is the mechanism which allows the MSEC to switch to different codes within the available code suite, to suit best the requirements of the connection, and the present state of the channel. This mechanism relies on the monitoring of the radio link at the receiver side, in terms of the achieved QoS parameters, and their comparison to the parameters of the QoS contract. The outcome of the comparison may trigger the generation of a control packet at

the receiver, indicating that the transmitter should code packets with a more or less powerful code. The receiver then starts operating at a "hunting" state, looking for the first packet of the connection coded with the new code to arrive. This first packet has a particular bit in its header field set to 1 (last bit in the SN field, as will be shown), and this is the bit the receiver is looking for in the "hunting" state. Upon arrival, the receiver starts decoding packets with the new code.

The monitoring of the radio link is achieved with a reserved channel with the value of the VPI/VCI field of the packet header set to zero. Packets with this VCI/VPI value contain all zeros, so the receiving station only measures the Hamming weight of the incoming packet to determine the number of errors in the packet, and estimate the BER of the time interval between successive control packets. Subsequently, the receiving station determines the achieved performance parameters taking under consideration the correcting/detecting capabilities of the code currently used. The detailed description of this mechanism is described in chapter 3. The proposed scheme with its adaptive feature, makes efficient use of the available bandwidth, since it appends only as many parity bits as needed to the packets sent over the radio link, given the QoS contract of the connection, and the quality of the link at the time.

1.4 Thesis overview

The thesis work was divided in the development stage and the simulation stage. At the development stage, the properties of radio channels were investigated in terms of their error characteristics. Then a number of appropriate codes were selected for use in the MSEC scheme. In addition, all the mechanisms of the MSEC were defined, including packet formats and segmentation, the feedback mechanism, and switching conventions. At this stage, several guidelines were followed for the success of the constructed scheme as described in chapter 3.

At the simulation stage, the MSEC was simulated using the OPtimized Network Engineering Tools (OPNET) simulation package. All the models used for the simulations were built from the bottom up, in the hierarchical manner utilized in OPNET. The models used are described in detail in chapter 4. The performance parameters of interest in the simulations were the achieved bit error rate per connection, the achieved packet loss rate per connection, the number of retransmissions, the adaptation time of the scheme, and the types of codes used (indicating bandwidth utilization) at any given time.

The simulations were run for a varying error environment, and the results are presented and discussed in chapter 5.

Finally, chapter 6 contains conclusions on the performance of the MSEC scheme, and some suggestions for further work.

CHAPTER 2

BACKGROUND

2.1 Introduction

Since its introduction, ATM has been advocated as an important technology for the wide area interconnection of heterogeneous networks. Although the technology is still at its infancy, it is almost universally adopted as the scheme of choice for the transport of multimedia traffic of the Broadband Integrated Services Digital Network (B-ISDN), due to many attractive features that it offers. In summary, ATM offers high bandwidth, reliable transportation of data, support of connection-oriented and connectionless traffic, point-to-point and point-to-multipoint connectivity, and dynamic bandwidth allocation on demand with a fine degree of granularity. Due to the importance of the B-ISDN and ATM in the case of the next generation PCN, the fundamentals of them are presented in the following sections. In addition, in the section 2.5, we present the next generation PCN (NGPCN), describing its design and architecture, and describing the problems associated with the integration of the wireline ATM network. We primarily focus on the wireless

part of the network, since it is the operating environment of the error control scheme we are proposing.

2.2 The demand for multimedia services

Over the past decades, the communication needs of society have gone far beyond the plain telephone system. Technology advances in new communication networks, and market pull for more sophisticated means of communications resulted in a large number of communication systems, including telex, fax, data transfer over the telephone lines via modems, and after the success of the personal computer (PC), sophisticated computer networks. However, the gradual deployment of the modern communication networks, resulted in the development of several, self-contained networks each one of which serves only one service. Despite this fact, the idea of one unified network, capable of supporting a variety of services was slowly evolving. A classic example is the data transfer over the telephone lines via modem: An existing network, designed for voice communications, is used for data transfer with somewhat reduced throughput (28.8 Kbps in modern modems), a poor performance due to system limitations. In addition, the prospect of new services for commercial use such as videotelephony or video library for interactive TV distribution, fueled the research in the area of the Broad band Integrated Digital Network (B-ISDN).

2.3 The evolution towards the B-ISDN

If there is a word to describe today's communication systems is specialization, with each communications network optimized for the transport of a single service. An important consequence of this specialization is their inefficiency in terms of the resources offered as opposed to the resources used at a given time. More specifically, since a given network accommodates only one kind of service, bandwidth is allocated for the worst case traffic, and thus remains unused for the rest of the time. For instance, the peak hours in the telephone network are between 9:00am and 5:00pm, while the peak hours for CATV networks are during evening. Since resources are not shared, each network has to offer bandwidth according to the worst case traffic, which translates in inefficient use of resources.

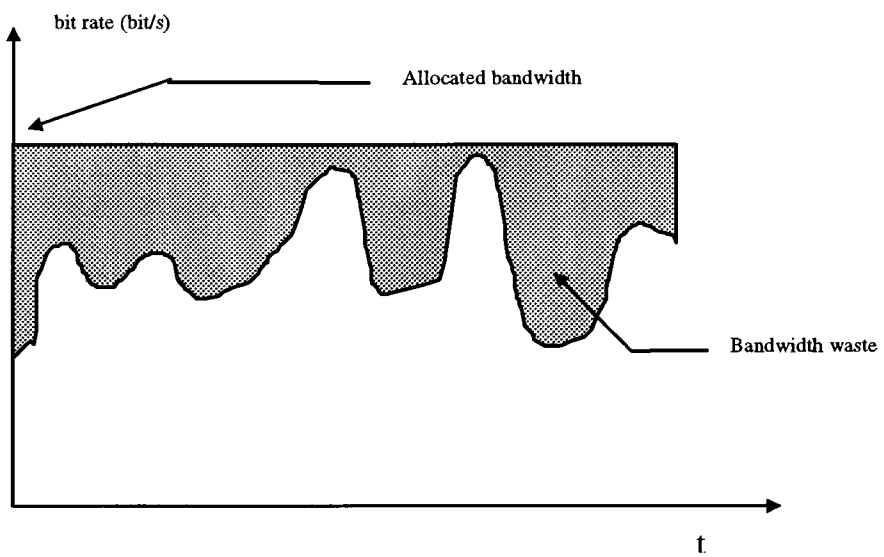

Figure 1: Peak rate bandwidth allocation.

Another handicap of the specialization of the communication networks is inflexibility to adapt to new service requirements, or the inability to accommodate new services with different bit rates. For example, consider the case where a network is designed to transport a particular service with a bit rate of 50 Mbps. If in the future the same service requires only 40 Mbps due to improved coding, the network switches and transmission systems are not directly suited for the new bit rate, so they would need adaptation to work efficiently. Eventually, the network will be able to carry the traffic, but with a large inefficiency: Only a fraction of the network resources will be used even at the worst case traffic. In short, we can identify the major shortcomings of today's networks with regards to their ability to transport multimedia traffic to be service dependence, inflexibility, and inefficiency in the allocation of resources. A single service-independent network such as the B-ISDN would address these issues, in order to be efficient for the transport of multimedia services. The B-ISDN would offer flexibility to adapt to newly introduced services, efficient resource allocation, and independence from the type of service being transported. Certainly, these characteristics of the B-ISDN indicate that it will be the universal communications network. However, the idea of the B-ISDN matured to this level because of several developments taking place at the same time in the communications field. For one, the development of the optical fiber as a physical medium provides a number of features suitable for the transport of multimedia traffic, such as large bandwidth, very good BER performance, and immunity to electromagnetic noise. On the other hand, the demand for real-time services at a high bit rate, made the implementation of X.25 based networks for this purpose unsuitable due to their high

complexity from node to node. For the transport of multimedia traffic, the focus of network design is on simplicity of switching and routing, and on connection-oriented operation for real-time services. These concepts are incorporated in the Asynchronous Transfer Mode networks.

2.4 The Asynchronous Transfer Mode (ATM)

ATM was developed specifically for use in the B-ISDN. The first ideas on ATM and related techniques were published in 1983 by two research centers (CNET, AT&T Bell labs). Since then, ATM underwent several developments, and finally was adopted as the scheme of choice for the transport of multimedia traffic.

2.4.1 The concept of ATM

The central idea around which ATM was build was the removal of many functions performed inside the network, in order to increase the information transport speed, while maintaining a high level of reliability and performance. In addition, it was desired that ATM would provide for flexible bandwidth allocation for a variety of services. We can summarize the solutions implemented in ATM as follows:

a) No error protection or flow control on a link by link basis.

Although the concept of ATM is independent of the choice of the physical medium, some of its design characteristics rely on the reliability of the link as in the case of error control. It is assumed that the link has exceptional BER performance, such that no form of error control is needed on a link-by-link basis. In addition, no flow control is performed at each node. To control queue overflow, ATM is relying on the correct resource allocation at call setup time, along with proper queue dimensioning at the

network nodes. The only preventive action employed in ATM networks is the error protection of the header of the packet (cell in ATM terminology) to prevent misrouting. However no error control is applied on the information field of the cell on a node to node basis. Error control is left to the upper protocols residing at the communicating terminals as shown in Figure 2.

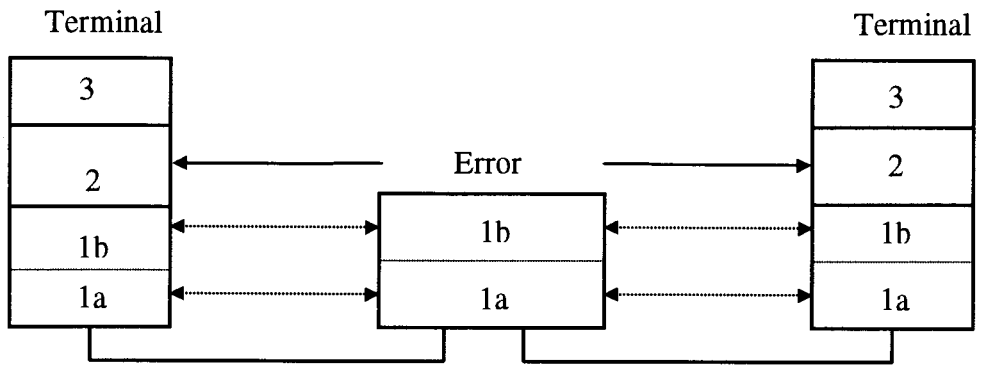

Figure 2: ATM Error Control.

b) ATM operates in a connection-oriented mode.

In contrast with connectionless packet switched information networks, ATM operates in a connection-oriented mode. At call setup time, the network establishes a logical virtual connection and reserves the necessary resources if these are available. A Virtual Connection Identifier (VCI) field in the header of the ATM cell identifies the virtual connection. The establishment of the VC guarantees a minimal packet loss ratio, since a fraction of the network resources are devoted to serve the particular connection. Typical probabilities of packet loss for fiber-based ATM systems range from 10^{-8} to 10^{-12}.

c) The header has limited functionality.

The main function of the cell header is VCI/VPI identification for routing of the cell. In addition, error checking is performed at the header to avoid misrouting and consequently multiplication of errors. In addition to these fundamental functions, the cell header supports a few other functions. The header may have two possible formats, one for the User-Network Interface (UNI), which supports a flow control function to the traffic directed to the network, and one for the Network-Node Interface (NNI) which does not support such a function. Formats of the ATM cell header are shown in Figure 3.

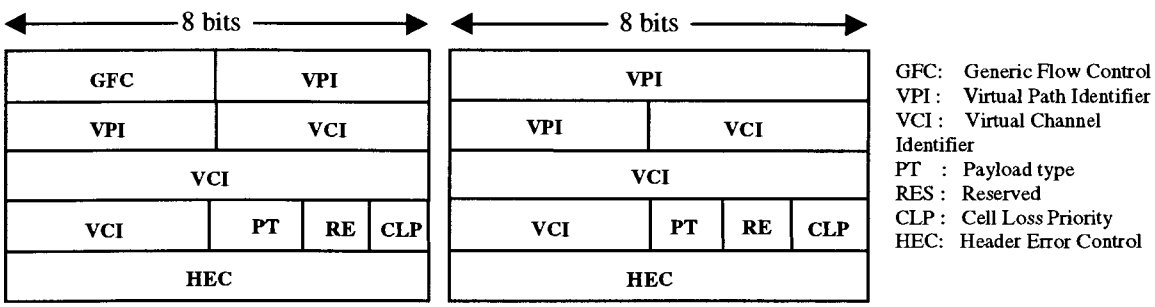

Figure 3: ATM header structure at UNI (Left), and at the NNI (Right).

d) The information field is small.

The ATM cell is 53 bytes long, of which 5 bytes is the header and 48 bytes is the information field. The size of the ATM cell was selected to be 53 bytes by CCITT in 1989 as the best compromise between a number of conflicting performance parameters [25]. The small cell size allows the use of small buffers in the network nodes, reduces the queuing delay, and allows the multiplexing of many traffic sources in a single outgoing

stream according to demand with a fine degree of granularity (Only 48 bytes of information at a time).

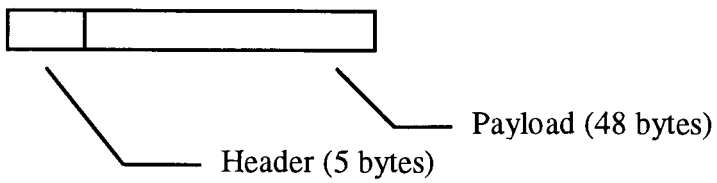

Figure 4: The ATM cell.

2.4.2 The overall ATM protocol stack.

As the concepts of ATM were realized, a large number of functions had to be grouped in functional entities. The functional grouping gave rise to the BISDN ATM protocol reference model illustrated in Figure 5. The model follows the OSI reference model, but as ATM is a transfer mode, only the lower layers of the communications hierarchy were specified, leaving the upper layers to be an independent choice. However, to conform better with different kinds of traffic, the ATM Adaptation Layer (AAL) has actually 5 classes of service with different frame formats for each one. The physical medium is also independent as we have seen already, with the optical fiber being the medium of choice. The BISDN ATM protocol model is divided in three planes as described in the CCITT Recommendation I.320 [13]: The user plane, to transport user information, the control plane for the transport of signaling information, and the management plane, which is split in two planes (Figure 5). These two planes are the

Layer Management, used to perform maintenance and operation functions for the network (Operation And Management functions, OAM), and the plane management, which has no layered architecture, and it is responsible for the management and coordination of the other three planes.

In summary, three layers constitute ATM: the physical layer which mainly transports information (cells), the ATM layer which performs switching/routing, and the ATM Adaptation Layer which adapts the information from the above layers to the ATM stream. Further to the protocol reference model, the layers are divided into sublayers with functions specified within them. A detailed picture of all the functions of ATM is shown in Figure 6 for the user plane.

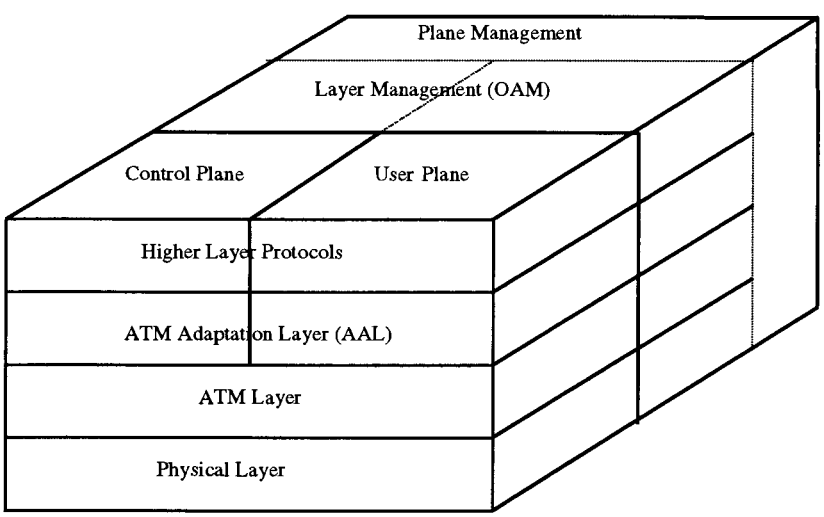

Figure 5: B-ISDN ATM Protocol Reference Model.

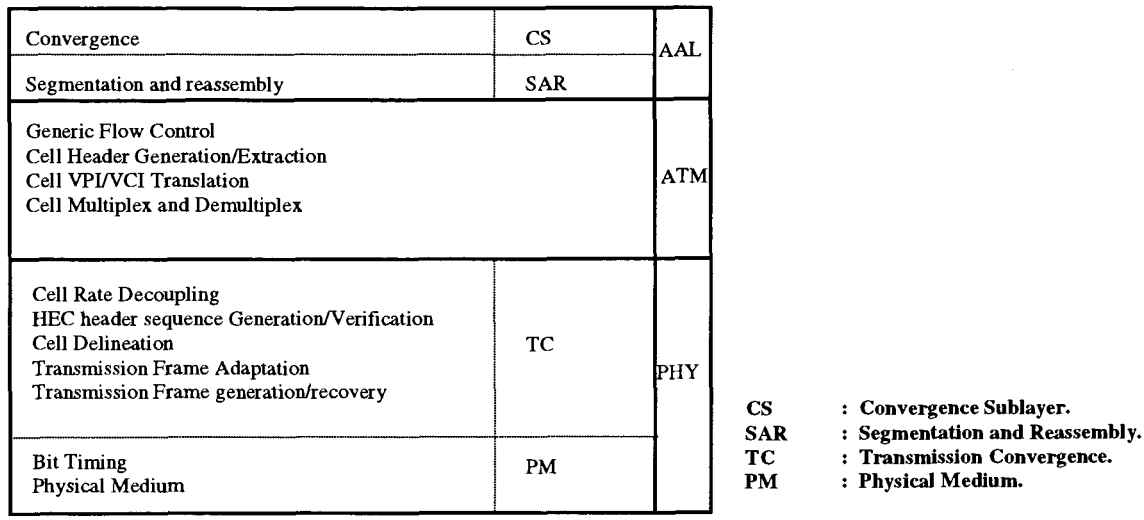

Figure 6: User Plane Sublayers and Functions.

Although some of these functions are self-explanatory, others are more complicated. The detailed explanation of the above functions is outside the scope of this introduction. More details can be found in [10], [12[, and [24].

2.4.3 Overview of the error control in the ATM case

In ATM, limited error control is performed on the header of the cell. More specifically, a HEC field protects the header of the ATM cell as shown in Figure 3 (For both the UNI and the NNI) with size 1 octet. This code has the ability to correct a single bit error on the header, or detect multiple errors. The underlying reason for protecting only the header of the cell is rather simple: If a single error occurs in the information field of the cell, the effect of it is a slight degradation of the quality of service offered to the

end user. However, when the error occurs in the header, the effect of it is the misrouting of the cell, and the error multiplies: The receiver of the cell actually receives 48 bytes of information in error (This is not the intended destination), and the intended receiver never receives the cell (Another 48 bytes in error). The scheme is adaptive (Figure 7), and it operates in the correction mode until an error (or multiple errors) is detected, when it switches to the detection mode. In the detection mode, if errors are detected, the cells are discarded and the misrouting is limited to one cell (the first cell with multiple errors). When the burst error is exhausted, the algorithm returns to the single error correction mode again.

The *(40, 32)* HEC code of the cell header is generated as follows:

The 4 octets of the header (before the addition of the parity check bits) are multiplied by the polynomial x^8, and then we obtain the remainder (modulo 2) of the resulting polynomial by the polynomial x^8+x^2+x+1 (generator polynomial). The 8-bit remainder is XORed with the pattern 01010101, and the result is inserted in the last octet of the header as the parity check bits. At the receiver, assuming we have received the cell with a header forming the polynomial R(x), the syndrome is calculated as R(x) mod G(x), where G(x) is the generator polynomial x^8+x^2+x+1.If the syndrome pattern is zero, there are no errors in the header, or the error pattern is such that it transformed the original codeword to another valid codeword (This happens with probability $(2^k-1)/2^n$ for a *(n, k)* code).

There are exactly 40 patterns with 1 bit error:

$$(R(x) \bmod G(x) = x^i \bmod G(x), \text{ with } i = 0,1,2,\ldots 39) \qquad (1)$$

all of which can be corrected in the correction mode.

The HEC code is also capable of detecting multiple bit error patterns in the detection mode, with a maximum of 8 bits only if they are consecutive (burst). The ITU-T and the ATM Forum's User-Network Interface (UNI) specification suggests that the use of the Error Detection Mode only, but it does not require it. Under error conditions, the HEC operates for most of the time in the error detection mode, thus discarding all the cells with erroneous headers, while the 48-byte information field is not protected. Figure 7 depicts graphically the error protection of the ATM cell header.

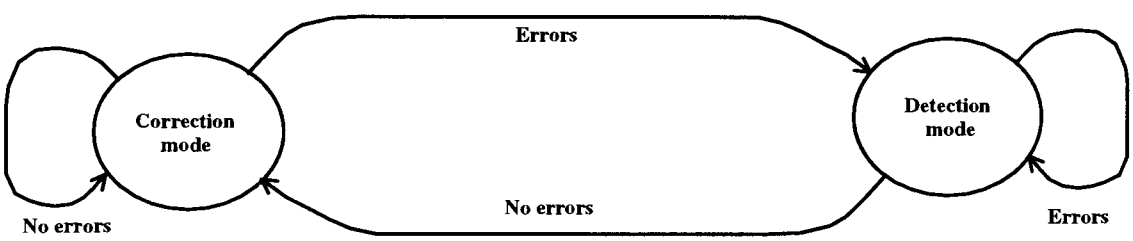

Figure 7: Error correction/detection algorithm in the ATM cell header.

In an optical fiber-based network the HEC protection of the cell header is fast and very effective: Experimental results conducted by ATT and Bellcore [15] suggest that (in an optical-fiber based system) the probability of having a single bit error within an errored second of data reception is 99.64% under normal conditions, with the remaining 0.34% shared between 2,3 or 4 bit errors. So the HEC protection of the header is able to correct errors in the header for the majority of the cases. However, when physical media with

lower quality error characteristics are considered, then an appropriate error control scheme becomes necessary.

2.5 The next generation PCN (NGPCN)

In the years to come, the world of personal communications will move in a new era which is characterized by the provision of broadband services to mobile users. This transition from the cellular telephony to broadband services, is the result of several developments taking place today. For one, the provision of broadband services in wireline networks is becoming reality, due to the development of the ATM technology. In addition, the enormous success of today's personal communication systems (over 20% growth annually) is fueling research aiming to offer new exciting services to this very active market. On the other hand, continuous research for the development of new coding techniques is expected to decrease dramatically the bandwidth required for the transmission of data and other services, and make their transmission over wireless channels possible. Already, there are wireless networks in operation that allow the transport of data over the existing cellular telephony networks, when voice channels become available. For example, the Cellular Digital Packet Data (CDPD) technology is already implemented in several cities in North America. The next step in personal mobile communications is the integration of data transfer and voice, and ultimately the provision of broadband services, supported by portable multimedia PCs. Although the provision of

broadband services in wired networks is realized by ATM, in the case of wireless networks we must take into account the characteristics of the radio link, and make appropriate modifications to ATM.

2.5.1 General description/Architecture

The next generation Personal Communication Network (PCN) will likely be based on an ATM wireline network with wireless extensions at its ends [1], [5], [26]. This architecture is the most likely evolution of today's wireless networks, given the deployment of wireline ATM for the transport of broadband services. Therefore, the next generation PCN (NGPCN) will likely consist of the ATM wireline transport network with fixed nodes extending up to the Base Stations (BS), and of the Mobile Stations (MS) linked to the network via radio links. Each BS is serving all the MSs inside a certain area around it (cell). The mobiles are by definition not constrained to a certain position and they can move from one cell to another. Apparently, the mobility of the end systems requires the support of additional functions such as hand-off and registration. In addition, the heterogeneous nature of the architecture introduces a "mismatch" in terms of performance and available resources with serious implications in the system design and communication protocol implementation.

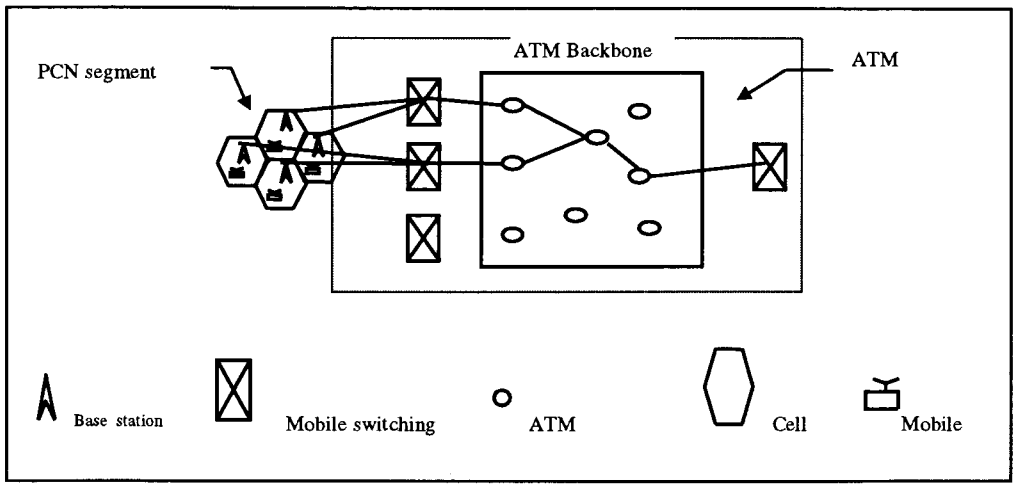

Figure 8: The next generation PCN.

2.5.2 Characteristics of the next generation PCN

a) Differences with cellular telephony networks

The NGPCN as already described demonstrates many similarities with the existing cellular telephony networks, such as the cellular structure, the concept of a backbone transport network, the utilization of base stations as points of attachment of the mobiles to the network, and the connection-oriented mode of operation. However, many underlying differences exist as well, caused by the nature of services supported, and the different backbone network. We can summarize the major differences as follows:

- Microcells (~ 0.5 km) or picocells (~ 100 m) as opposed to macrocells (~ 5 km).
- Use of a MAC (Medium Access Control)/channel sharing technique at the data link layer for the support of multimedia services.

- ATM compatibility in terms of the communication protocols used and data packet format in the case of the NGPCN.

All of the above features have as a common objective to achieve a certain degree of transparency for a subset of broadband/ATM services in the wireless environment. Apparently, due to the nature of the radio link and the optical fiber it is impossible to achieve a perfect match in terms of performance; a "qualitative" match of the two media is the realistic goal of the next generation PCN.

b) Characteristics of the radio link.

In cellular radio systems, the quality of transmission is a function of many parameters that may change over time causing unstable and unpredictable performance. More specifically, radio channels are sensitive to fading, electromagnetic interference, weather conditions, atmospheric composition, and geographic anomalies. Since these parameters are constantly changing as the mobile user is moving, the characteristics of the channel are also changing, causing the transmitted signals to be more or less corrupted at the receiver. Typically, the bit error rate (BER) of a cellular radio channel ranges from 10^{-3} to 10^{-2}, which is a poor performance according to the information networks standards. In addition, the errors occurring in radio channels are not independent, but bursty; many errors may occur in sequence or very close to each other, like the system temporarily entered a "bad state" of high BER. The characteristics of the

radio channels impose certain limitations to the implementation of ATM, since ATM (although independent of the choice of the physical medium) assumes excellent BER.

2.5.2 Functional problems in the NGPCN.

The architecture of the NGPCN as described in the previous sections introduces a number of problems for its implementation. We can briefly describe the most prominent of them to get a good grasp of the designing challenges of wireless multimedia networks.

a) The "bottleneck effect".

By this term we want to describe the mismatch in operating speeds of the ATM backbone and the wireless network(s) at its end. The wireline ATM network is capable of running at hundreds of Mbit/s, provided that the medium is optical fiber, while wireless networks usually operate at lower speeds. For example, wireless LANs usually operate at 2-3 Mbits/s. This performance difference will be obvious at the point that the two networks interface. The interface point (internetworking point) can be either the base station (BS), or the mobile switching center (MSC). A potentially dangerous situation may arise when the ATM backbone network sends packets (ATM cells) to a given MSC or BS, at a rate higher than the wireless part of the network can support. In this case, we have queue overflow at the interface point, and cells are lost at a disastrous rate. This situation is intensified by the fact that cell retransmissions are taking place at the wireless part of the network, requiring that a portion of the queue is reserved for unacknowledged packets.

b) Additional processing overhead at the internetworking point.

At the point that the two networks interface, a header "translation" must take place, in order for different functions specific to the wireless network (or the ATM network) to be supported. In the direction towards a mobile, the header must contain fields to support hand-off control, a different error control scheme, and a Medium Access Control scheme (MAC). These functions are not needed when the cells are transversing the ATM network. Instead, a Virtual Channel Identifier field (VCI) and a Virtual Path Identifier field (VPI) are used. The requirement of this "translation" is adding processing time at the interface point of the two networks, in contradiction to the demand for minimum processing at that point. The header translation classifies the system that performs it as a gateway; however this translation is kept to a minimum, making the two interfacing networks as compatible as possible. Raychaudhuri and Wilson in [27] are proposing a specific form of the PCN data-link packet format, made possible by compressing the ATM header and adding a PCN-specific header. The result is a 54-byte ATM-compatible frame.

c) BER incompatibility.

Aside the differences in operating speeds, the ATM and the wireless network have differences with respect to the Bit Error Rates (BER) of their links. More specifically, the optical fiber has a bit error rate of about 10^{-10}, when the airlink of about 10^{-3} or even worse. The difference in bit error rate is one of the key issues for the design of an effective error control scheme. Looking at the network as a whole, one realizes that the

wireless portion of the network, is more prone to error than the wireline part of it. An efficient way to compensate for the shortcomings of the radio links of the wireless network is to utilize the resources of the ATM network as much as possible. This technique is widely accepted and it will be used extensively throughout the construction of the next generation PCN.

d) Mobility of the end systems / hand-off problem.

The mobility of the end systems creates the need for redirection of data, every time the mobile moves to another cell and it is being serviced by another base station. This problem, referred to as the hand-off problem, has been treated successfully in the case of the cellular telephony. However, the case of the next generation PCN is somewhat different. At first, the NGPCN will more likely operate in a microcell or picocell environment, as opposed to the macrocell environment used for cellular telephony, and therefore hand-offs will occur more frequently, requiring fast and seamless service more often. In addition, the next generation PCN will support connectionless services as well, requiring buffering of unacknowledged packets at the transmitter. A problem associated with this form of services, is that unacknowledged packets residing in the currently serving base station, must be passed to the new serving base station immediately after a hand-off occurs (for the event of possible retransmissions). Thus, the hand-off treatment in this case, aside the renegotiation of the connection must also deal with the transfer of the unacknowledged packets to the new serving base station. If we also take under consideration the classification of the unacknowledged packets in different service

classes, we can understand the complexity of the hand-off problem in the NGPCN case. Acampora and Naghshineh in [1] propose a methodology for handoff in wireless ATM networks. Clearly, the design and implementation of the NGPCN is a challenging issue, with many problems still pending. This thesis presents research work in the area of error control in the wireless part of the network.

CHAPTER 3

THE MOBILE-SPECIFIC ERROR CONTROL SCHEME (MSEC)

3.1 The construction of the MSEC scheme

For this part, a careful consideration of all the characteristics of the next generation PCN leads to the requirements that the MSEC must fulfill in order to be efficient for use in the mobile multimedia environment:

- Ability to handle services with diverse QoS requirements.
- Efficiency under (relatively) high data bit rates.
- Simplicity in its implementation, for BS and mobile system considerations.
- Possession of a large error detecting/correcting code suite.
- Adaptability, to adapt to the changes of the radio link.

Many of the above requirements are contradicting; thus the design of the error control should be the best compromise between them. Key issues of interest in the design are also the selection of the error recovery mechanisms, the format of the packets for transmission over the radio link, and the adaptive mechanism, which allows the switching to another code with different error handling capabilities if the quality of the link improves or deteriorates. In addition, the design of the MSEC must be (to the best possible extend) compatible with ATM principles. For this reason, in the MSEC we have fixed length packets over the radio link, formed from the ATM cells with some limited segmentation. Furthermore, both the BS and the mobile are running the ATM protocol suite, with few (as few as possible) modifications to provide for the MSEC scheme. The schemes of choice for the MSEC are FEC for the real time services, and hybrid ARQ for the rest of the services. These two schemes can be realized with a variety of codes; in our case, we use Reed-Solomon codes from a 64-ary alphabet (6-bit symbols) for FEC, while in the case of non real-time services we use codes derived from the 63-bit binary BCH codes for reasons explained in section 3.2. The adaptive mechanism of the MSEC will be realized as a feedback mechanism that monitors the link quality and sends a control message to the transmitter. Upon reception of the control message, the transmitter determines the packet loss rate and makes the decision whether or not to switch to a different error control code. The transmitter notifies the receiver by setting the last bit in the SN field of the cell header to 1, which indicates that this packet is the first one encoded with the new code.

3.2 The selection of the codes

The process of the code selection is threefold: First, we examine the characteristics of the medium, with respect to the BER observed and the burstiness of the errors; second, we look at the services to be supported in the wireless environment and their expected QoS, and third we investigate the available codes and their suitability for the particular services. In the previous sections, we investigated the properties of the radio link, and we concluded to the BER of 10^{-3} or even worse, with bursty characteristics. In the next generation PCN, we expect the provision of multimedia services as in the case of ATM, with the difference that the expected QoS will be somehow reduced in comparison with fiber-based systems. In addition to these requirements, we must consider the efficient use of bandwidth, using only as many error control bits in the packets as necessary. Given the fact that the length of the radio channel packet should be constant to ease the packet delineation function and the synchronization between transmitter and receiver, the solution is to keep the length of the packet constant, and change the length of the *information field inside it*, allowing the accommodation of more or less parity check bits, yielding a more or less powerful code. This topic is discussed in one of the following sections.

In table 1 we have summarized a few example services along with anticipated QoS contracts [18]. In this table, the source relative burstiness is a measure of the fluctuation in the natural bit rate generated by the source of the information.

Data type	Bit rate (kbits/s)	BER	Relative burstiness	Delay	Jitter
Control	~1	$\sim 10^{-8}$	1-10	Low	low
Text	1-1000	10^{-4}-10^{-7}	1-10	High	high
Audio	64	10^{-4}-10^{-7}	2	Very low	very low
Video	1000	10^{-4}-10^{-7}	2-3	Very low	very low

Table 1: Example services with QoS requirements.

We can describe the natural bit rate generated by the source as a stochastic process *s(t)*, which lasts for a duration of *T* seconds. From this process we can obtain two important parameters: The peak natural bit rate *S*, where $S = \max[s(t)]$, and the average natural bit rate $E[s(t)]$, where

$$E[s(t)] = \frac{1}{T}\int s(t)dt \qquad (2)$$

The source relative burstiness is defined as:

$$B = \frac{S}{E[s(t)]} \qquad (3)$$

and it is an important parameter in the design of a multimedia-capable network. From the above table becomes apparent that in the case of real-time services where no retransmissions occur, the codes selected must be designed to handle burst errors. Many codes have been developed for this purpose, such as the Fire, or the Burton codes [16].

However, an important class of non-binary BCH codes, the Reed-Solomon codes, have excellent properties that make them suitable for burst error correction. Mainly, the ability of Reed-Solomon codes lies in the fact that they correct symbols of a particular length, regardless of the number of bit errors in them. Thus, they are capable of correcting a number of error bursts. In addition, Reed-Solomon codes have the property that they are *maximum-distance-separable codes.* With this term we refer to the (n, k) codes that have a minimum distance $d_{min} = n-k+1$, which is the largest value that d_{min} can have [16]. The importance of d_{min} is that it determines the number of correctable symbols t in a code of length n symbols encoded with k parity-check symbols. Typically, $d_{min} = 2t+1$, so the largest the value of d_{min}, the higher the correcting ability of the code. Although the minimum distance of a (n, k) can be as high as $n-k+1$, few codes have this desirable property. A family of Reed-Solomon codes based on a 64-ary alphabet (6 bit symbols) will be the basis for the FEC of our scheme. In addition, for the hybrid ARQ scheme, we will use short length binary BCH codes for error correction/detection. Short codes are more immune to burst errors than long codes, since the bit stream is encoded in much smaller blocks [24]. Short BCH codes are also more efficient than long ones with respect to the ratio of detectable (and/or correctable) bits to parity check bits. This property makes them more suitable in channels with constrained bandwidth. A graphical illustration of the above property can be seen in Figure 9, where we have plotted the ratio $\frac{d_{min}}{n}$ as a function of the code rate $R_c = \frac{k}{n}$ for the BCH codes of length 31 and 63, along with some known upper and lower limits of the normalized minimum distance $\frac{d_{min}}{n}$. In

particular, the lower bound on the normalized minimum distance $\frac{d_{min}}{n}$ for (n, k) block codes developed by Gilbert (1952) and Varshamov (1959) applies to all binary and non-binary codes, and it satisfies the

inequality

$$\frac{d_{min}}{n} \geq \alpha \qquad (4)$$

and the code rate Rc to be:

$$R_c = 1 + \alpha \log_2 \alpha + (1-\alpha) \log_2 (1-\alpha) \qquad (5)$$

where

$$0 \leq \alpha \leq \frac{1}{2} \qquad (6)$$

In addition to this lower bound, we plotted two upper bounds developed by Plotkin (1960), and by Elias (1968). These two upper bounds were developed to give a tighter bound on the minimum distance than the one mentioned before ($d_{min} = n-k+1$) which is very loose for binary codes (In fact, there isn't a binary (n, k) code with $d_{min} = n-k+1$). The Plotkin upper bound predicts that:

$$\frac{d_{min}}{n}\left(1-\frac{1}{2d_{min}}\log_2 d_{min}\right) \leq \frac{1}{2}\left(1-Rc+\frac{2}{n}\right) \qquad (7)$$

which in the case of large code length n ($n \rightarrow \infty$) simplifies to:

$$\frac{d_{\min}}{n} \leq \frac{1}{2}(1 - Rc) \tag{8}$$

Finally, the Elias upper bound predicts that for a (n, k) code we have:

$$\frac{d_{\min}}{n} \leq 2A(1\text{-}A) \tag{9}$$

where the parameter A is related to the code rate R_c with the equation

$$R_c = 1 + A\log_2 A + (1\text{-}A)\log_2(1\text{-}A) \tag{10}$$

with

$$0 \leq A \leq \frac{1}{2} \tag{11}$$

From Figure 9 we can clearly see that n=31 bit BCH codes have always a better performance than the n=63 bit BCH codes, in respect to the normalized minimum distance. Longer BCH codes have worse performance which falls close to the Gilbert-Varshamov lower bound. However, in our case the selection n=31 bits is not optimal since this short length limits the size of the code suite we can develop for the adaptive MSEC scheme. The selection n=63 is the best compromise between the conflicting requirements of versatility and efficiency for the hybrid ARQ scheme to be developed.

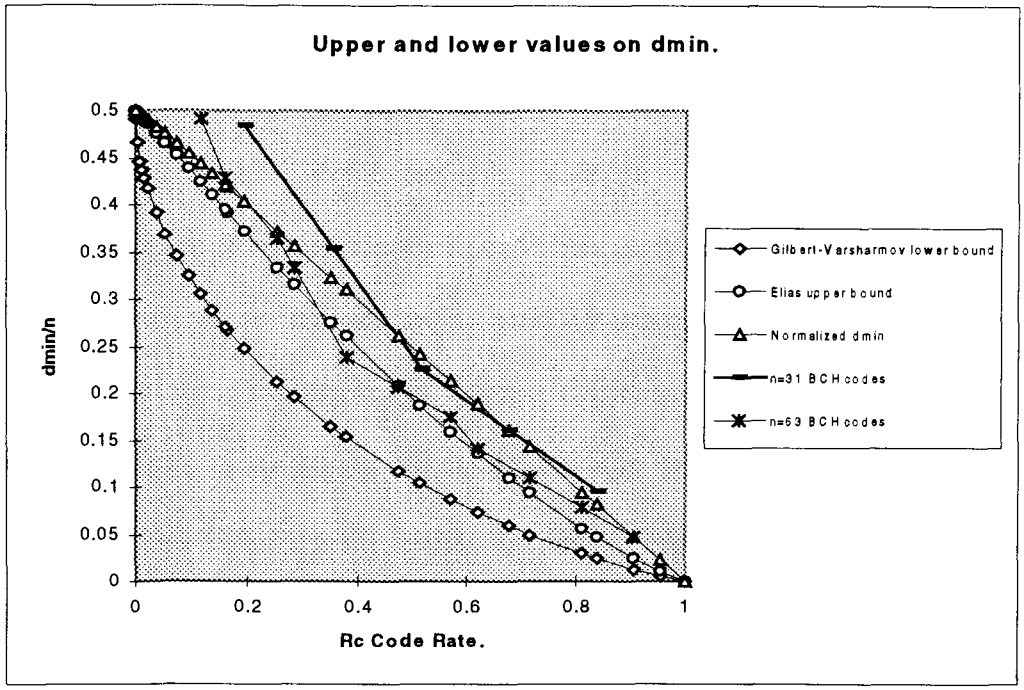

Figure 9: Performance of binary BCH codes.

In the MSEC scheme we want to have an adaptive scheme, capable of accommodating services with diverse QoSs. For this reason, we employ a family of BCH codes with different error correcting and error detecting capabilities in the case of non-real time services, and a family of Reed-Solomon codes for the case of real time services.

a) The protection of the header of the packet

In the MSEC scheme, the VCI field of the packet plays a vital role in the implementation of a variable and adaptive error control scheme. The VCI field is examined by the receiver first, in order to determine the error control scheme to be employed for the detection and/or correction of the incoming packet. An erroneous assessment of the header of the packet multiplies the error, because the packet potentially will be corrected

with a different code than it was coded with, and delivered to a service other than the one it belongs. For this reason, we provide each packet (regardless of the VC it belongs) with a header of 30 bits, which includes fields for VCI, hand-off control, and sequence number. Due to the importance of the VCI field, we employ the extended (24, 12) Golay code for error correction on the VCI field upon generation. The (24, 12) Golay code is a binary linear code with a large minimum distance $d_{min}=8$. The functions the VCI and Golay fields are two: First is the identification of the incoming packets, and second is the packet delineation, which guarantees the synchronization between transmitter and receiver. The Golay code we use has rich algebraic structure, and several practical ways to decode it [16]. The generation polynomial of the Golay code is:

$$g_1(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}, \qquad (12)$$

or,

$$g_2(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11} \qquad (13)$$

In addition, the correct decoding of the header is eased by the fact that the acceptable codewords is limited to few (as many as the number of the active connections), so after the decoding, the resulting codeword is compared to the set of the VCI's of the active connections. In the case that the decoded codeword does not match one of them, the VCI with the smallest Hamming distance from the codeword is selected. An efficient VCI allocation algorithm could be used at the base station, so that the Hamming distance between VCI's allocated for connections to a particular mobile host to be maximum. The

selection of a powerful code such as the extended Golay code is essential for the proper delivery of the packets.

b) The FEC scheme

For the FEC schemes we use a family of Reed-Solomon codes derived from the Galois field GF(2^6) with 6-bit words. The resulting code has length 378 bits, which is a short enough length for the characteristics of the link (transition probability is proportional to the packet length). Since the ATM cell is larger than 378 bits, we need a limited segmentation. The ATM cell for transmission over the wireless link, and the proposed segmentation is presented below:

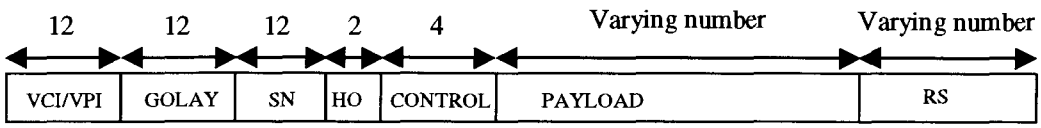

Figure 10: Air packet format (Data link).

Note that the VCI/VPI field (12 bits) is protected by the extended Golay field (12 bits), while the rest of the fields with the Reed-Solomon code currently used.

The resulting packet has length 402 bits (constant length for both kinds of services supported), or otherwise 67 6-bit words.

From the size of this packet, only 378 bits, or 63 6-bit symbols must be encoded, since the VCI field is already encoded with the (24, 12) Golay code. Keeping this in mind, we

describe the selected family of Reed-Solomon codes in the next table. All the sizes in the table are 6-bit words.

For a given T (desired number of correctable 6-bit words), the code has the following parameters:

LENGTH	T	PARITY CHECK	INFORMATION WORDS	ATM PAYLOAD	MINIMUM DISTANCE
63	1	2	61	58	3
63	2	4	59	56	5
63	3	6	57	54	7
63	4	8	55	52	9
63	5	10	53	50	11
63	6	12	51	48	13
63	7	14	49	46	15
63	8	16	47	44	17

Table 2: The Reed Solomon codes used in the MSEC.

From the above table, we realize that in order to employ these Reed-Solomon codes, we must segment our 67-word packet down to 61, 59,...,47-word packets. This is possible, if we provide MSEC with the ability of segmenting the ATM packets in blocks of 6 bits, and taking the appropriate number of these blocks, to form the information part of the packet to be transmitted, as illustrated in the above table. Note that for reasons of uniformity, the FEC scheme has been selected such that the resulting packet has always the same length of 63 symbols (plus the 24 bit VCI + Golay field) for a total of 402 bits for easier processing at the receiver's side, although we have adaptive error-correcting capabilities. We can also see that the resulting FEC is capable of correcting from 1 to 8

erroneous symbols anywhere in the 63 symbol packet. Also note that the FEC constructed this way is easily expandable to more powerful error correction levels, by taking smaller information fields, 12 bits at the time (2 6-bit words), with a trade-off in bandwidth utilization.

c) The hybrid-ARQ scheme

For the non-real time services supported from the future generation PCN, the error control scheme is selected to be adaptive hybrid-ARQ. In this section, we will describe the construction of the hybrid-ARQ scheme, and explain how it can accommodate a wide variety of services with diverse QoSs. The discussion of how the scheme can be adaptive will be described in section 3.4.3.

The hybrid-ARQ scheme is based on a family of binary BCH codes of length 63 bits, and correcting capabilities of 1, 2, and 3 bits. The choice of the length 63 was selected for its efficiency in terms of the minimum distance as a function of the parity check bits. Binary BCH codes are in general efficient, but as the length of the code increases, its efficiency decreases (See Figure 9). Therefore, shorter BCH codes provide a larger d_{min} (proportionally) than the longer ones. The (31, x) BCH code is more efficient, but it does not give us a large variety of codes to select from. In our case, the choice of codes is based on the binary BCH codes of length 63 shortened to 54 bits. The reason for this selection is the uniformity with the case of real time traffic case. More specifically, we want the final packets to be sent over the wireless link to be of equal size, regardless of the service they belong to.

Thus, since the packet in the real time case is 402 bits ((63x6) + 24) then the length of the non real time case frame must be 402 bits as well. In addition, this length must be the result of a number of 6 bit words from the ATM payload and the 18 bit field from the header of the packet (SN, HO, ATM CONTROL, Figure 10), all encoded in blocks of 63 (54 is our selection) bits. Unfortunately, if we select 63 bit encoding, the information field of the packet must have odd number of bits, which is not feasible to construct if we have the ability of segmenting the ATM payload in 6 bit words. In addition, the size of the code must be a submultiple of 378 bits in order to get an overall packet length of 402 bits. A good choice for the binary codes to be employed is a shortened version of the 63 bit BCH codes with length 54 bits. With this length, we can get the desirable length of 378 bits as a group of 7 code words (54x7=378), and also the information field is always a multiple of 6, so the code is compatible with the segmentation requirements. The family of the employed BCH codes is summarized in the next table.

LENGTH	MINIMUM DISTANCE	PARITY CHECK BITS	CORRECTABLE	DETECTABLE
54	3	6	1	1
54	5	12	2	2
54	7	18	2	4
54	9	24	3	5

Table 3: The BCH codes used in the MSEC scheme.

Important in this case is that the shortening of the BCH binary codes results in non-cyclic codes, but with the property that they have at least the same error correction/detection

capabilities with the original BCH codes. In addition, shortened codes are realized with the same circuitry with the original codes, and thus no implementation complications are involved. As in the real-time case, the construction of the codes is such that they can be easily extended to more powerful ones with the addition of new members in the "family" of codes.

Finally it is worth mentioning that only one of these codes is used at a given time. The transition from one code in the "family" of code to another (FEC or ARQ) is decided upon monitoring the link quality for some time. A degrading channel (when we receive many erroneous packets) triggers the transition to a more powerful code, when a channel with good characteristics triggers the transition to a less powerful code (to conserve bandwidth). The mechanics of the transitions are given in section 3.4.3.

3.3 Justification of the codes used in the MSEC

In section 3.2 we described the codes used in the MSEC scheme, along with the rational for their selection. In this section we revisit the properties of the selected codes that make them suitable for use in the MSEC.

3.3.1 The (24, 12) Golay code

The (24, 12) extended Golay code is used for the error protection of the VCI/VPI field of the packet. As mentioned before, the VCI/VPI field of the header has two important functions: First it is used for the identification of the connection that the packet belongs to, in the same fashion with wireline ATM. Second, it is used for *packet delineation*, which is the correct recovery of the packet boundaries in the bitstream, again in the same fashion the HEC field in the ATM cell is used. Thus we need an efficient code for its protection. The extended Golay 924, 12) code has the following attributes:

a) It has an overall length compatible with our segmentation requirements. Indeed, 24 is a multiple of 6 bits, which is the "unit" of our segmentation.

b) It has large minimum distance ($d_{min} = 8$). Among the codes with comparable size the Golay code has a large minimum distance, a feature that eases the recovery of the correct codeword during decoding. In addition, the Golay code is the only known perfect code that can correct all the error patterns with up to three bit errors in a block of 24 bits [16].

c) There is a number of efficient decoding techniques for decoding it, such as the Kasami Decoder, and the Systematic Search Decoder [16].

d) It has well known structure and has been used in communication systems widely.

3.3.1 The (63, x) Reed-Solomon codes

The Reed Solomon codes are used in the MSEC for Forward Error Correction in the case of time-sensitive services. Reed Solomon codes have a number of attractive features that make them suitable for use in the wireless environment, such as:

a) R-S codes are burst-error correcting codes. In particular, in our case the R-S codes are 64-ary codes, with symbols selected from the Galois field GF(64). Symbols in these codes have length 6 bits. The R-S codes used are capable of correcting a certain number of symbols in the codeword, regardless of the number of *bit errors* in the *symbol.* Thus, R-S codes are capable of handling channels with burst-error characteristics.

b) R-S codes are *maximum-distance separable codes.* A *(n, k)* code is maximum-distance separable when it has minimum distance $d_{min} = n-k+1$, where n is the length of the code, and k is the length of the information field inside it. The minimum distance can be as high as $n-k+1$, but very few known codes have this desirable property, with R-S the most popular.

c) R-S codes have been used widely for FEC in various wireless systems, such as CDPD.

3.3.2 The (54, x) BCH codes

The (54, x) BCH codes are used in the MSEC for error detection and/or correction in the time-tolerant services. BCH codes are suitable for this purpose, due to the following features they have:

a) They can be configured for error correction and detection, or only for detection in different implementations.
b) They have a good performance with respect to normalized minimum distance in comparison to longer BCH codes.
c) They are random-error detecting/correcting codes, and in combination with interleaving they can be effective in a burst-error environment, where retransmissions are possible.
d) They have well-known structure, and there are many effective algorithms to decode them, such as the Berlekamp's iterative algorithm, and Chien's search algorithm.
e) They are related to the R-S codes (R-S codes are non-binary BCH codes).

3.4 A specification of the MSEC scheme

In the following section, we demonstrate the operation of the mobile and the BS in the two possible directions: As a transmitter, and as a receiver. In particular, we describe the operation of the mobile station, with respect to the segmentation, packet generation from the header of the ATM cell, and the mechanics of the feedback mechanism which determines the transition between different codes. We assume that operation happens for two services, one time-tolerant, the other one time-sensitive.

3.4.1 The mobile as a transmitter

1) The mobile sends packets until the transmit window is exhausted (calculated as a multiple of the round trip delay), a retransmission request is received, or it has no more packets to send. Retransmissions have always precedence over new packet transmissions.

2) The BS receives the newly arriving packets from the mobile, and after determining the type of service associated with each packet (from the VCI value of the header), starts the error detection procedure. If packets have been received erroneously, they can either be corrected using FEC (In such case the BS corrects the erroneous bit(s) and processes the packet), or they must be retransmitted from the mobile. The BS requests the packet

immediately. If packets are received correctly and in order, the BS does not send acknowledgments (ACKs) for each one of them, but one ACK when the number of the received packets is about 50% of the transmit window size. This operation ensures the efficient use of bandwidth. Also, it is important to say that in this direction the retransmissions are persistent in the sense that they will take place until the packet is delivered correctly to the BS, or some negligible error rate has been achieved in comparison to the promised QoS.

3) The mobile receives an ACK. The ACK acknowledges the packet whose sequence number appears in it, and all the packets before it. Thus, the mobile discards these packets from its buffer, making room for new packets. In this case the mobile continues to send new packets to the BS.

4) The mobile receives a NAK. The effect of a NAK to the mobile is twofold:
First, it requests a retransmission of the packet whose SN appears in the NAK, and second it acknowledges all the packets transmitted before it. The action taken by the mobile in this case is as follows: First the mobile discards all the packets successfully received by the BS, up to the erroneously received packet. Second, the mobile stops transmission of new packets, and retransmits the erroneous one. MSEC uses selective repeat ARQ. Finally, the mobile continues the transmission of new packets, until another interrupt arrives, or the transmit window is exhausted, or it has no more packets to send. Figures 11 and 12 depict the functionality of the mobile as a transmitter.

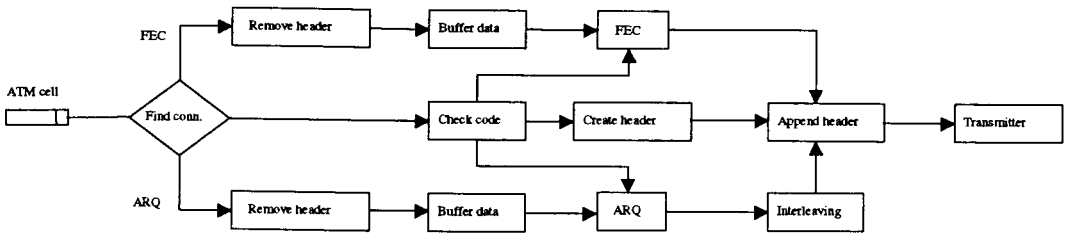

Figure 11: The mobile as a transmitter (Packet formation).

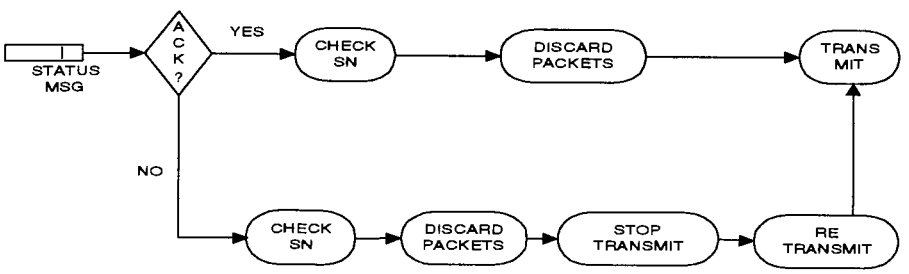

Figure 12: The mobile as a transmitter (Ack/Nak handling).

3.4.2 The mobile as a receiver

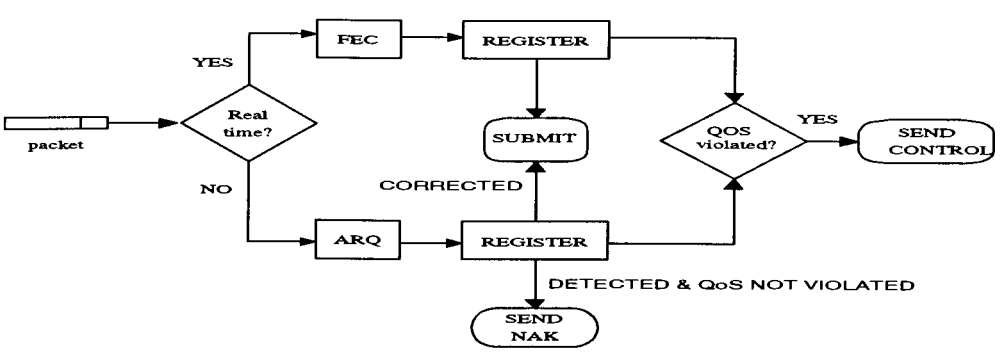

Figure 13: The mobile as a receiver.

In the opposite direction, the mobile is the receiver. The functions as a receiver are as follows:

1) The mobile receives a new packet. First, the mobile determines whether or not the arrived packet belongs to a real time application or not. The distinction lies in the value of the VCI/VPI field of the packet, which is read first for every new arriving packet.

1a) If the packet belongs to a real-time application, the mobile performs the FEC decoding using the RS code currently used (Mobile stations keep in memory an index indicating the code used for each connection), and delivers the packet to the layer above for further processing, after it has updated the register which maintains the status of each incoming packet (1: packets without errors; 0: packets with errors).

The content of the register is used for monitoring the quality of the radio link. Its contents can give easily the packet loss rate for the most recent N packets delivered. This number is checked every time a packet comes in against the cell loss rate of the QoS agreement. The mobile also maintains and the cumulative BER achieved in the connection, and it also compares this number with the BER of the QoS contract. Although these two numbers are not exactly the cell loss rate and the BER at the ATM layer, they are close, since the air packet is very similar in size with the ATM cells, and they are a good "tool" for assessing the performance of the code currently in use.

In case one of the two (or both) parameters are violated, the mobile issues a control packet to notify the transmitter to move to a more powerful code, and switches to a "checking mode" where it checks the last bit of the SN field. When the bit is set to one, it

is an indication to decode it with the next more powerful code from the code suite (The transmitter sets this bit to 1 in the first packet encoded with the new code).

1b) If the packet belongs to a non- real time application, the mobile (this is determined by table look-up checking the VCI value, to find the corresponding QoS parameters) performs ARQ decoding. If the number of errors is within the error correcting capabilities of the code, the mobile corrects the errors and delivers the packet to the layer above for further processing. A NAK is issued only when the errors are detected and the promised QoS is violated. MSEC is designed in a way that retransmissions are limited, only to the number that the QoS is not affected, and thus economizing in bandwidth. Figure 13 depicts the functionality of the mobile at the data-link layer.

4.3.4 The feedback mechanism

The feedback mechanism is an important component of the MSEC scheme. Its purpose is to monitor the quality of the link and create control packets to the transmitter to use a more or less powerful code. This is accomplished by monitoring key performance parameters at the receiver.

The monitoring of the link is based on a signaling broadcast connection with a data bit rate of ~1 kbit/s. In our system, since we are using narrow-band modulation (QPSK), it means that equally spaced time slots in the TDMA frame are allocated for signaling. The signaling time slots contain a single radio link packet with all the bits set to 0. Essentially this convention means that the VCI value of this broadcast connection is reserved to be zero. The receiver simply calculates the Hamming weight of the received codeword to determine the number of errors. These errors are used for the calculation of the (cumulative) BER and PLR, before the application of error control. The packet is then decoded using all the error control schemes used in the active connections. If the resulting codeword is the zero codeword, then decoding was successful, otherwise not. The parameters BER and PLR after the application of error control are now updated, and they are compared with their respective QoS parameters. If the latter are violated, a control packet is issued to the transmitter. On the other hand, if the QoS is maintained for a predetermined time (selected to be optimal), another control packet is generated to the transmitter (for switching to a less powerful code). In the simulation results we present in section 6, the feedback mechanism is used only for switching between different codes.

The BER (before and after error control) we present, are the actual parameters of the active connections, measured with the simulation tool.

CHAPTER 4

SIMULATION WORK

4.1 Simulation tools

For the simulation work of the thesis, the OPNET (OPtimized Network Engineering Tools) [19] simulation package was used. OPNET is a latest generation simulator, with built-in models of ATM and other protocols, which can be used in the development of custom models, thus reducing the coding effort to a large degree. The main distinctive features of the OPNET package are:

- Fully customizable built-in models for use in specific networks.
- Hierarchical, object-based modeling.
- Graphical representation of networks and node modules.
- Fully programmable process-level objects for detailed customization (e.g. on a physical link object: BER, capacity, propagation delay, etc.)

- Utilization of *proto-C* programming language (Consisting of graphical finite state machine environment with embedded C statements).
- Friendly graphical user interface.

The main concepts around which OPNET was built and operates are as follows:
One is the concept of a sequence of events that are programmed to happen at particular times during the simulation. The simulation kernel keeps the list with the pending events, and executes them in sequence, and if dictated by the underlying processes, it deletes some of them, or appends some more in the list. The pending events can be thought of as real life events (e.g. Packet arrival), and their execution drives the simulation.

The second main concept of operation is the formulation of software structures into meaningful entities called "objects". Although the simulator is based on the C programming language, which does not support object oriented programming (OOP), it is possible to think of the "objects" as a form of pseudo-OOP style programming. These "objects" also have real-life counterparts (e.g. Links, antennae). This formulation eases the graphical representation of network models, where objects have been given representational icons, and can be placed on the screen.

The third main concept of OPNET is the notion of *"interrupts"*. Interrupts occur at the execution time of a pending event, and they usually trigger a particular reaction. The reaction is usually the execution of a block of code located at the process level, and at the

process model which receives the interrupt. The term "interrupt" is derived from the mode of operation of OPNET: The system is "idle", until one event from the event list begins execution. At that time, an interrupt is delivered to the process model specified by the event, and the process "wakes up" and begins the reaction (execution of code).

Finally the fourth concept in work with OPNET is the layered architecture. The package gives the opportunity to the developer to design networks in a hierarchical manner. First the network model is specified, then for each of the nodes of the network, there is an underlying node model, where software modules are separated, and finally for each module there is one (or more) process model(s) in which the C code is inserted.

For the particular needs of simulating the performance of the MSEC scheme, specific OPNET models were developed, mostly from the ground-up, since the built in ATM models needed heavy modifications. In the next section the models used are described in more detail.

4.2 The network model

The network model used is depicted in Figure 14. It consists of five (5) node objects each of which has a particular functionality. The node *src* is the source of the traffic fed into the network. According to the OPNET methodology, the source node has a set of attributes that specify the type of traffic that is going in the network. Such attributes are

the bit rate, the packet size, the packet interarrival distribution, and in our case the QoS parameters. The node *node1* is a switching node, and its purpose is to read in the VPI/VCI values of the incoming packets, and direct them to the next node in the path.

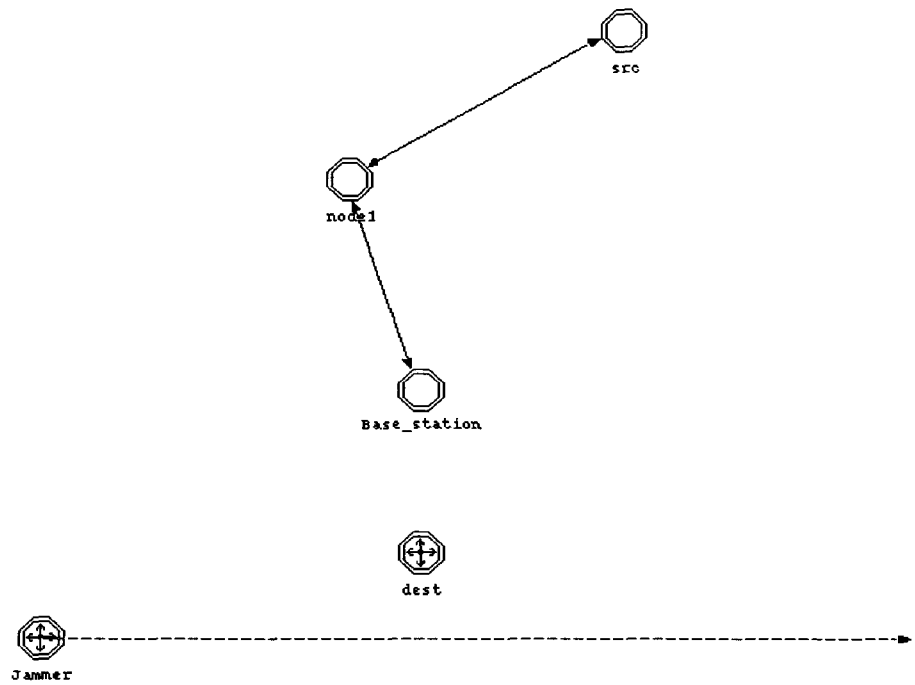

Figure 14: The network model

In our case, since the focus of the model is at the wireless part of the network, it simply passes the incoming packets from the source node object to the base station.

The *Base_station* node object is the internetworking point as specified in the description of the NGPCN. It receives the incoming packets from the *node1* node, performs the header translation, limited segmentation, encoding with RS or BCH codes, appends the new header, and forwards the packets to the *dest* node object (mobile station). At the *dest* node object, the reverse functions of the MSEC are performed, the original ATM cells are

reconstructed, and forwards them to the above layer for further processing. In our case, the layer above (AAL) is not explicitly modeled, so in its place there is only a packet destruction module. However, all the necessary interfaces are present, so an upper layer protocol can be easily appended to the model.

4.3 The node models

Each of the node objects shown in Figure 14, has an associated, underlying node model, in accordance to OPNET modeling conventions. The node models of the nodes are presented below.

a) The src node model

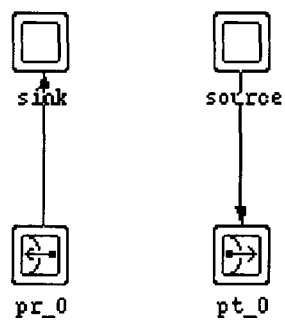

Figure 15: The source node model.

This node model is located inside the *src* node shown in network Figure 14.

The *source* module is generating the traffic specified by the user in the attributes of the *src* node object as described previously at the network model. The *sink* module would be used in case the user wished to establish a two-way communication. The *pt_0* module is the point-to-point transmitter, and it forwards the generated cells to the *node1* node object.

b) The node1 node model

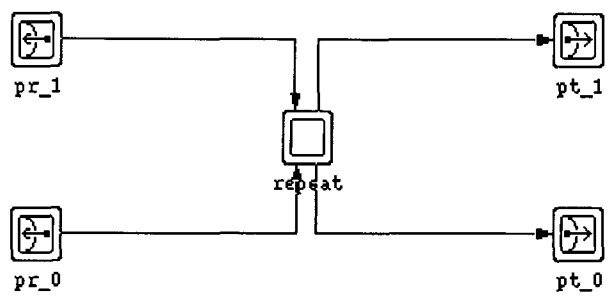

Figure 16: The *node1* node model.

This node model is located in the *node1* node shown in the network Figure 14.

In this case, the model only has to direct the incoming cells to a single node, so only a pair of receivers and transmitters suffice. In the case the switch had to direct traffic to various directions, many pairs of transmitters/receivers would be required.

c) The Base_station node model

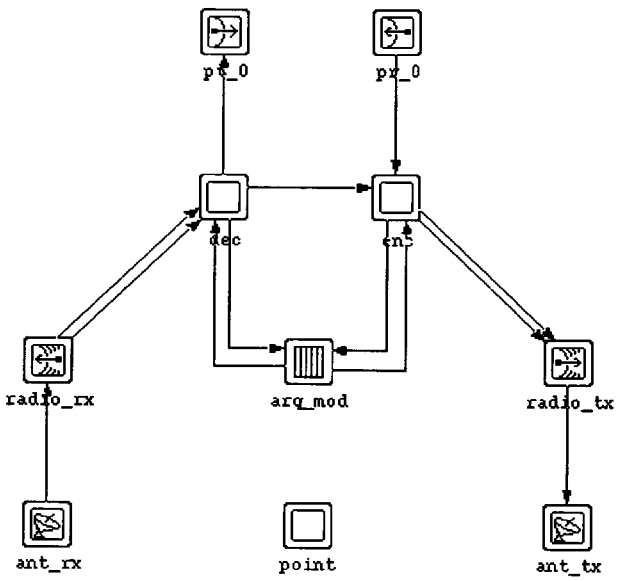

Figure 17: The Base_station node model.

This node model is located inside the node *Base_station* node in Figure 14.

The incoming ATM cells are received from the *pr_0* module (point-to-point receiver) and forwarded to the *enc* module where the segmentation, generation of new headers, and encoding with the appropriate code takes place. The *arq_mod* module implements the sliding window protocol, which is used to keep copies of transmitted packets that belong to time-tolerant connections, for the case that the receiver requests a retransmission. The *dec* module is used in the opposite direction, when packets are received, and they are decoded, using the currently used code. Also in the *dec* module the code for the link

monitoring is located, and control packets are generated. In addition the *dec* module performs the reassembly of the ATM cells and passes them to the upper layer (the *pt_0* module in this case). The *dec* module also passes packets to the *arq_mod* for the generation of acknowledgment (ACK) or not acknowledgment (NAK) packets.

d) The dest node model

This node model is located inside the *dest* node shown in Figure 14.

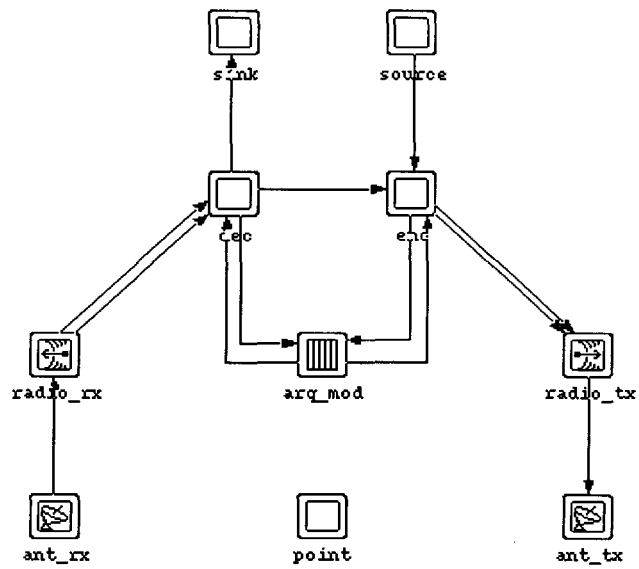

Figure 18: The mobile node model.

The *dest* node model is essentially the same with the *Base_station* module, with the difference that it receives the packets from the *ant_rx*, the antenna module, and it is

connected to *source* and *sink* modules at the upper layer, for the generation of traffic (user-defined for two-way communication), or the destruction of cells that go out of circulation to free memory. The node model of the mobile node is given in Figure 18.

e) The jammer node model

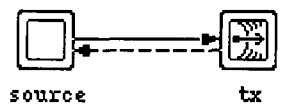

Figure 19: The jammer node model.

This node model is located inside the *Jammer* node shown in Figure 14.

The *jammer* node is used for generating interference. The power of the transmission is controlled by the user, and it is possible to change it over time in a way that the resulting SNR would fluctuate. In addition, the *jammer* node can be disabled and enabled at regular or random time intervals, in a way that it simulates bursts of noise. The node model of the *jammer* consists of a source of packets, and the radio transmitter *tx*. The packets transmitted are designed to collide with user packets destined from the base station to the mobile station, and inflict errors on them.

4.4 The process models

According to the OPNET modeling methodology, every module of every node object that we described so far contains at least one process model, which contains the C code used for responding to interrupts delivered to that particular process. In the model used for the simulation runs, most of the process models are user defined. In more detail, the process models used are as follows:

a) The encoding process model

This process model is located inside the *enc* modules shown in figures 17 and 18.

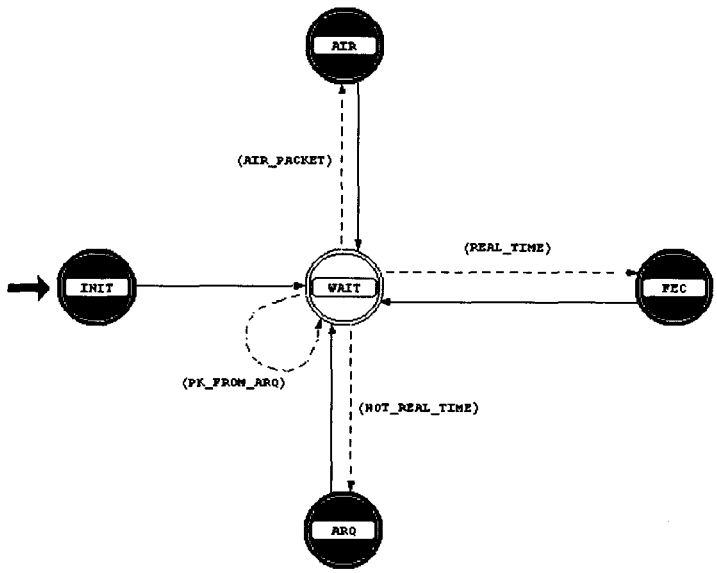

Figure 20: The encoding process model.

The process model can be explained as follows:

The state INIT is the state where the process is "resting" at the time the simulation begins. The attribute *beg_sim_intrpt* of the process model has been set to TRUE, so that the simulation kernel delivers an interrupt to the process. The process executes the C code that is written inside the state INIT, (initialization of variables, creation of lists,..) and transits unconditionally to the state WAIT. Here it waits for interrupts, which are arrivals of ATM cells or arrivals of control packets which contain information about switching to more or less powerful codes, depending on the SNR of the radio link and the achieved QoS parameters. The process then transits conditionally to one of the states FEC, ARQ, or AIR, depending on the value of the VCI/VPI field of the cell, executes the code contained there, and unconditionally returns back to the WAIT state and stops execution, returning control to the kernel. The FEC state contains the code for encoding cells with RS codes, while the ARQ state contains the code for encoding the packets using BCH codes. The AIR state is checking the fields of the control packet and switches to more or less powerful codes for error protection.

b) The decoding process model

The process is located inside the *dec* modules shown in figures 17 and 18.

The decoding process model, like the encoding process, uses an INIT state, where all the initializations take place, and transits to the WAIT state waiting for interrupts. The process is going to the state TO_ARQ under the condition that the packet belongs to a

time-tolerant connection, to the state FROM_ARQ if the packet is coming from the arq_mod module and destined for an upper layer, and to the state VCI/VPI otherwise.

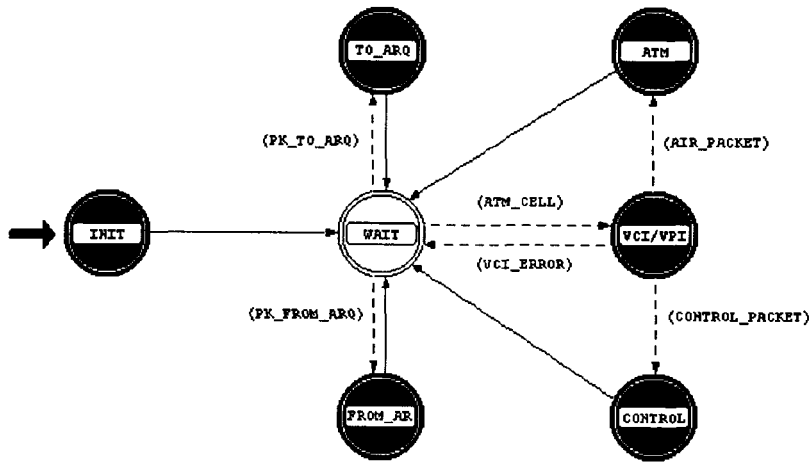

Figure 21: The decoding process model.

From the VCI/VPI state the process may transit to one of the states CONTROL or ATM, depending on the value of the VCI/VPI field. Data packets are directed for decoding to the ATM state, while control packets are directed to the CONTROL state.

c) The sliding window process model

The process is located inside the *arq_mod* modules shown in figures 17 and 18.

This process, originally built-in the OPNET simulator received some modifications in order to handle traffic from different sources, and was provided with interfaces to the *enc* and *dec* modules communicate information about the active connections. The figure describing the process is given in Figure 22.

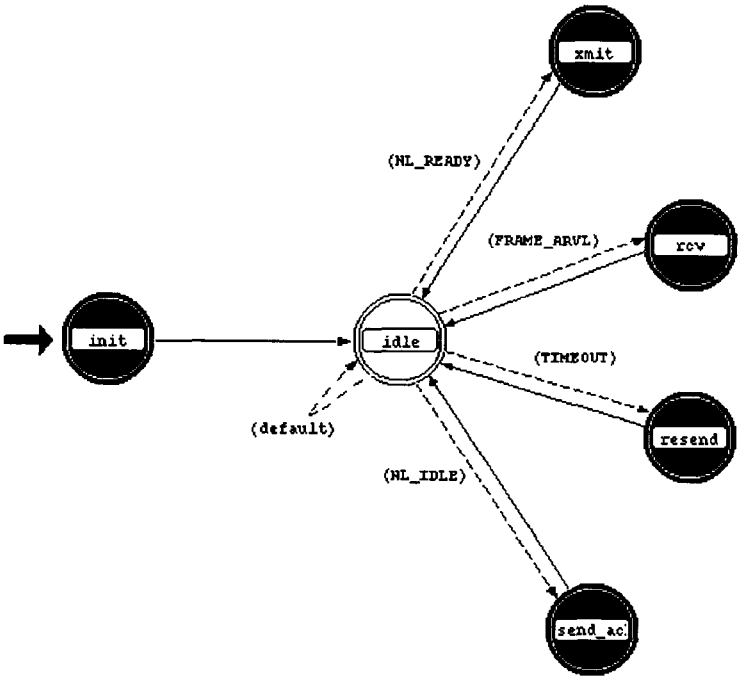

Figure 22: The sliding window protocol process model.

The process starts at the state INIT, where all initializations take place, and transits unconditionally to the state IDLE, where it waits for interrupts. Possible interrupts are:

- Arrival of packet from the network layer (NL_READY), where a packet arrives for transmission, it is copied in the transmit buffer, and forwarded to the transmitter module.
- Arrival of a packet from the receiver module, where the packet is placed in the receive buffer, the piggy-backed acknowledgment is handled, and packets are passed up to the network layer.
- The retransmit timer expires (self interrupt), where an acknowledgment for a previously sent packet has not been received within an acceptable amount of time, and the packet is considered lost, and is retransmitted.
- If the network layer has been idle for too long, there are not outgoing packets for the sliding window protocol to attach acknowledgments for received packets on them, and it issues a separate acknowledgment frame and sends it.

d) The traffic source process model

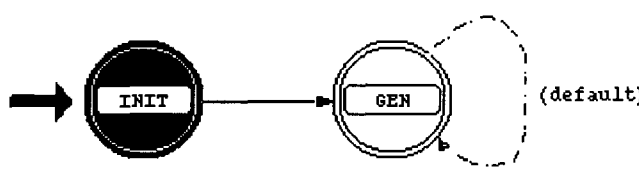

Figure 23: The traffic source process model.

This process model is contained in the *source* module shown in Figure 15.

The process is resting at the state INIT, where at simulation start time, it reads all the user input for the traffic to be generated (At simulation run time the user is prompted to input these parameters), such as bit rate, packet size, and QoS parameters. When the process has all the information needed, it schedules self-interrupts for times it must generate packets, as a sequence of events. The process then transits to the state GENerate traffic, waiting for the self-interrupts, where it waits for interrupts to arrive, and generate the requested traffic.

CHAPTER 5

RESULTS

The MSEC scheme as described in chapter 3 has been simulated using the models described in chapter 4. In the simulations some assumptions have been made, which do not affect the functionality or performance of the proposed error control scheme, while simplifying the coding effort to a large degree. Probes have been put inside the network at points at which we want to get measurements of parameters of interest. The description of the simulation parameters used, and the overall context of the simulations are described in the following section.

5.1 The context of the simulations

The results of three simulation runs are presented, with different noise levels each time. The purpose such structure of the simulations is to demonstrate how the MSEC behaves in a variety of conditions. We anticipate that at low bit error rates the MSEC will utilize the least powerful codes from the suite of available codes, to conserve bandwidth,

and increase the throughput, while at high bit error rates, it will switch to more powerful codes to maintain the promised QoS. In the network used, the fiber link speeds have been set to 155 Mbits/s, thus simulating the ATM OC3 speed, most widely used in ATM networks. The ATM cells generated at the traffic source are statistically multiplexed into the OC-3 frame. Two services are generated at the traffic source, one of which has characteristics of a time-sensitive service, with a constant bit rate (CBR) of 6kbits/s, while the other one represents a time-tolerant service with a constant bit rate of 5kbits/s. The time-tolerant service of 6kbits/s may be thought as a compressed voice signal with the quiet periods of the speech not coded. However, we avoid to use the term "voice", since voice over ATM is not standardized as of today by the ATM Forum, and equipment manufacturers are hesitant to provide proprietary schemes that may be outdated when the standardization happens. This selection is representative, since a variable bit rate (VBR) service differs in the way the call is admitted in the network, and once it is admitted it has guaranteed bandwidth. However, the way the error control scheme is performed is independent from CBR or VBR characteristics. The speed of the link is set to 19.2kbits/s, allocated at call setup so it would be enough to accommodate both services and the overhead bits to be transported. The channel sharing technique is TDMA, and the modulation is QPSK. The channel sharing technique is not explicitly simulated. Research is currently being carried out for the fair sharing of the radio channels, in order to accommodate integrated services in a reliable manner [31]. In our case, we assume that the two services have guaranteed bandwidth to be transported to the mobile station. In summary, the assumptions made for the simulations are as follows:

- The connection setup protocol of ATM has not been included. That is the Q.2931 protocol for establishment and tear down of the connections. We assume a load of two services one time-tolerant, and the other time-sensitive.
- We assume the Medium Access Control/ Channel sharing problem in the inbound (mobile to Base Station) channel has been solved, and the two services under consideration have guaranteed bandwidth. The outbound channel is shared with asynchronous TDM.
- We do not assume hand-offs during the course of the simulations.
- We use QPSK as our modulation scheme, which is built in the OPNET simulator models.

The jammer node in the network is used for the generation of noise. Precisely, the jammer node is generating co-channel interference, and the power of transmission is one of the attributes of the simulation. Thus, we can adjust the power of transmission to control the SNR appearing at the receiver. The OPNET simulator also calculates the level of the background noise as a part of the radio transmission pipeline procedure. In addition, the jammer is following the path shown in Figure 14 for the entire 1000 seconds of the simulation runs, so the SNR changes as the jammer approaches or goes away from the mobile station. The jammer is set to go on and off periodically (Idle for 10 seconds, then active for 2 seconds), in a way that represents burst errors in the received packets. The errors in the information packets are calculated as follows: The simulator has built-in code that calculates the SNR, given the power of the jammer node, the power of

transmission of the two communicating nodes, and the relative distance between them. Thus, as the jammer moves closer or further from the nodes, the SNR changes at regular time intervals. Between changes, the value of the SNR is considered constant, and the simulator calculates the BER for this time "slot", and the number of errors in the segment of the packet being transmitted during this time interval. A single packet may have segments inside it that have different BER, and thus more or less errors.

Among the most important parameters for the simulation of the performance of the MSEC is the QoS contract established during connection setup. The signaling protocol Q.2931 [10], and ATM Forum's User-Network Interface (UNI) specification [3] provide fields in their SETUP messages for the specification of these parameters. The base station, being the internetworking point operates up to the ATM Adaptation Layer (AAL), and keeps the QoS parameters of each requested connection. The mobile, as the destination node has also knowledge of these parameters. The QoS parameters used for the simulations are shown in table 4.

	TIME-SENSITIVE SERVICE	TIME-TOLERANT SERVICE
CLR	10^{-2}	10^{-3}
BER	10^{-3}	10^{-4}
SOURCE BIT RATE	6kbps	5kbps
ETE DELAY	80ms	~10s

Table 4: The QoS contract used in the simulations.

5.1.1 The fading model

The wireless computing environment, such as the one assumed in our simulations exhibits differences with the cellular telephony environment. In cellular telephony, we assume the user can roam freely within an area, and at a variety of speeds. In such environment we expect frequent changes in the SNR, and correspondingly in BER. In the mobile computing environment however, we expect a different behavior from the mobile users, which would reflect in the SNR at the receiver. In a mobile computing environment, the user is expected to be stationary, but *potentially mobile* at any given time. The above assumption is dictated by the very nature of mobile computing, which requires stability for prolonged periods of time. As a result, we expect that in the mobile computing environment co-channel interference would be a major factor in the SNR. Many modern electronic devices generate frequencies that may interfere with the frequency of the mobile station, and thus generating unwanted noise. In addition, the user may also be moving at higher speeds (operation in a car), and the fading model can be quite complex. In our simulations the BER observed, and the number of errors in the information packets is of significant importance. We control the BER by means of generating co-channel interference using the jammer. Variation in the received SNR is a result of fading, so under this view, we have a slowly fading channel (movement of the jammer, see figures 24, 41, and 57), with regular intervals with co-channel interference (the downward "spikes" in the same figures).

If the model was different, such as a fast fading channel, we would obtain different results For example, the switching delay measured in our simulations is approximately 0.6 seconds, would probably be slow for a fast fading channel. Certainly, differences would occur with a different set of assumptions. For example, the switching delay would be significantly lower if we operated in a wireless LAN environment, where the speed of the link in the order of 2 Mbits/s. In such case, the bit rate of the multicast control connection would be higher, and the switching delay would fall in the order of milliseconds, which would be sufficient even for a fast fading channel.

5.2 The parameters of interest

As representative results of the simulations we have the ones related to the QoS contract, such as Packet Loss Rate (PLR), and End To End (ETE) Delay. In addition, we want to measure two more time measurements, one for the adaptation time of the MSEC scheme (time required for the transition to a more or less powerful code), and one for the time it takes the MSEC to bring the achieved QoS parameters again in acceptable limits once they have been violated. The structure of the presented results is in increasing SNR, with the last simulation run the BER at the receiver has been as high as ~0.09. The results presented for the three simulation runs in section 5.3 are as follows:

- Number of lost packets
- Packet Loss Rate
- End-To-End Delay
- Number of corrected packets
- Cumulative BER
- BER per packet (Instantaneous BER)
- Code in use (Indicates the RS or BCH code used at any given time)
- Number of retransmissions
- Signal-to-Noise Ratio (in dB)

5.3 Results

5.3.1 The first simulation run

For all the simulation runs the power of transmission of the Base station was set at 5 Watts, and the power of transmission of the mobile station set at 100mWatts. The power of the jammer node was set at 700mWatts.

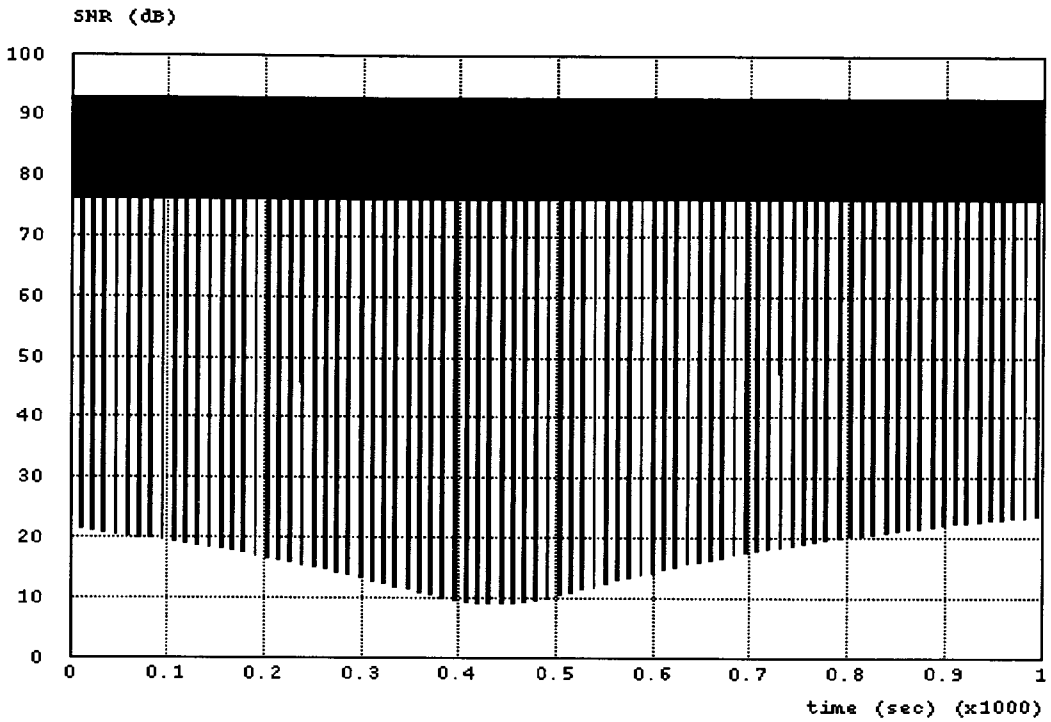

Figure 24: SNR (dB).

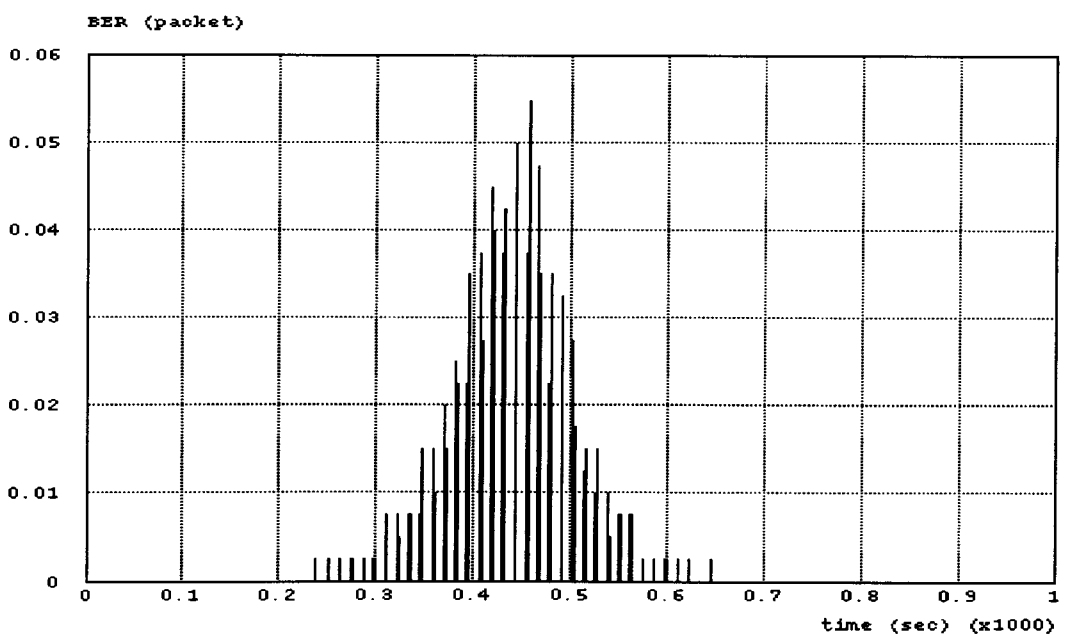

Figure 25: Bit Error Rate (Instantaneous).

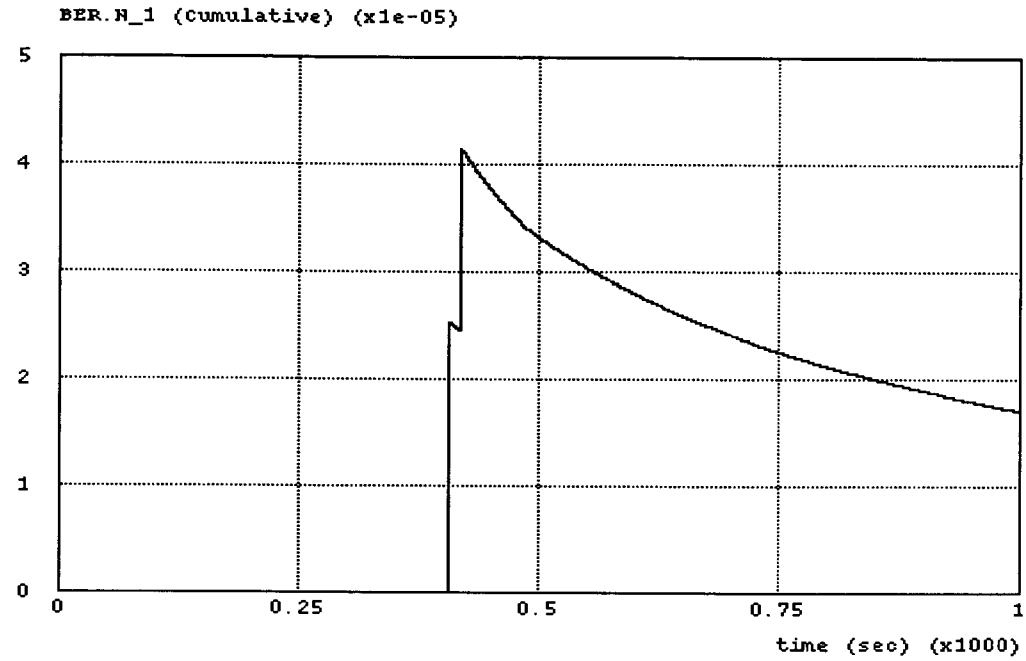

Figure 26: Bit Error Rate (Cumulative-ARQ performed).

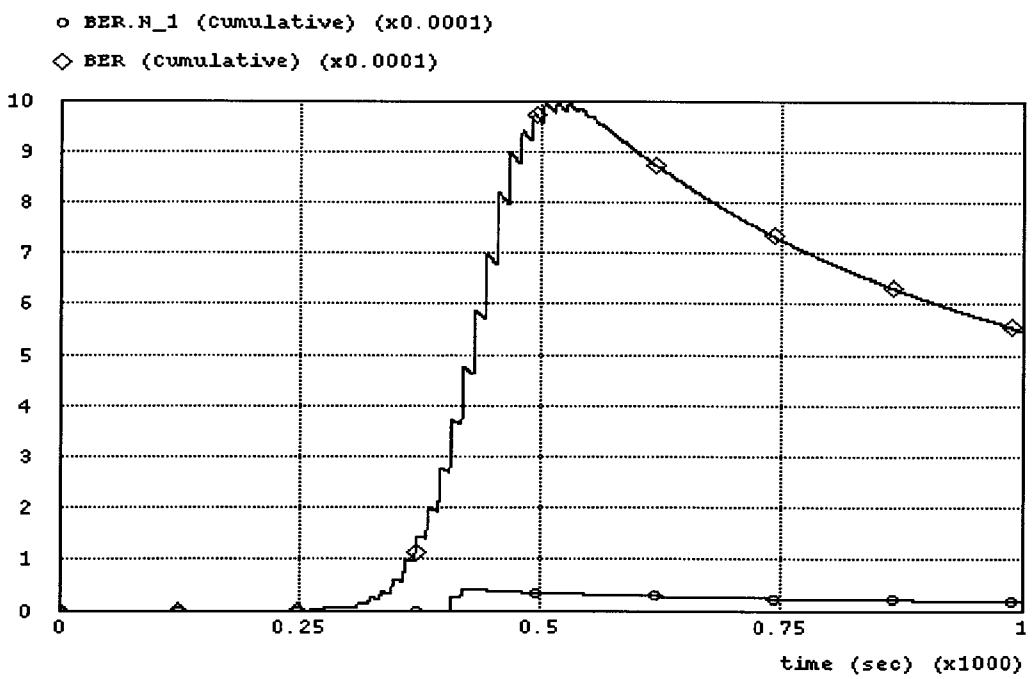

Figure 27: Bit Error Rate (Before-After).

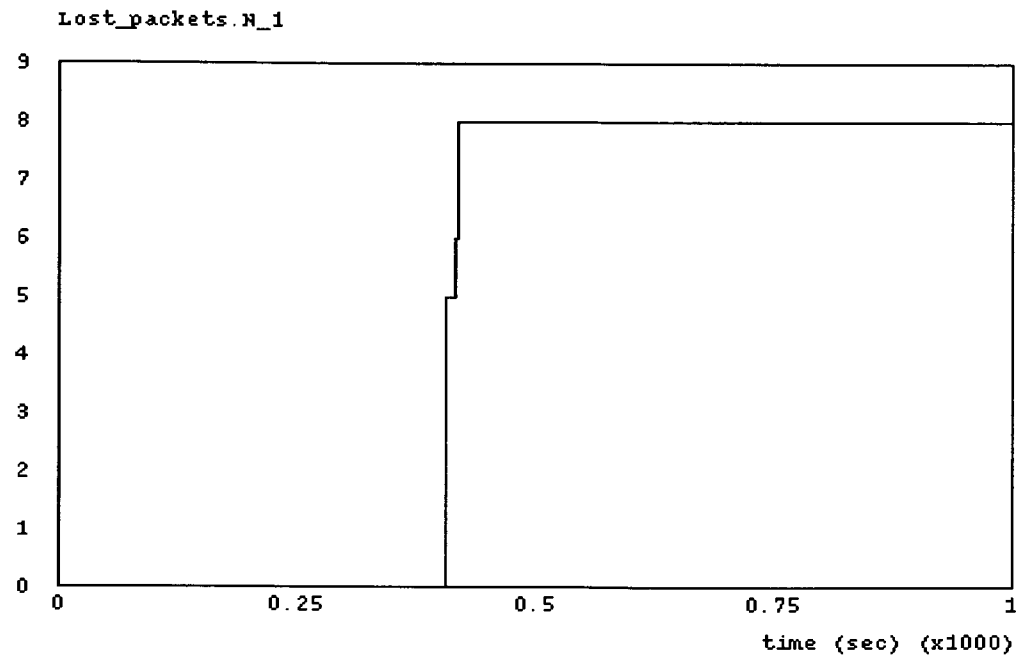

Figure 28: Lost packets (Time-tolerant service).

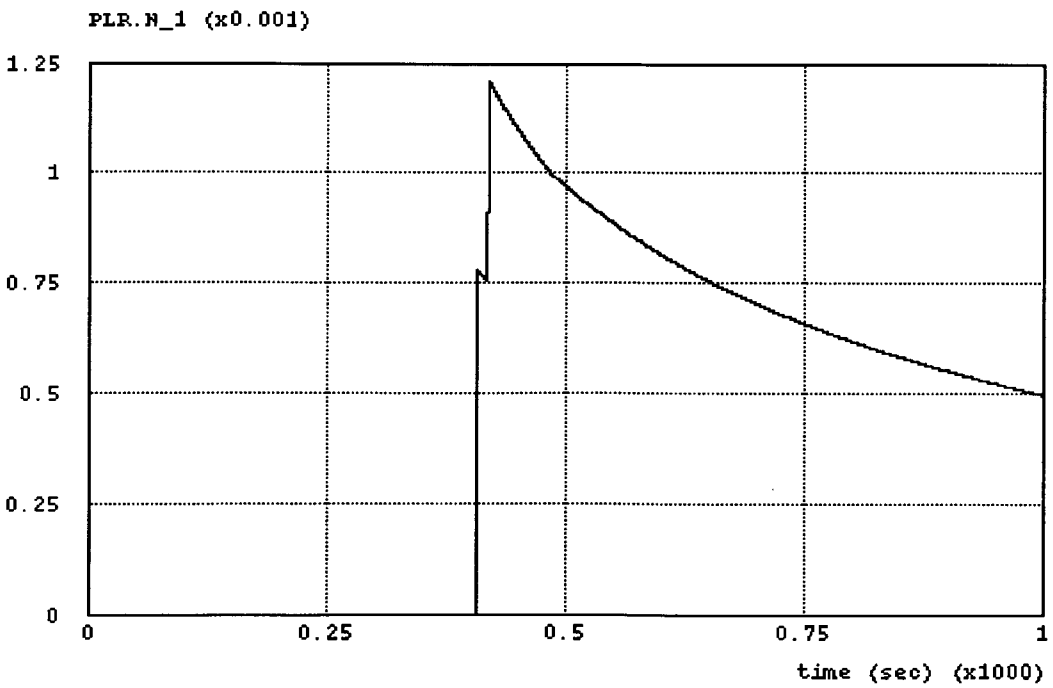

Figure 29: Packet Loss Rate (Time-tolerant service).

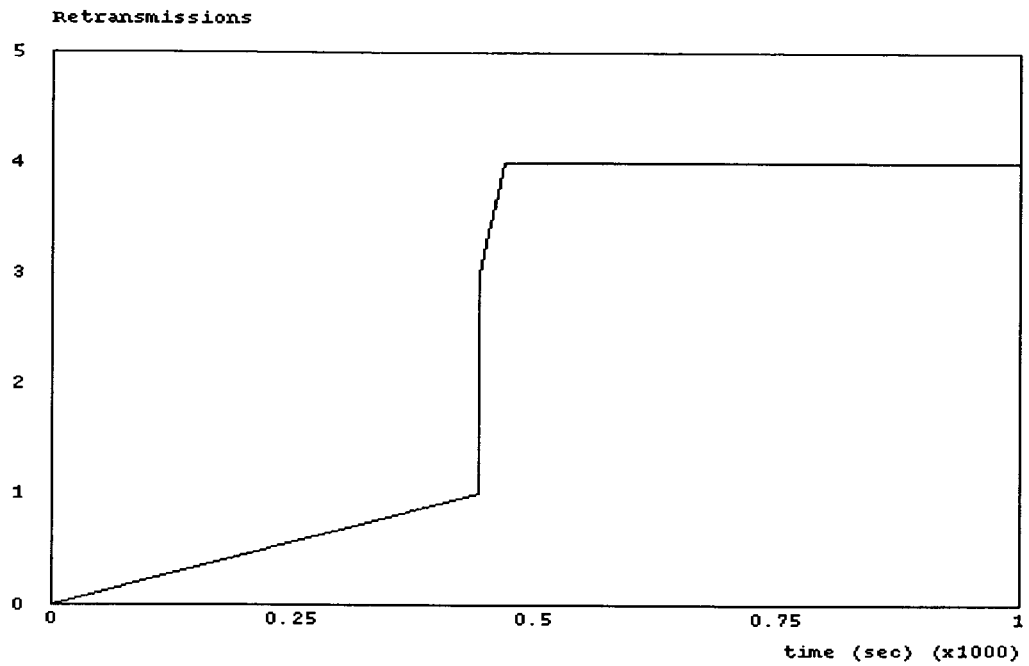

Figure 30: Number of retransmissions.

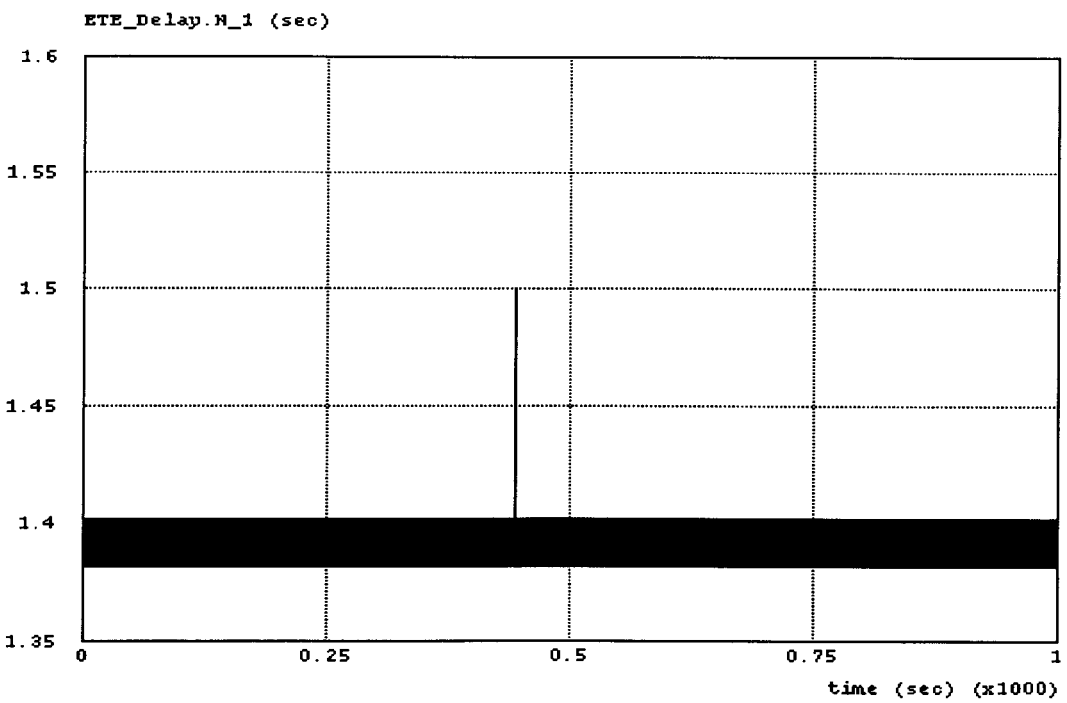

Figure 31: End-To-End Delay (Time-tolerant service).

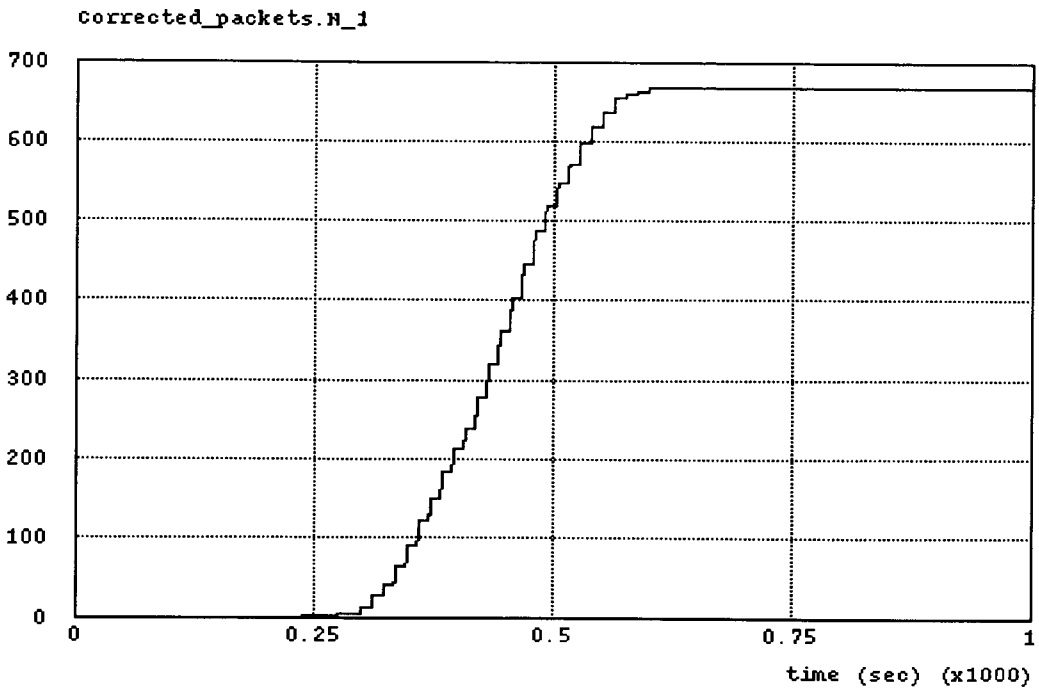

Figure 32: Corrected packets (Time-tolerant service).

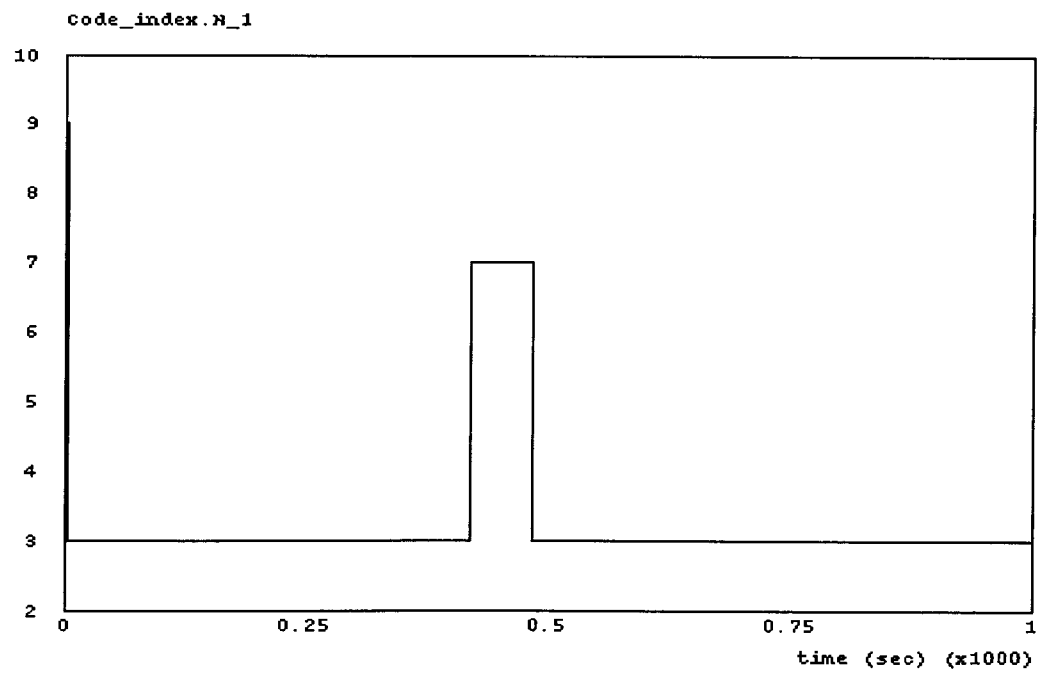

Figure 33: BCH Codes used (Minimum distance).

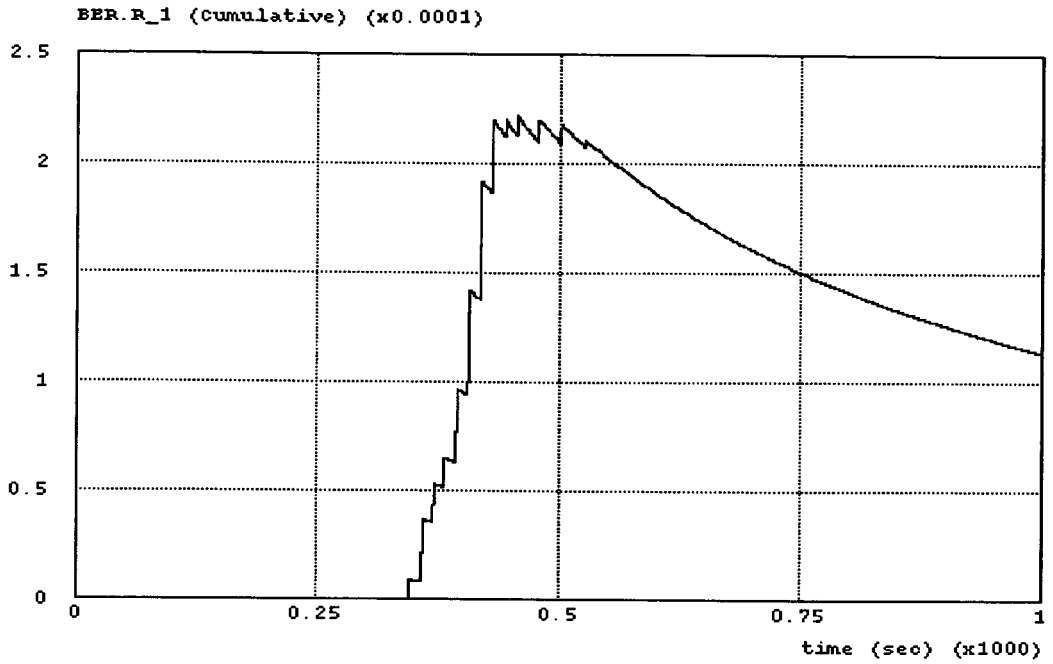

Figure 34: Bit Error Rate (Cumulative-FEC performed).

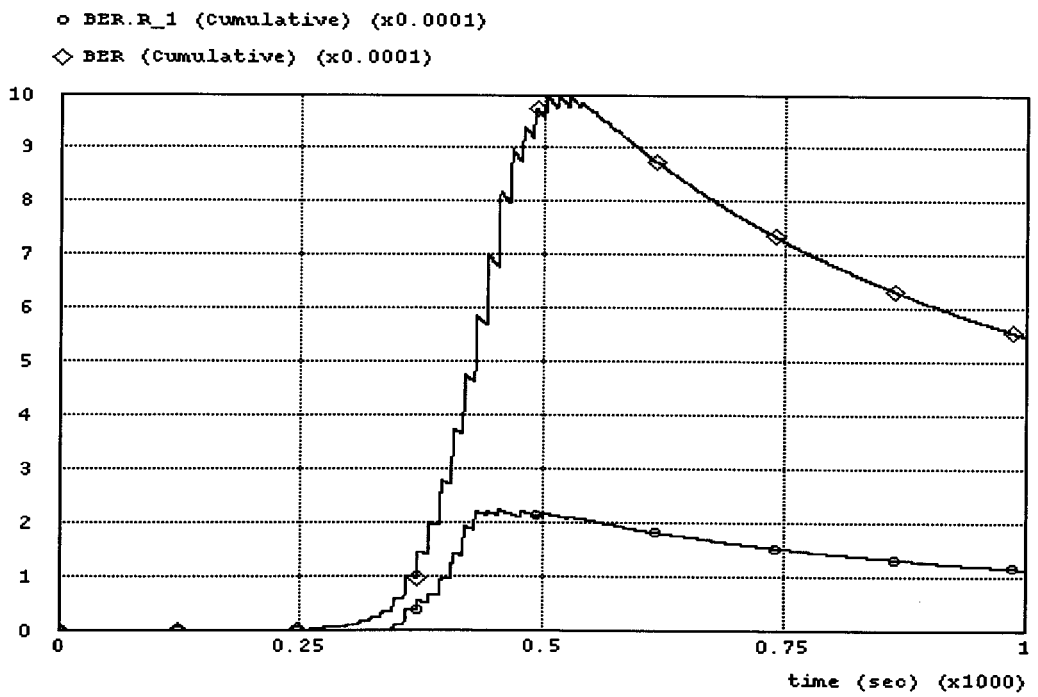

Figure 35: Bit Error Rate (Before-After).

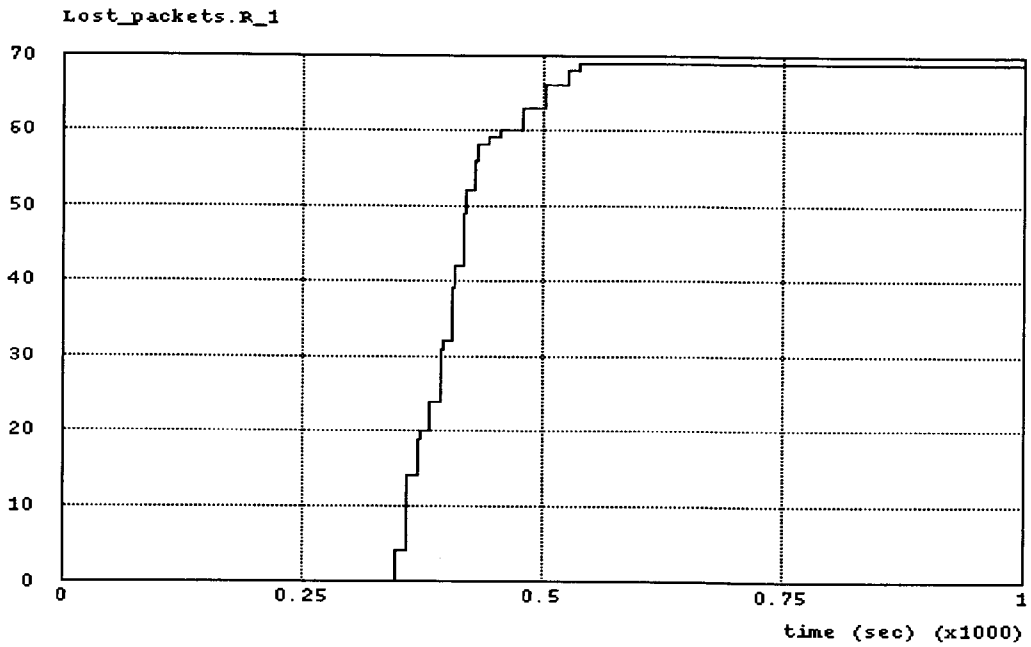

Figure 36: Lost packets (Time-sensitive service).

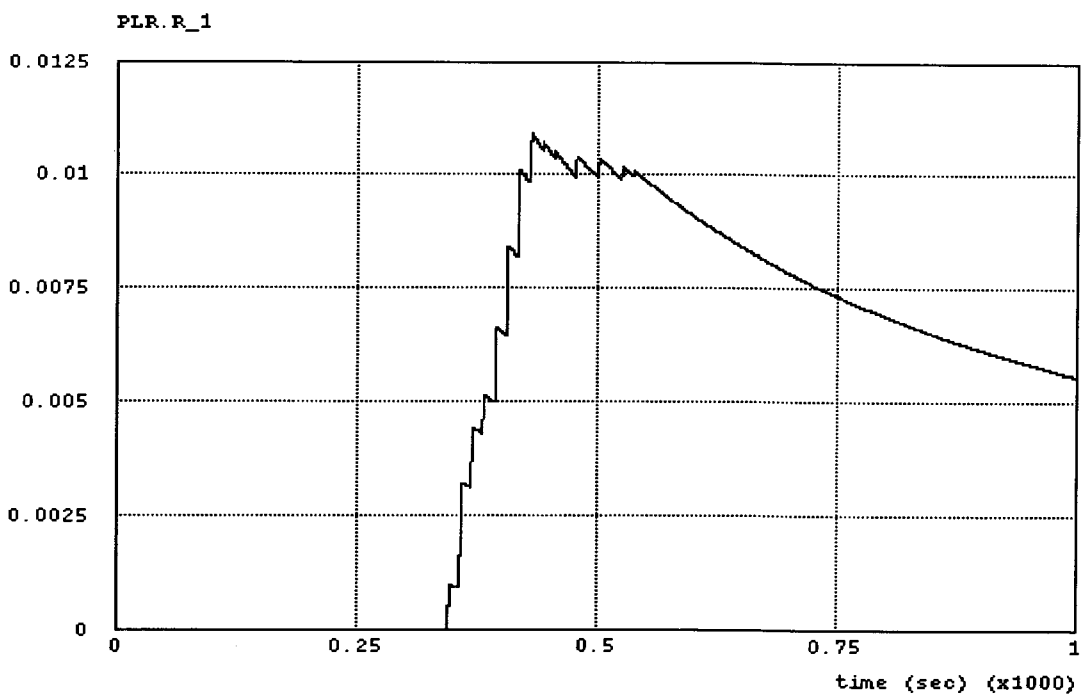

Figure 37: Packet Loss Rate (Time-sensitive service).

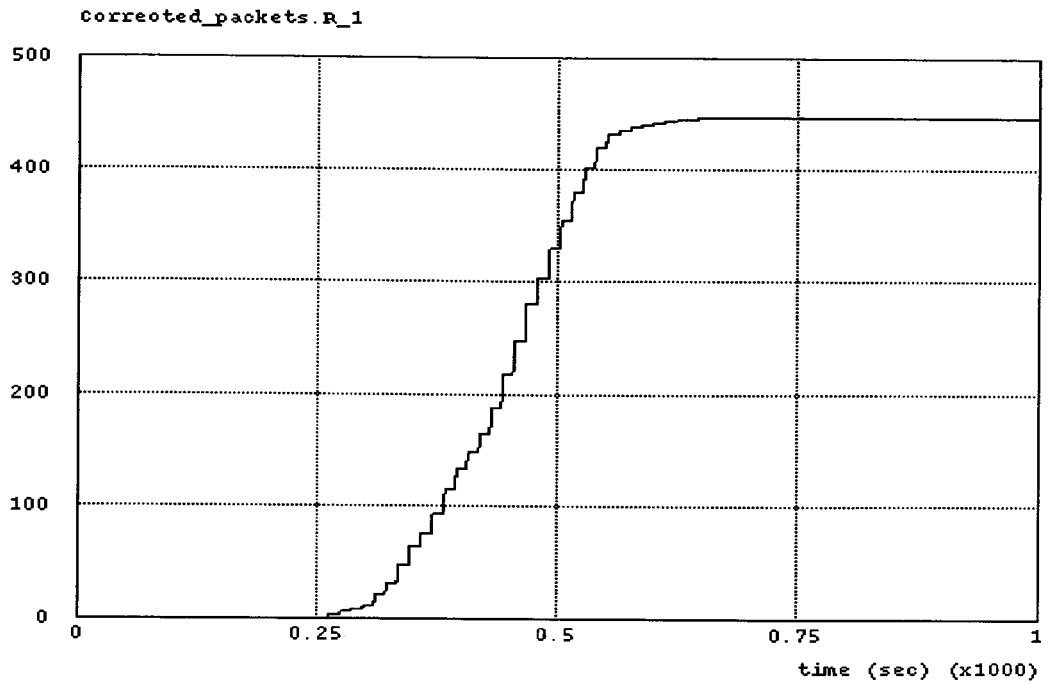

Figure 38: Corrected packets (Time-sensitive case).

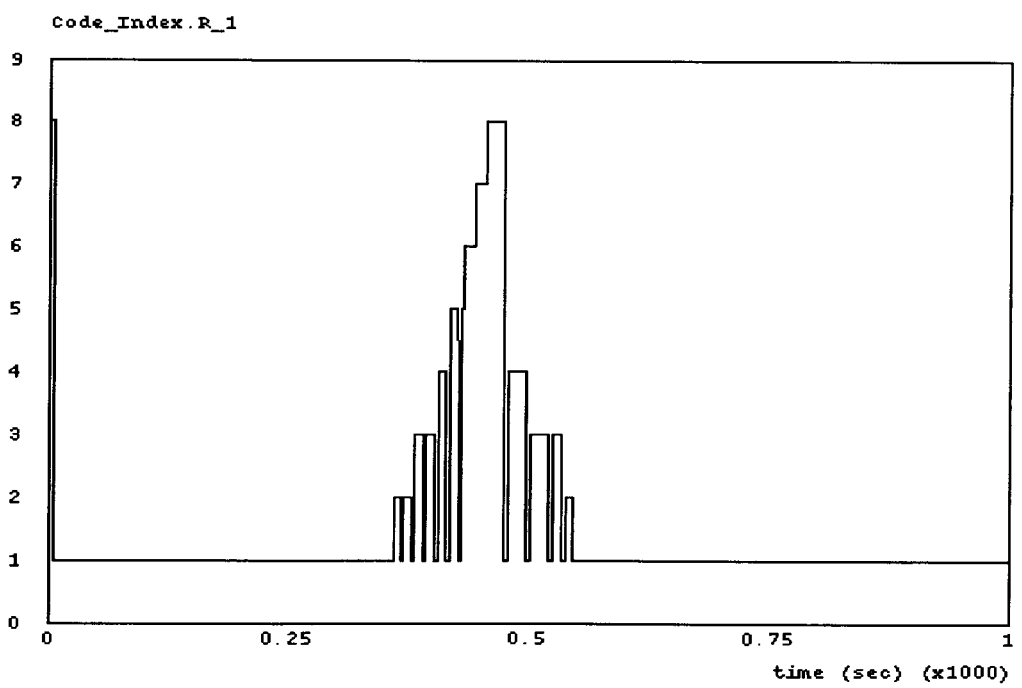

Figure 39: RS code used (Correctable 6-bit words).

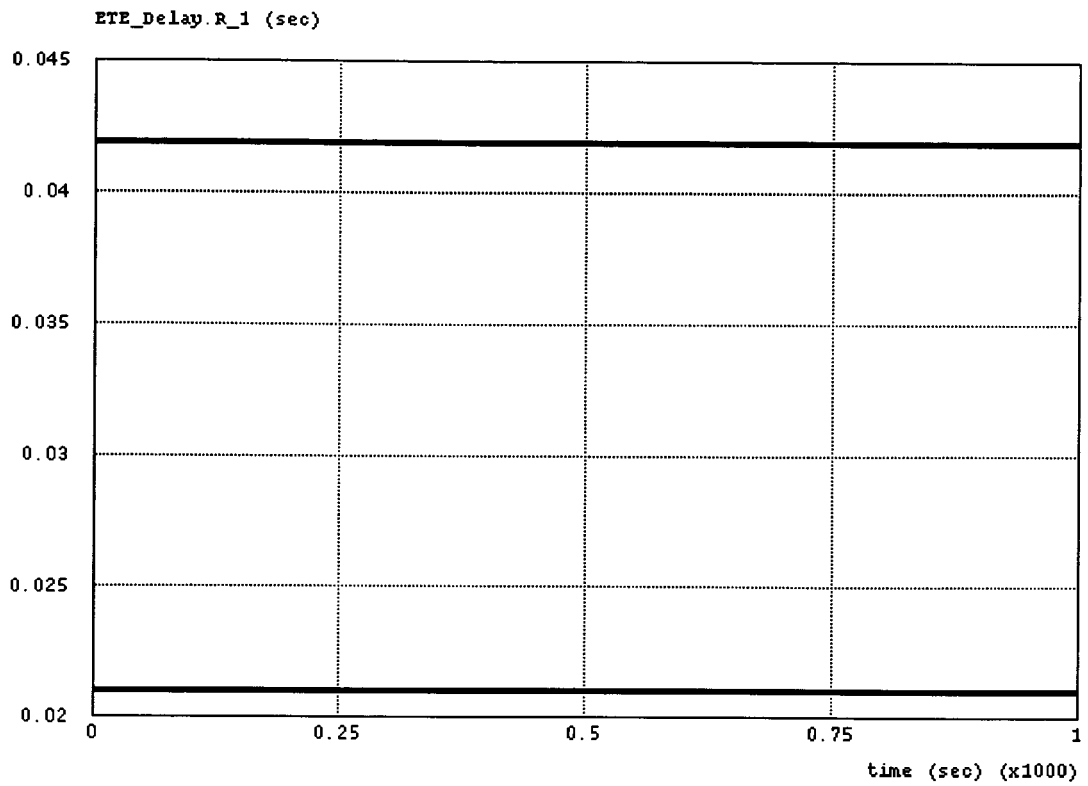

Figure 40: End-To-End Delay (Time-sensitive service).

5.3.2 The second simulation run

For the second simulation run the power of the jammer node was set at 900mWatts, while the power of the base station and the mobile station remained the same.

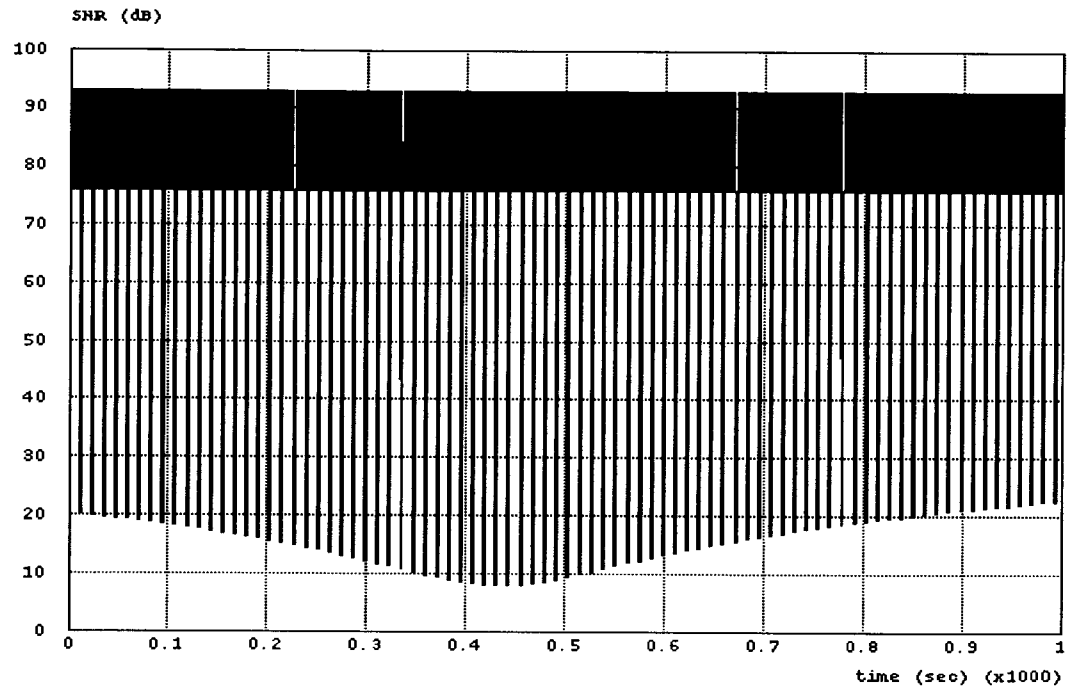

Figure 41: SNR (dB).

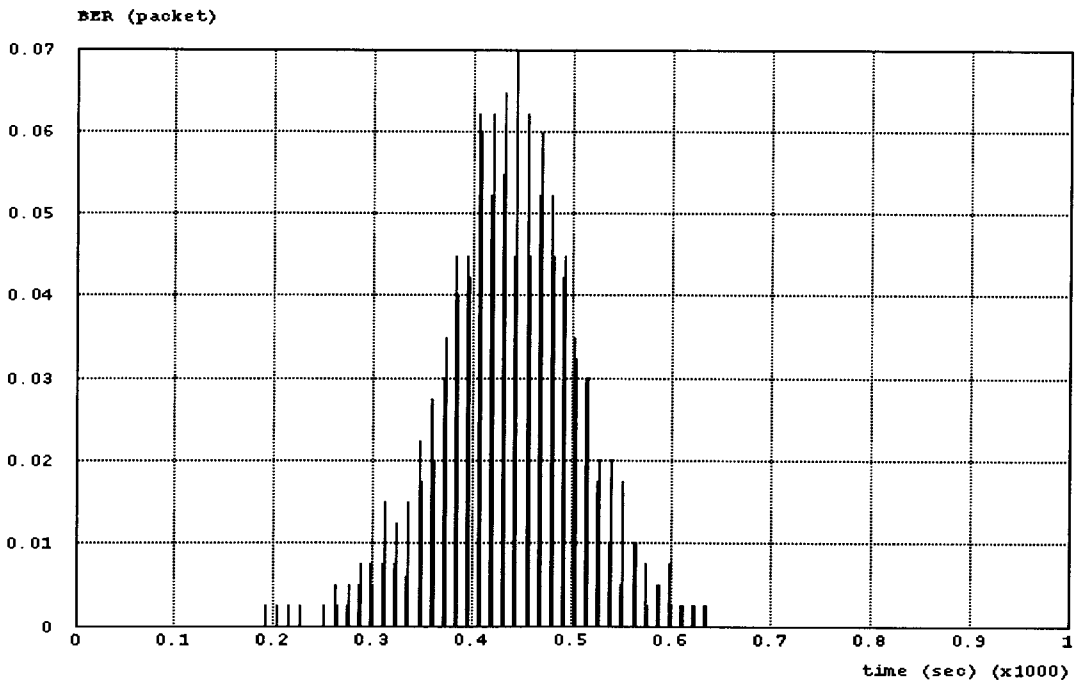

Figure 42: Bit Error Rate (Instantaneous).

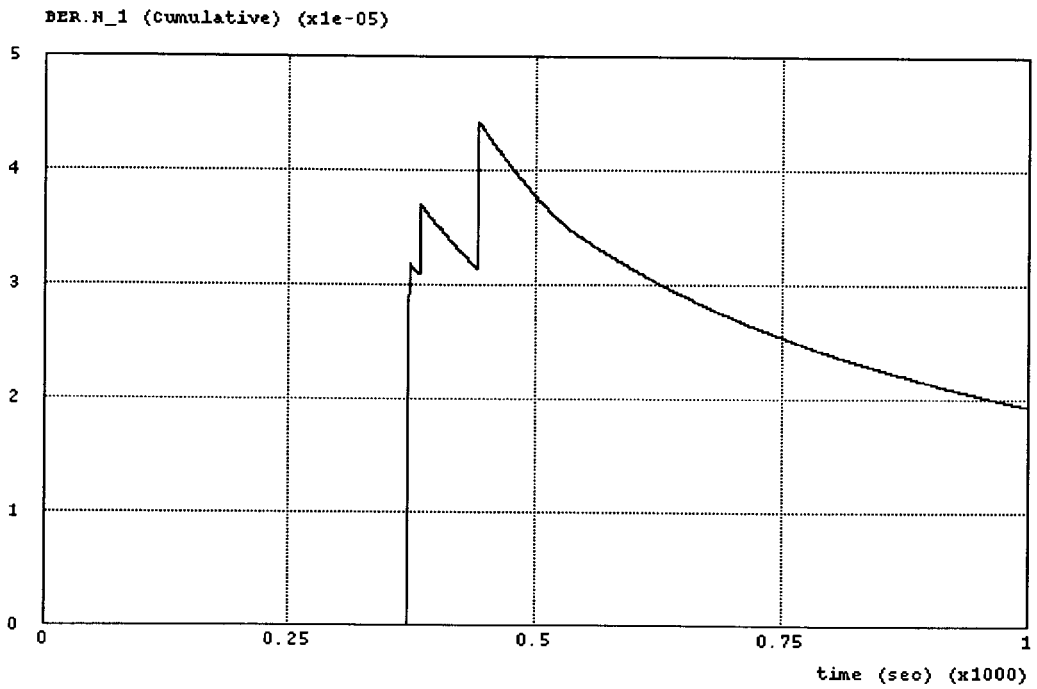

Figure 43: Bit Error Rate (Cumulative-ARQ performed).

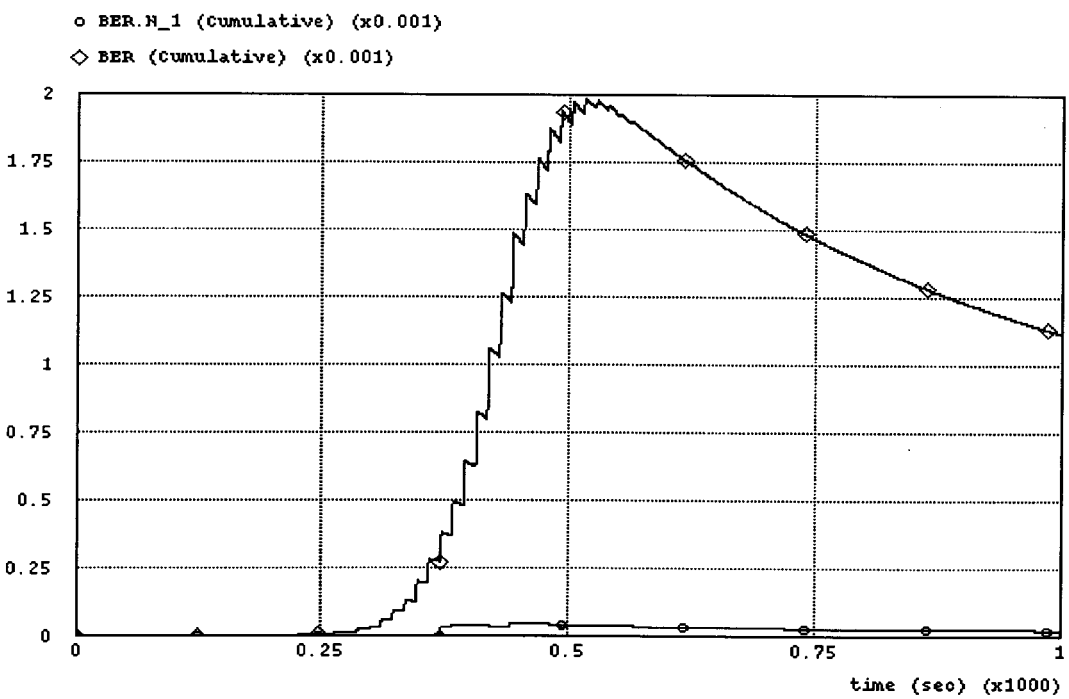

Figure 44: Bit Error Rate (Before-After).

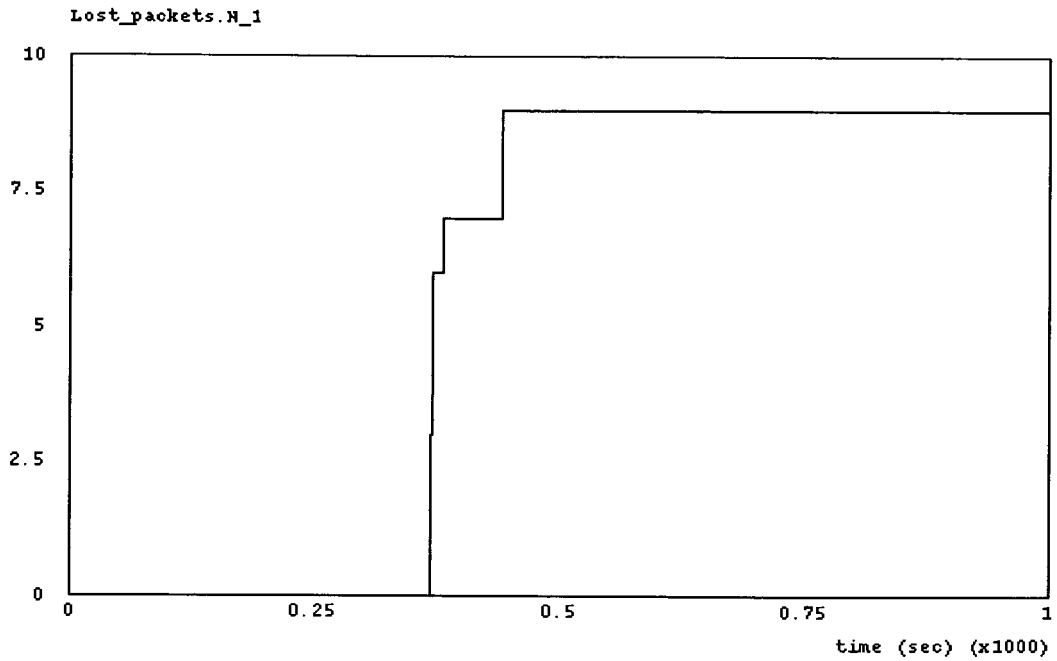

Figure 45: Lost packets (Time-tolerant service).

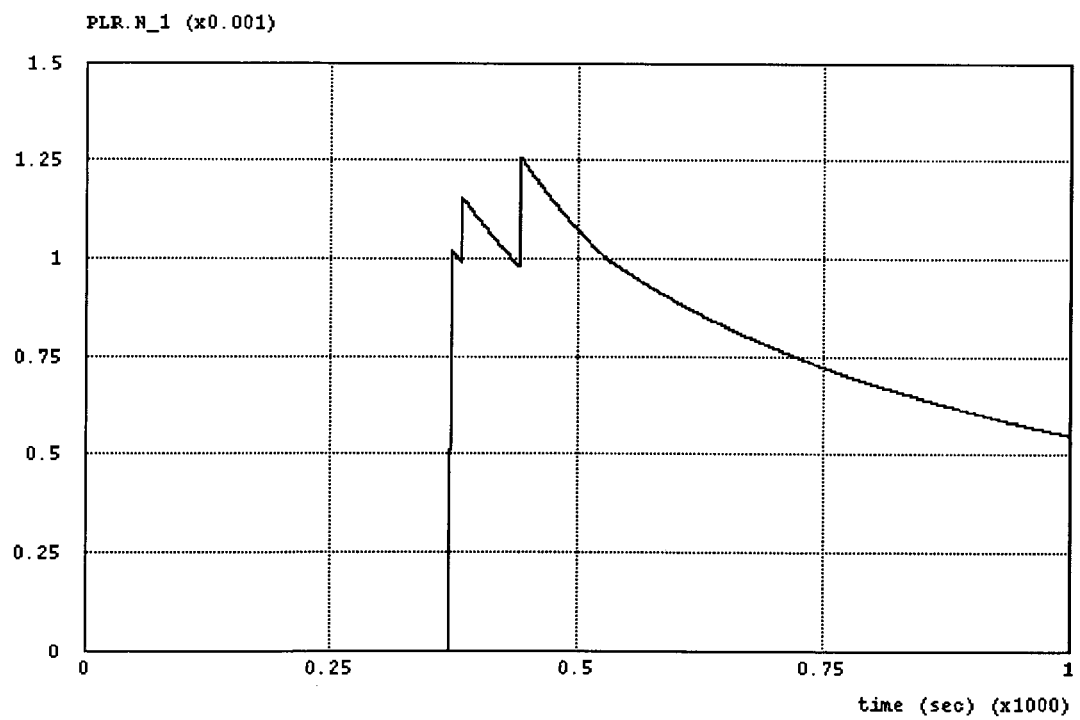

Figure 46: Packet Loss Rate (Time-tolerant service).

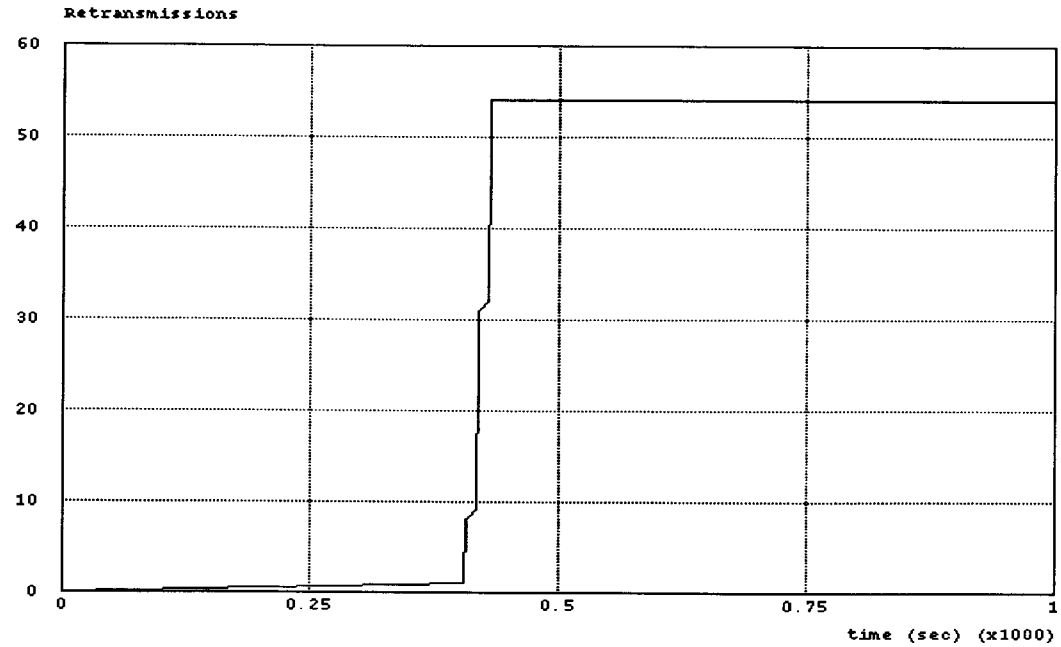

Figure 47: Number of retransmissions.

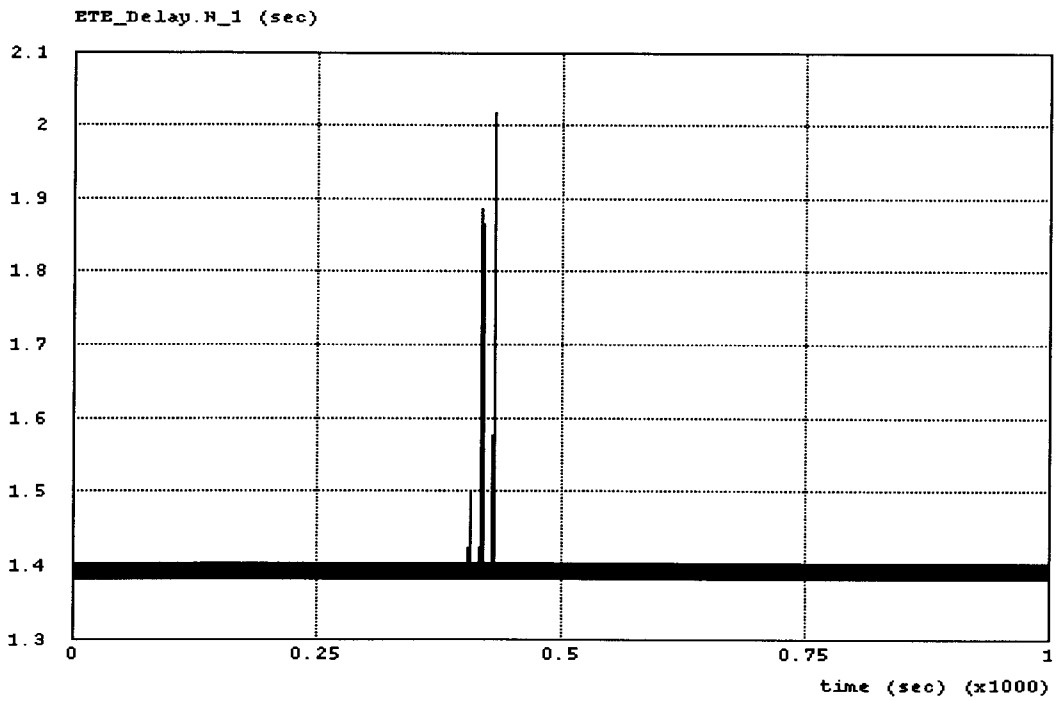

Figure 48: End-To-End Delay (Time-tolerant service).

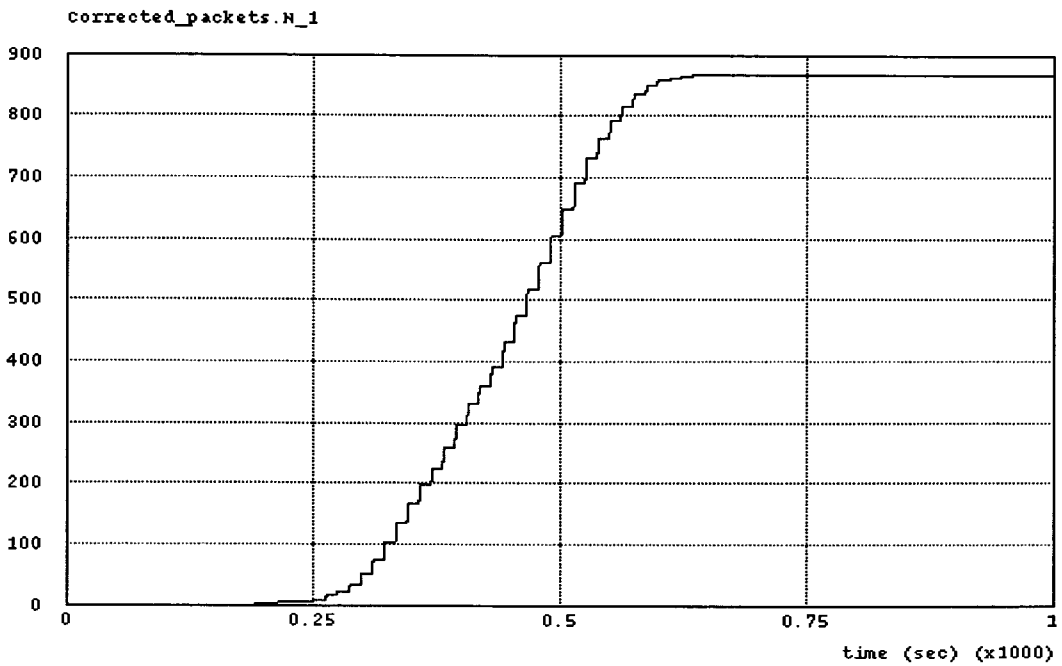

Figure 49: Corrected packets (Time-tolerant service).

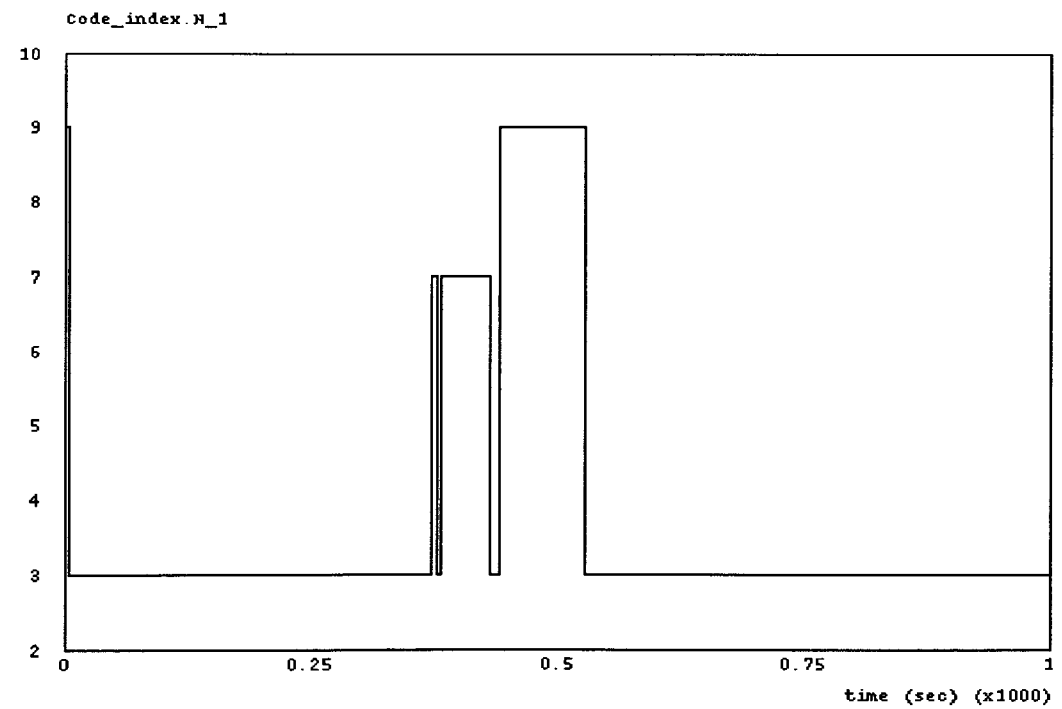

Figure 50: BCH code used (Minimum distance).

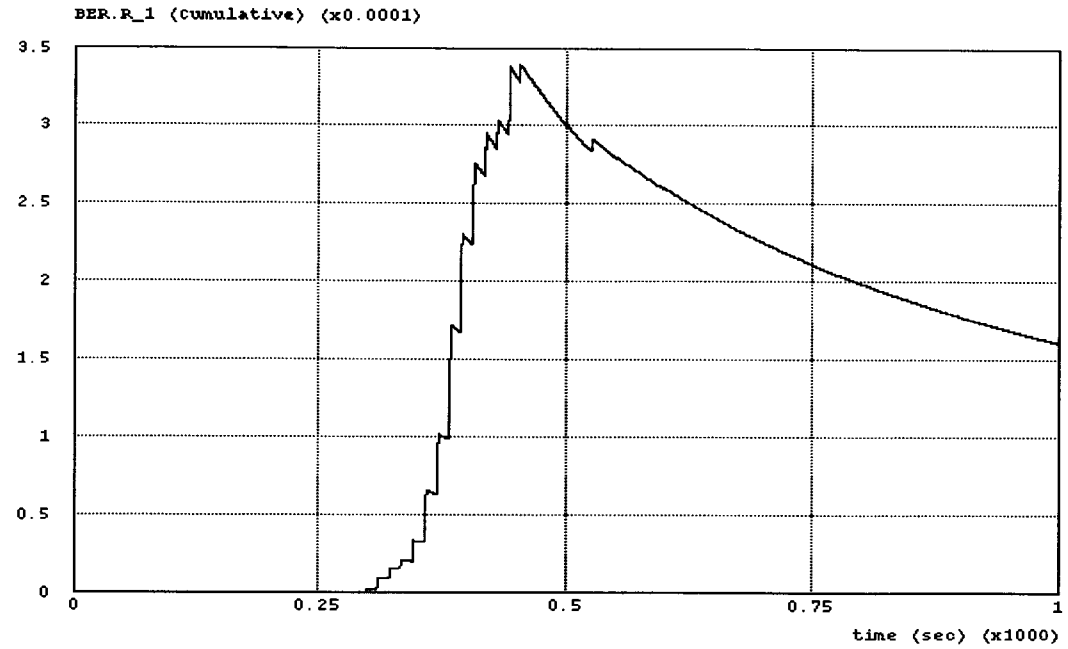

Figure 51: Bit Error Rate (Cumulative-FEC performed).

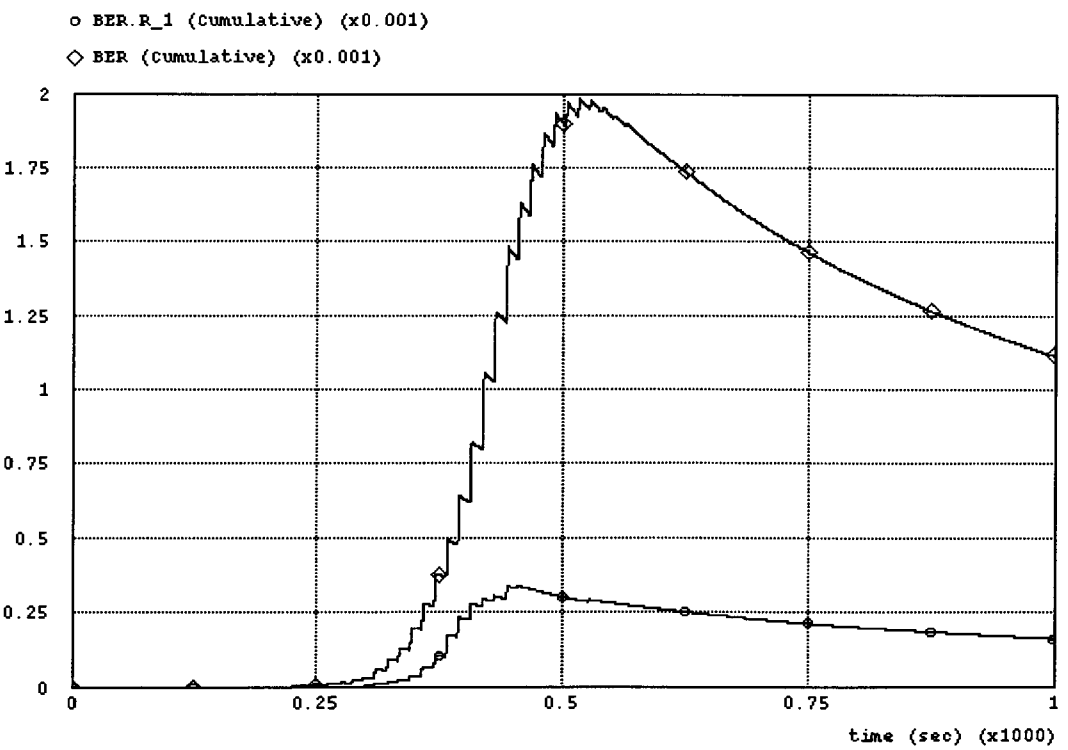

Figure 52: Bit Error Rate (Before-After).

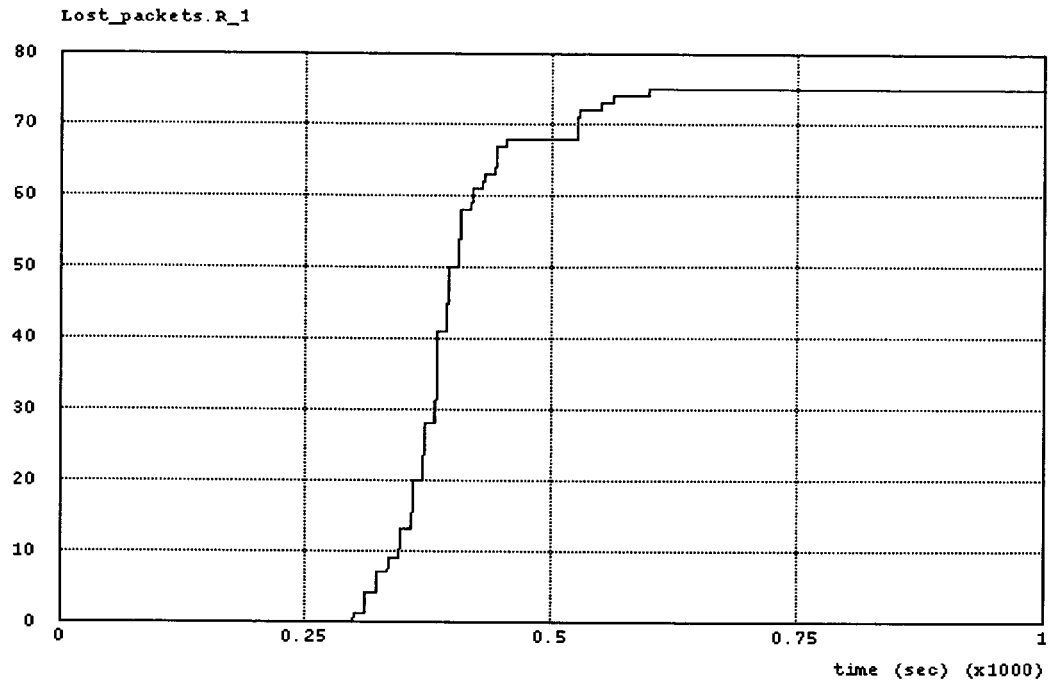

Figure 53: Lost packets (Time-sensitive service).

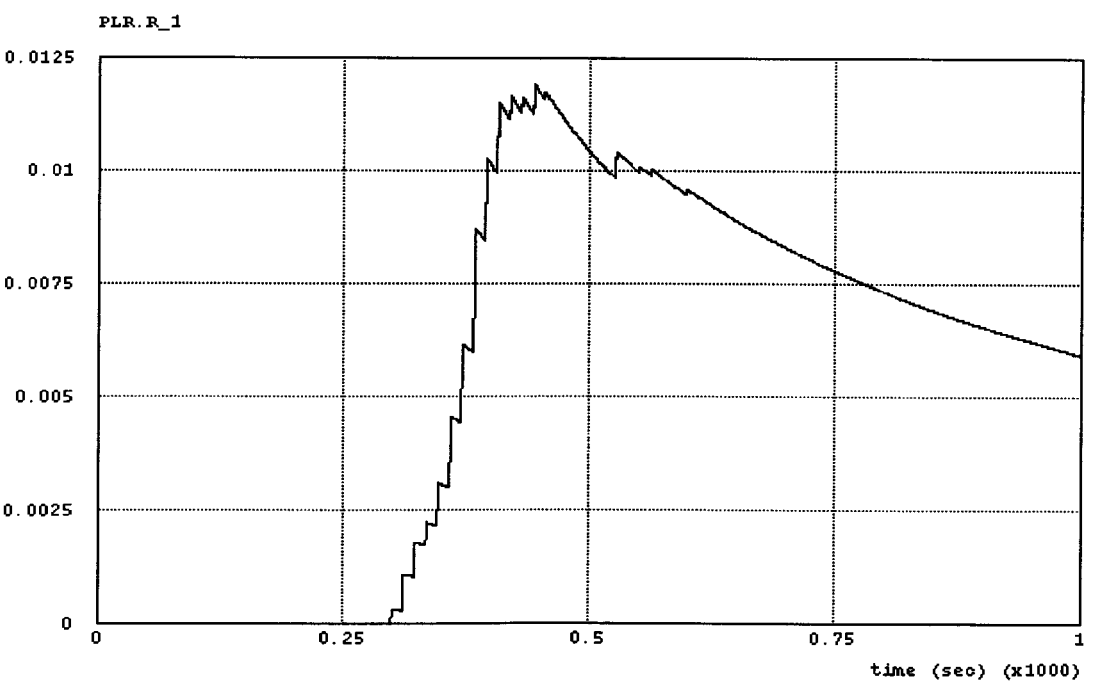

Figure 54: Packet Loss Rate (Time-sensitive service).

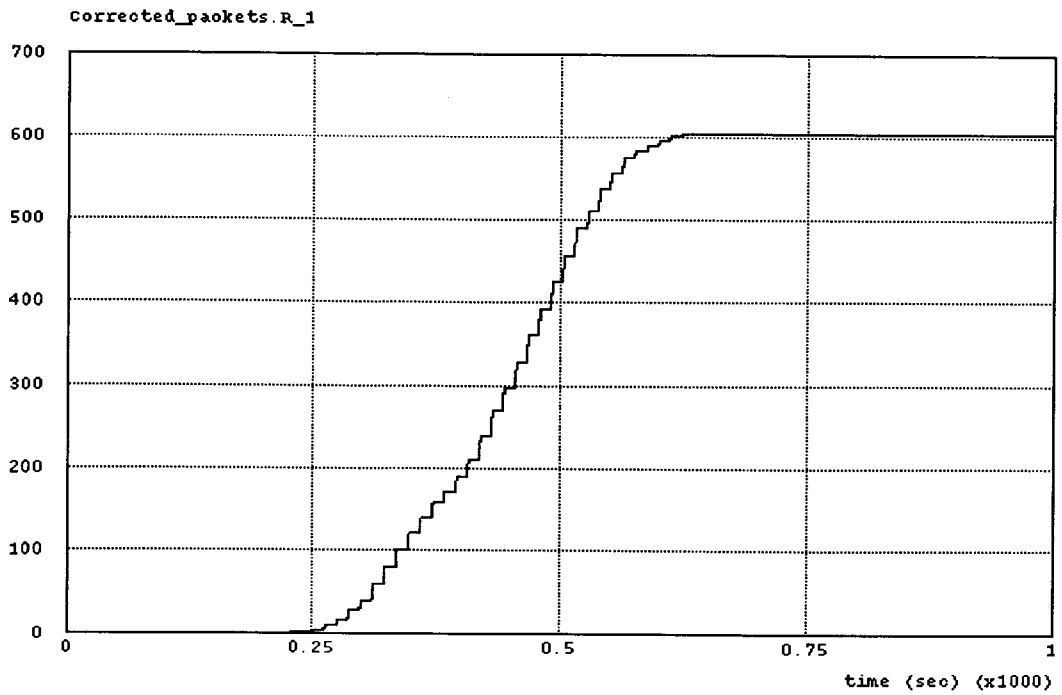

Figure 55: Corrected packets (Time sensitive service).

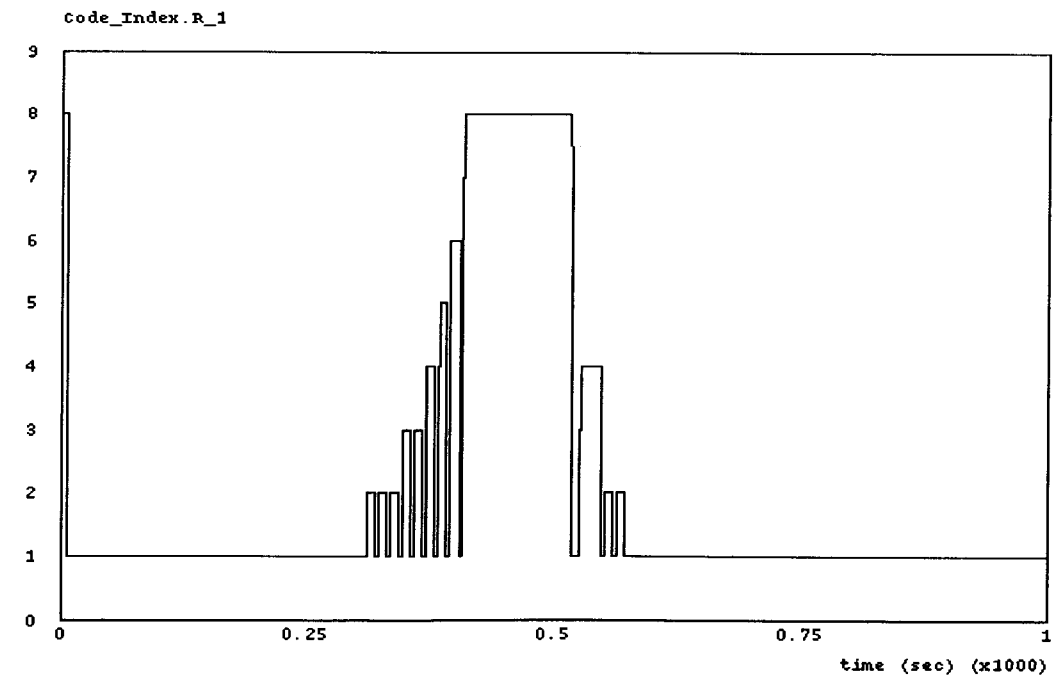

Figure 56: RS code used (Correctable 6-bit words).

5.3.3 The third simulation run

For the third simulation run the power of the jammer node was set at 1.1 Watts, while the power of the base station and the mobile station remained the same.

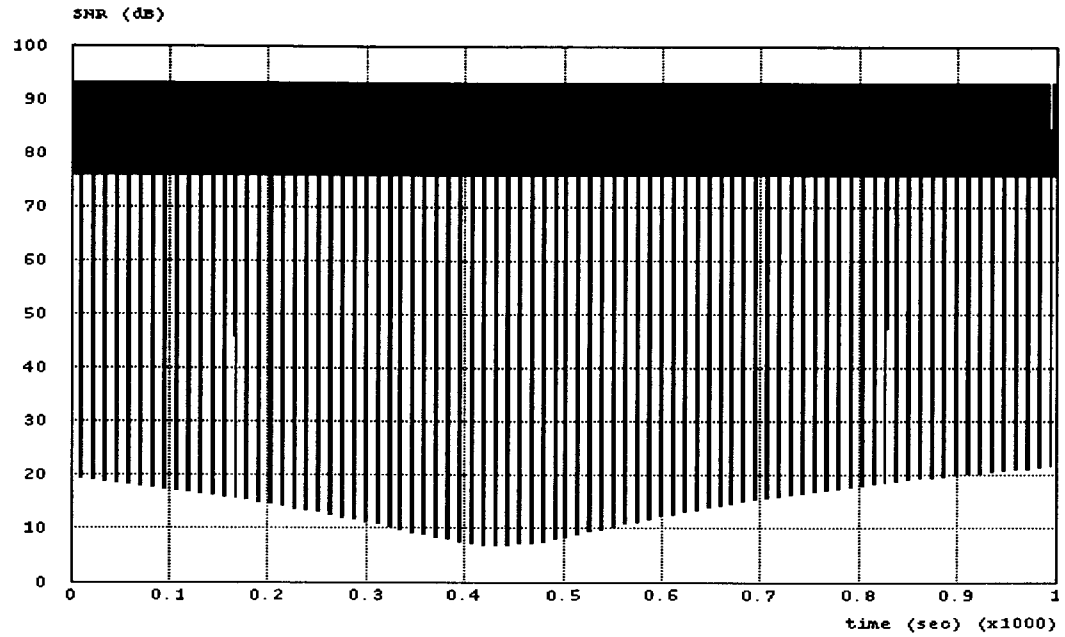

Figure 57: SNR (dB).

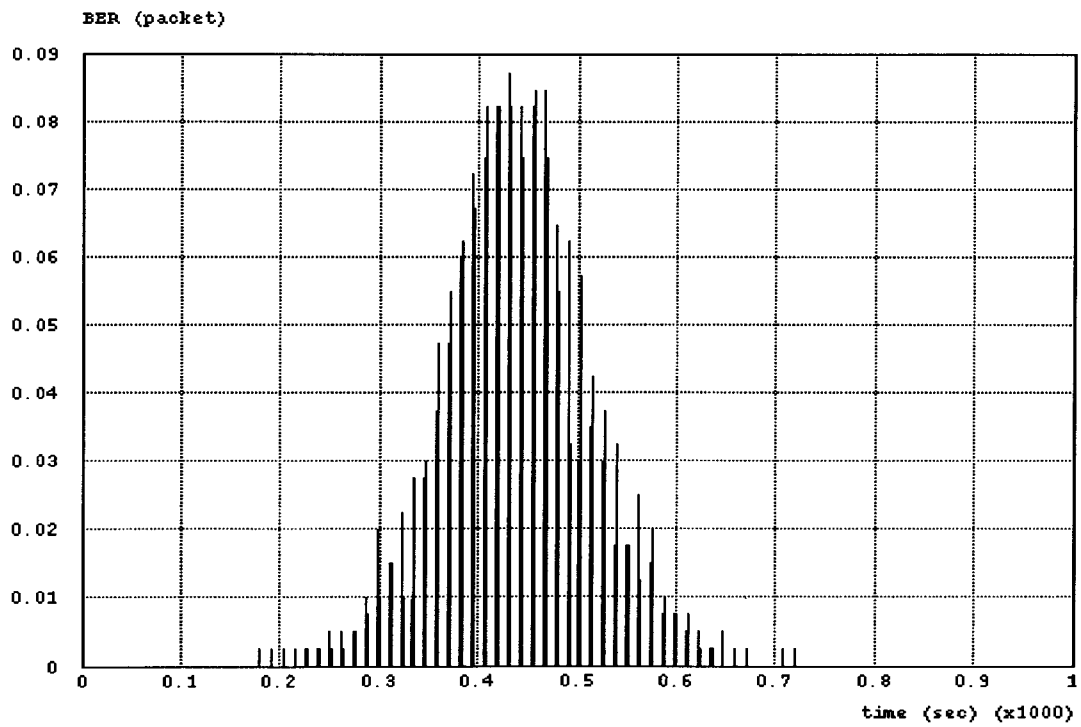

Figure 58: Bit Error Rate (Instantaneous).

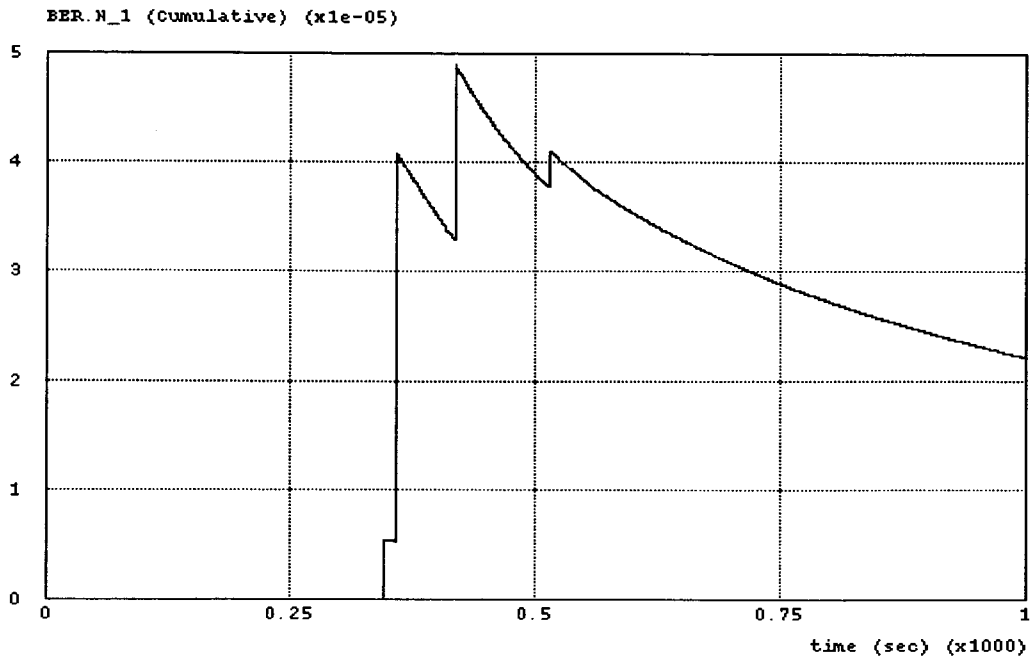

Figure 59: Bit Error Rate (Cumulative-ARQ performed).

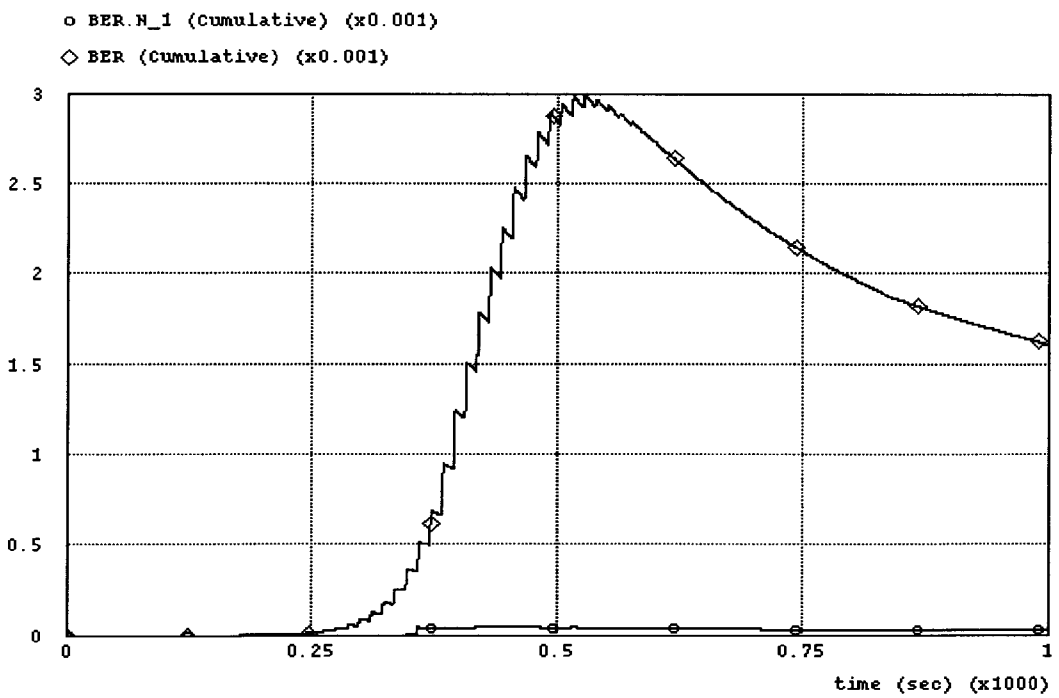

Figure 60: Bit Error Rate (Before-After).

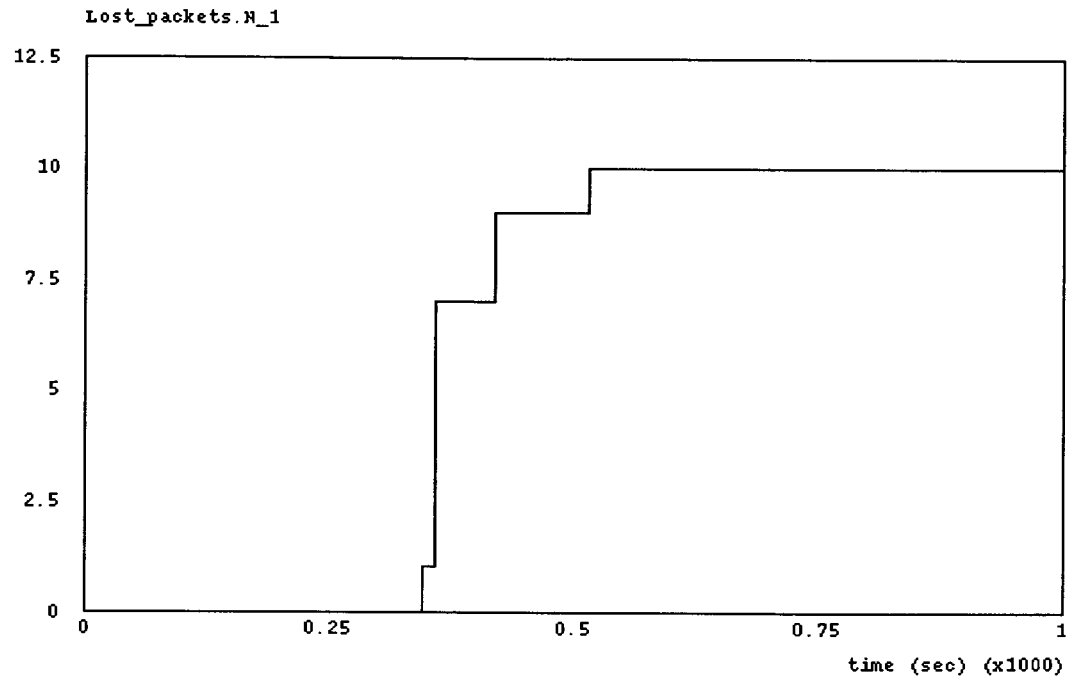

Figure 61: Lost packets (Time-tolerant service).

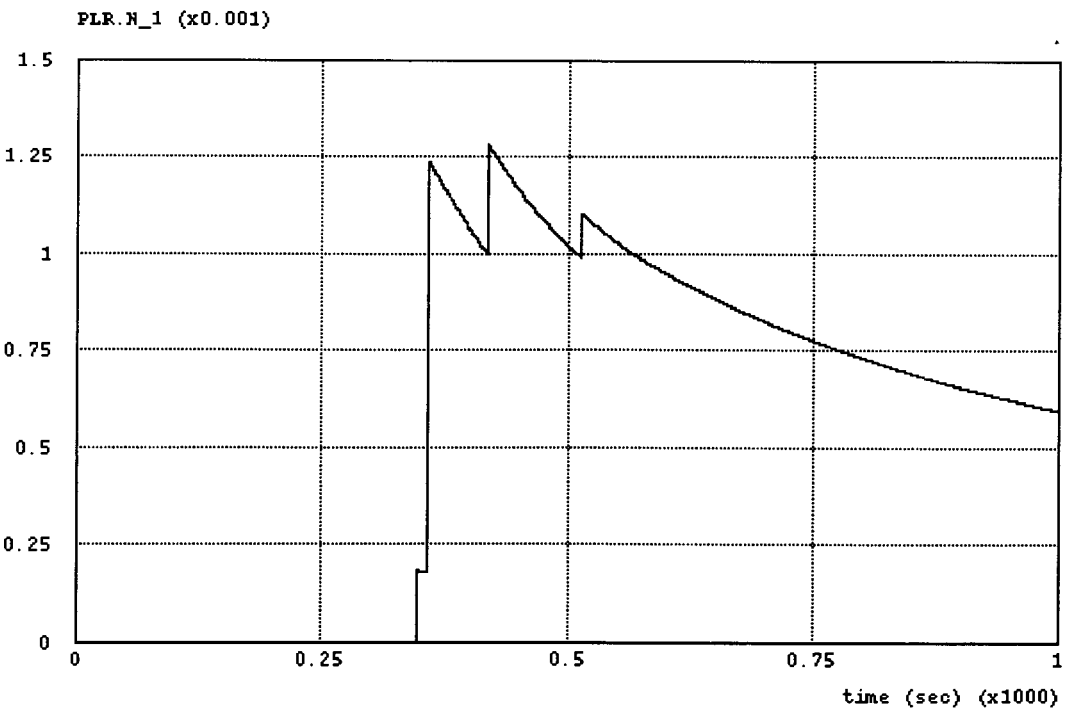

Figure 62: Packet Loss Rate (Time-tolerant case).

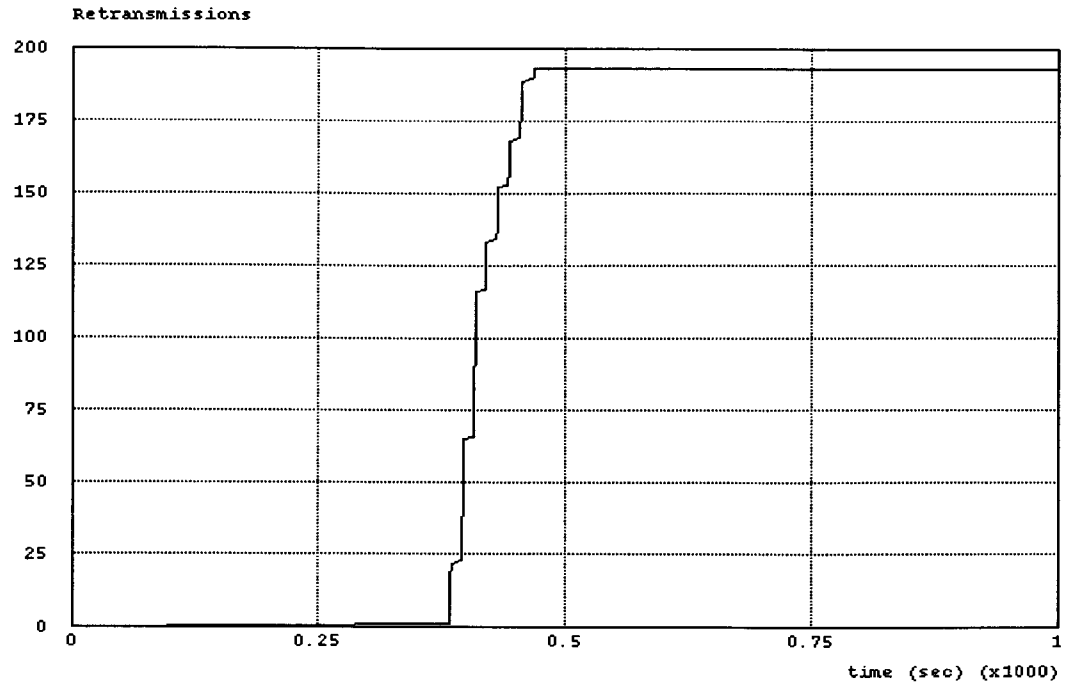

Figure 63: Number of retransmissions.

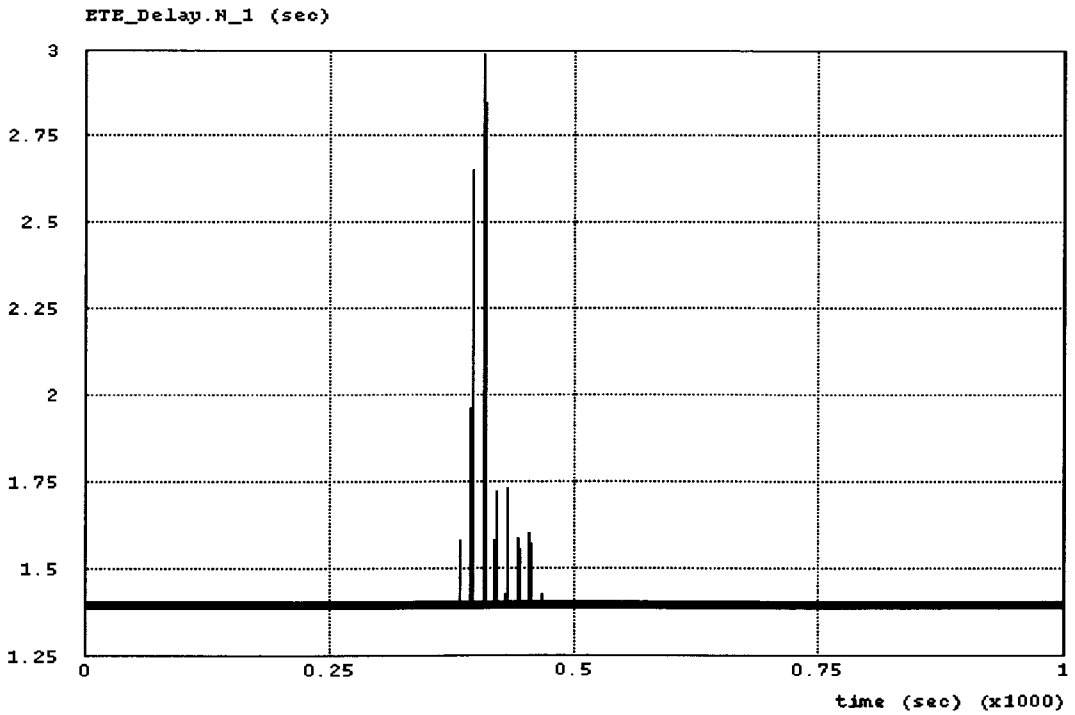

Figure 64: End-To-End Delay (Time-tolerant service).

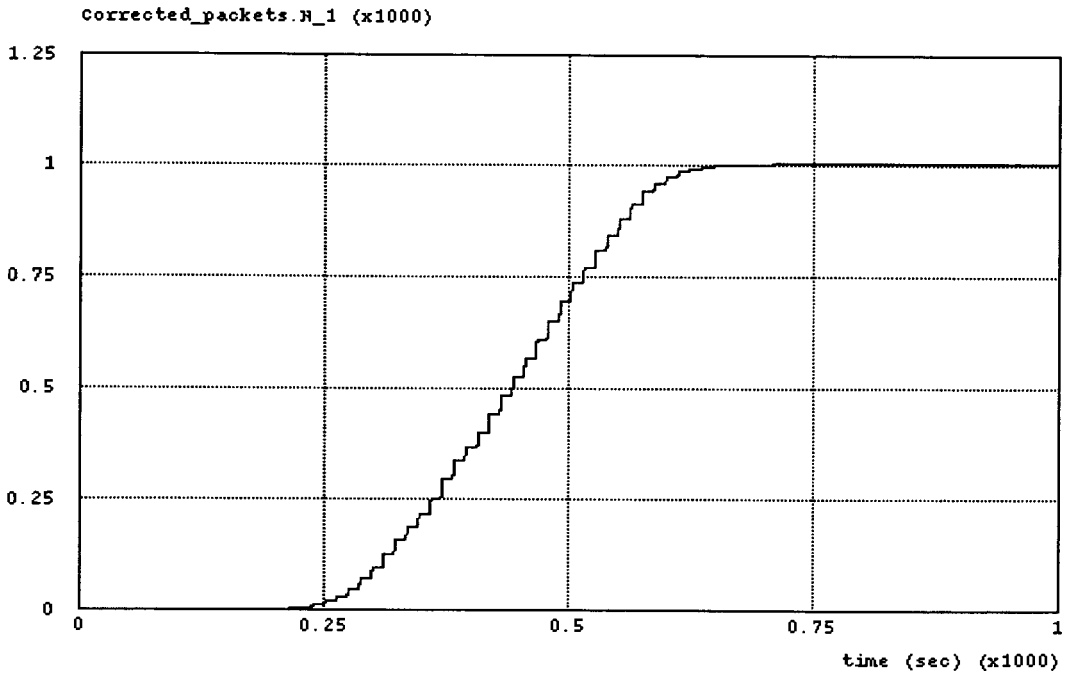

Figure 65: Corrected packets (Time-tolerant service).

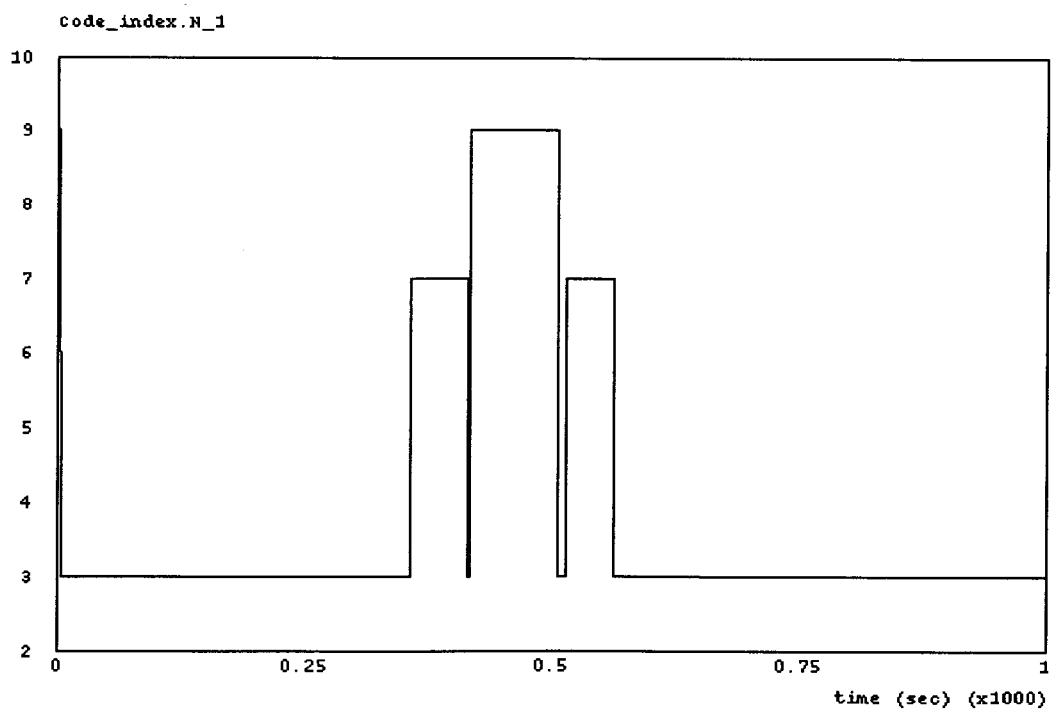

Figure 66: BCH code used (Time-tolerant service).

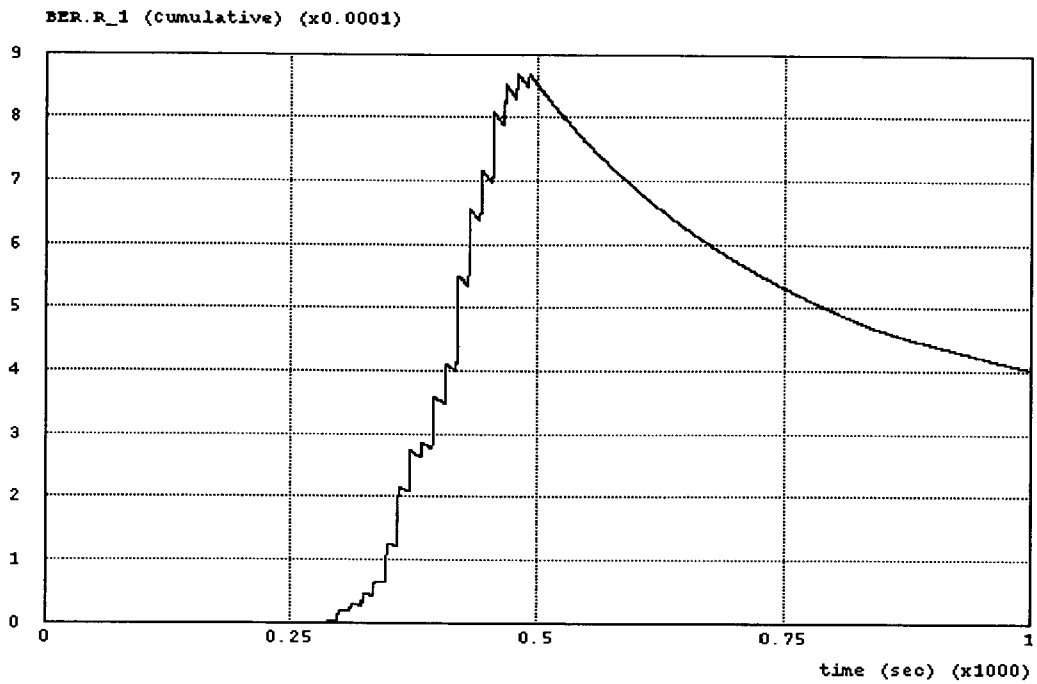

Figure 67: Bit Error Rate (Cumulative-FEC performed).

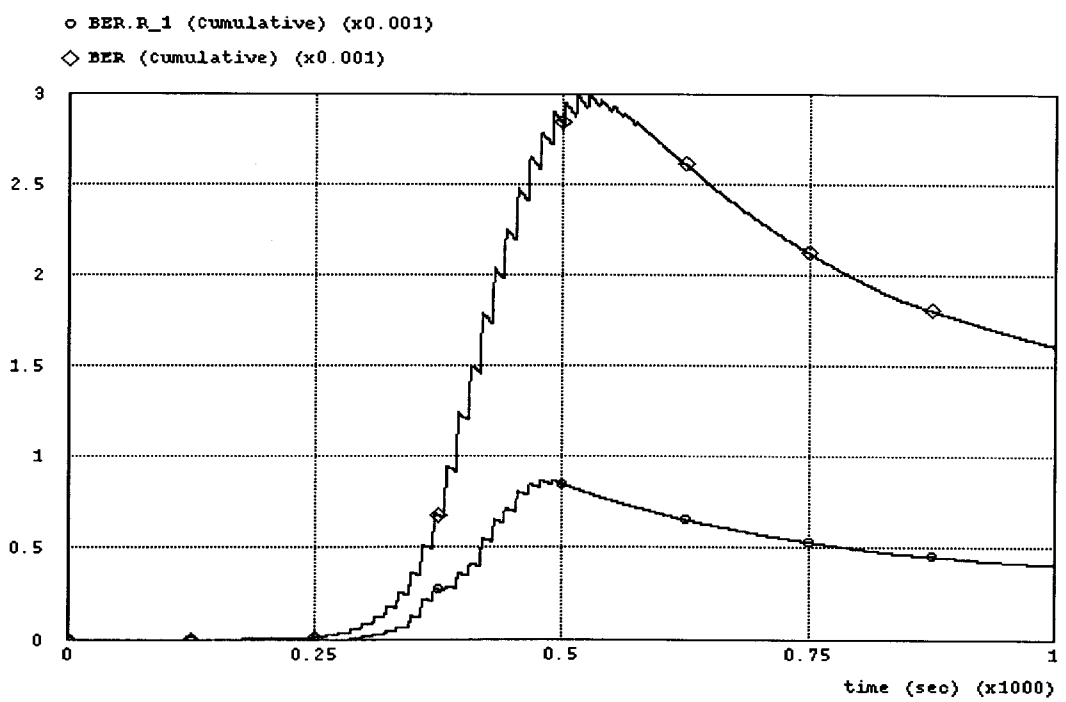

Figure 68: Bit Error Rate (Before-After).

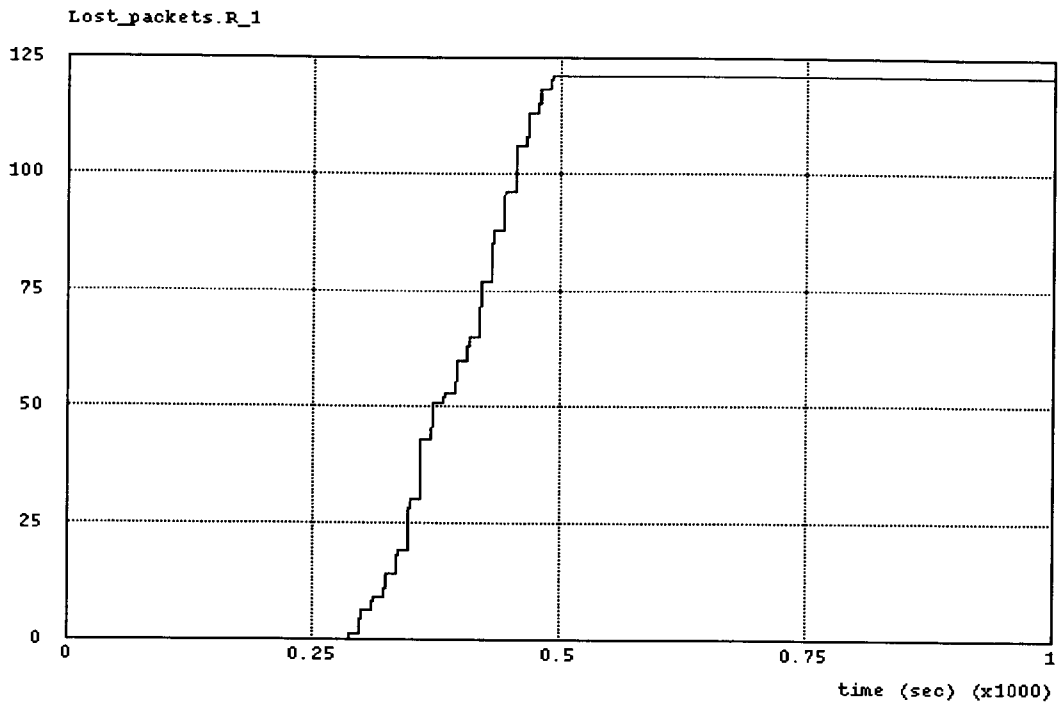

Figure 69: Lost packets (Time-sensitive service).

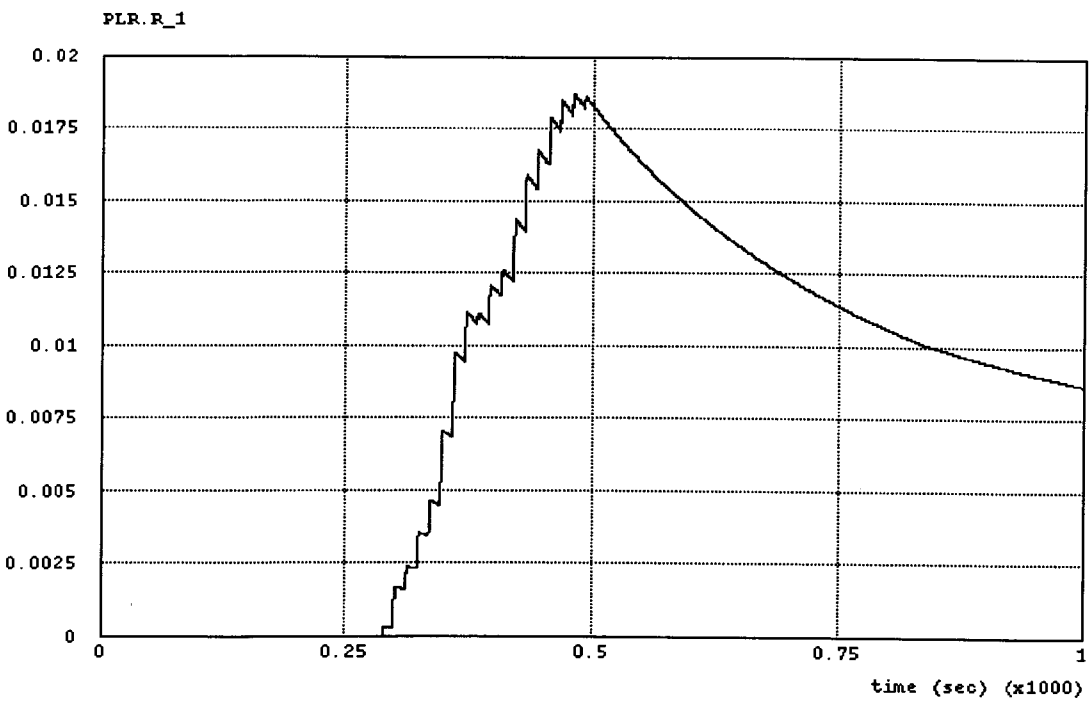

Figure 70: Packet Loss Rate (Time-sensitive service).

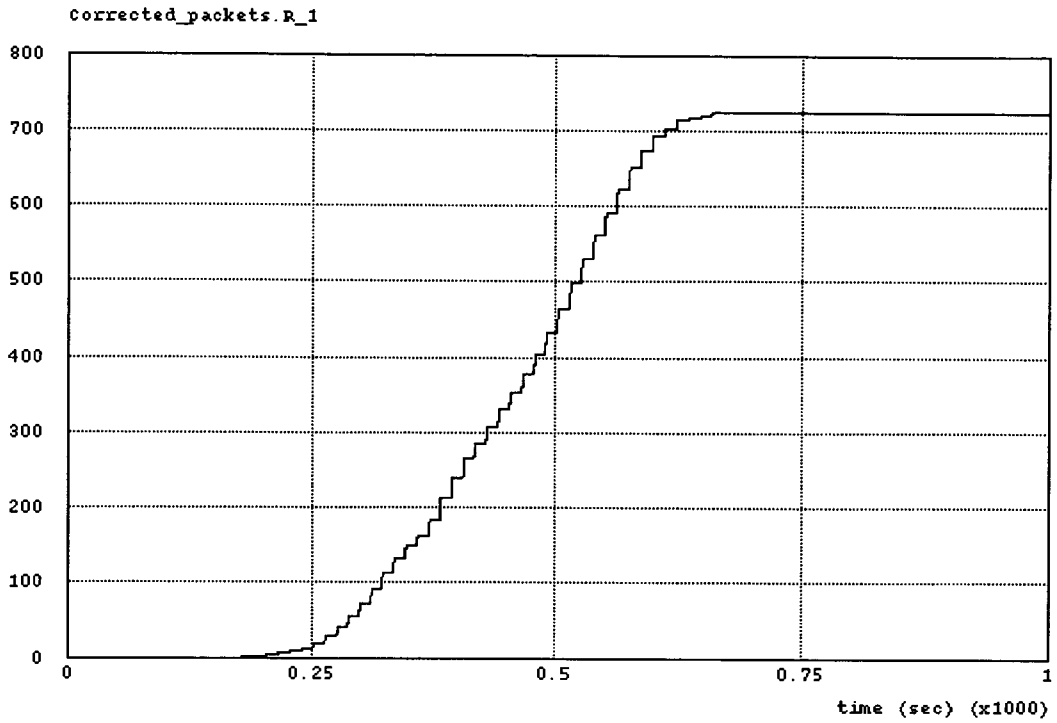

Figure 71: Corrected packets (Time-sensitive service).

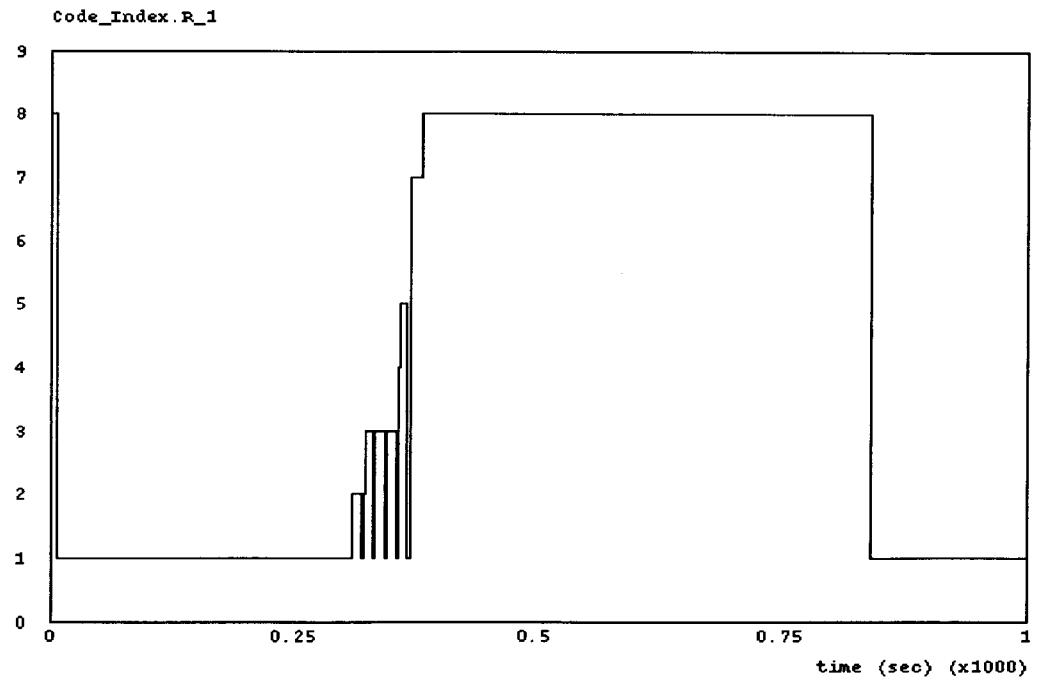

Figure 72: RS code used (Correctable 6-bit words).

5.4 Discussion of the results

5.4.1 The first simulation run.

In the first simulation run we see from Figure 24 a drop in the SNR of approximately 12.4 dB, and a corresponding rise in the bit error rate in the arriving packets (Figure 25). The BER peaks at about 0.055 at time approximately 455.7 seconds. At the time, the jammer node is at the closest point of its path to the two communicating nodes, and corresponds to the time when the lowest value of SNR is also observed.

a) Parameters for the time-tolerant service

From Figure 26 we see that the MSEC has kept the cumulative BER at very low levels, in the order of 10^{-5}, when the BER of the traffic contract is 10^{-4}. For comparison purposes Figure 27 illustrates the achieved BER versus the cumulative BER without error protection, and the benefit of the use of the error control is apparent.

From Figure 28 we see the number of lost packets is kept low, with 8 packets of data lost, while the Packet Loss Rate has exceeded the promised packet loss rate of the connection (10^{-3}) for about 65 seconds, from the time 418 seconds to 483 seconds. This is an important result because it indicates the time interval for which the promised QoS has not been kept. It seems that if all the call was completed within this interval, the overall QoS

would have been violated. However, during this time interval the channel demonstrated remarkably low SNR.

From the same Figure (28) and Figure 33, we can also extract another important parameter of performance: The time elapsed, from the time we violated the QoS contract to the time the MSEC responded by switching to more powerful codes. The time elapsed is ~0.6 seconds, which corresponds to two-way propagation delay in the medium (Assuming the speed of the radio waves in atmosphere approximately c, and a microcell environment with radius 150m, this parameter is ~1μsec), plus the transmission delay for two packets of length 402 bits (approximately 0.04sec). To these times we must also add the time required until the next control packet appears to the receiver (since the bit rate of the control channel is 1kbit/s, and the control packet length is 402 bits, the maximum delay here is the time elapsed between two successive control packets, ~0.45sec), plus processing time. The measured delay of 0.6seconds is in par with the estimated above times, and it is also fast compared with the amount of time the scheme remains in a state of a powerful code.

Figure 30 shows the number of retransmissions occurred during the duration of simulation. For this simulation run we got 4 retransmissions, at times that correspond to the time span of high BER. Note that the number of retransmissions is relatively small, due to the fact that the BCH codes used are also capable to correct a large number of errors as shown in Figure 32. Thus, only packets with errors more than the correcting ability of the code, and less of equal to the detecting ability of the code are actually retransmitted.

Figure 31 shows the End-To-End delay of the packets. In this Figure, we can see that the delay experienced by packets of the time-tolerant service is increased in comparison to the delay experienced by the packets of the real time connection (See Figure 40). This is attributed to the delay caused by the interleaving of the time-tolerant service packets. In addition, we can see the additional delay experienced by packets that are being retransmitted, reaching a maximum of 1.5 seconds. This delay is well within the QoS end-to-end delay for the connection shown in table 4. In this figure (31) we also observe that we have variable delay even for packets that are not retransmitted (This is the explanation for the "black strip" appearing at the bottom of the figure). This is attributed to the segmentation of the incoming ATM cells: As ATM cell arrive from the wireline network, they are segmented, and air interface packets are formed for transmission over the wireless link. However, the air interface packets are smaller than the ATM cells, so inevitably there will be remaining bits in the segmentation buffer, which accumulate. At some point the remaining bits suffice for the formation of an air interface packet. At this point, two air interface packets are ready for transmission simultaneously: The normally formed packet, and the packet formed from the remaining bits in the buffer. The latter will experience additional delay equal to one transmission time (until the normally formed packet completes its transmission). In our case, the link speed is set to 19.2 kbits/s, and the packet length is 402 bits, so the transmission delay is ~0.0209 sec, exactly the width of the horizontal "bar". The frequency with which this effect occurs depends on the codes used, because more powerful codes use less information bits, thus more bits are left in the buffer, and the additional air interface packets are formed more frequently.

Figure 32 depicts the number of corrected packets for the time-tolerant connection. The total number is 688 packets, a big number compared to the number of retransmissions, which is just 4, and it demonstrates the power of the BCH codes employed.

Figure 33 shows the type of BCH code used at any given time. We see that for the most time, the least powerful code is used, thus the packets have minimum overhead and the throughput is increased. We can also see that the scheme adapts relatively fast (delay is about 0.6 seconds, as already described) to a more powerful code once the QoS contract has been violated. In this particular case, the maximum code used has minimum distance 7, since the algorithm used for the transitions finds the code sufficient to handle the errors in the incoming packets. Having a look at figures 26 and 29 for the packet loss rate and the bit error rate, we see that indeed there is no need for more parity check bits in the packet. Clearly this approach demonstrates the focus on efficiency the error control scheme has.

b) Parameters for the time-sensitive service

Figure 34 shows the achieved Bit Error Rate for the time-sensitive service. The achieved BER is well below the BER of the traffic contract at all times, which means that all the transitions to different codes will be a result of the Packet Loss Rate parameter. Also, for illustrative purposes, Figure 35 depicts the BER before and after the application of the FEC. An immediate result from this figure, and Figure 39 is that the RS codes used are

not as powerful as the BCH codes used in the time-tolerant case, since the achieved BER is now higher.

Figure 36 shows the number of lost packets, which reaches a total of 69 packets, again showing that the RS codes used are not as powerful as the BCH codes. However, usually real-time services have higher tolerance of errors [27], and the scheme can be easily expanded with additional RS codes.

Figure 37 depicts the achieved Packet Loss Rate, which has a rather peculiar and interesting property: The PLR actually intersects the line of the PLR of the QoS contract (table 4), which is 10^{-2} several times. From this graph we can expect to see the MSEC switching to more or less powerful codes during this time interval, demonstrating a "hunting" effect. Indeed, if we look at Figure 39, which shows the RS code used over time, we see that the MSEC switches back and forth to less and more powerful codes depending if the PLR of the QoS contract is maintained or not. This result demonstrates how fast and how reliably the MSEC makes transitions over time. In Figure 39 we also see that the MSEC finally rests to the least powerful code at time approximately 547 seconds, where the PLR is crossing the QoS line for the last time, and the jammer is moving away generating less noise. From that time on, to the end of the simulation, the scheme is using the least powerful code, and thus it increases the throughput.

Figure 38 shows the number of corrected packets of the time-sensitive case. The total number of the corrected packets is 446, a number significantly lower than the number of the corrected packets in the time-tolerant case. This can be attributed to two reasons: First, the real-time connection has a lower bit rate than the time-tolerant connection (table

4), so the absolute number of erroneously received packets is smaller. Second, in the real-time connection we lost many more packets than in the time-tolerant connection (69 versus 8).

Finally, Figure 40 shows the End-To-End delay for the packets of the real-time connection, which is split into two levels. This effect is caused by the segmentation taking place at the base station. The incoming ATM cells, after their headers are stripped, are inserted into a segmentation buffer, where only part of the initial cell is used to form the radio link packet, and the rest of the packet remains in the buffer until another ATM cell arrives, and another radio link packet can be created and sent over the radio link. At some point the remaining bits suffice for the formation of an air interface packet. At this point, two air interface packets are ready for transmission simultaneously: The normally formed packet, and the packet formed from the remaining bits in the buffer. The latter will experience additional delay equal to one transmission time (until the normally formed packet completes its transmission). In our case, the link speed is set to 19.2 kbits/s, and the packet length is 402 bits, so the transmission delay is ~0.0209 sec, exactly the width of the horizontal "bar". The frequency with which this effect occurs depends on the codes used, because more powerful codes use less information bits, thus more bits are left in the buffer, and the additional air interface packets are formed more frequently.

It is worth noting that the ETE delay for the real-time service is the same for all the simulation runs, so it does not appear in the results of the remaining two simulation runs.

5.4.2 The second simulation run.

The results for the second simulation run are shown in figures 41 through 56.

The results are of the same nature, with the SNR and the BER being more severe as the jammer node approaches the communicating nodes. In particular, as shown in Figure 41, the SNR drops from 20.4dB at the time is at the farthest point from the base station and the mobile, to just 8.0dB at the time it is at the closest point. The SNR drop is 12.4dB, the same drop as in the first simulation run, but at a lower level. The corresponding BER is shown in Figure 42, and it peaks at time 443 seconds at a value of 0.07. Again we note the severity of the observed BER for a large time interval. In this figure we observe a sequence of vertical "bars". These bars correspond to the time intervals when the jammer is "on" and generates co-channel interference. At these time intervals the SNR drops dramatically. Another observation in this figure is the "black strip" at the upper part of the figure. It appears that even when the jammer is "off" the SNR varies with time (We have interconnected points representing the SNR values at various times, that form the "black strip"). Indeed, OPNET has a built-in function in the radio pipeline stages (radio link simulation), which calculates the background noise, and adds it to the co-channel interference (generated by the jammer) to estimate the overall SNR. The variation in the background noise creates the effect of the "black strip".

Basic discussion of the results follows the discussion made in the case of the first simulation run, the focus here will be on the points where the effects of the increased BER are felt.

a) Parameters for the time-tolerant service

The cumulative achieved BER, and its comparison with the cumulative BER before the application of the error control scheme is shown in figures 43 and 44. As before, the achieved BER is well below the promised QoS value of the BER, and well below the BER before the application of error correction. The switching algorithm will be based on the value of the Packet Loss Rate again. The number of lost packets in this case is increased to 9 as indicated in Figure 45, more than before as expected.

The Packet Loss Rate of the connection is depicted in Figure 46. Here we see that the promised PLR of 10^{-3} is violated for a longer period of time. Specifically, the PLR is violated from the time 370sec to 380sec, 384sec to 434sec, and from 442sec to 528sec. The appearance of the top part of the figure follows the bursts the jammer node generates. The jammer node is idle for 10 seconds, and then it is activated for two seconds. From the time 528sec on to the end of the simulation, the PLR is below the promised value of 10^{-3}. Apparently, this result suggests that we may not provide the QoS promised if the connection is terminated within these intervals.

The number of retransmissions in this case is up to 54, as shown in Figure 47, considerably increased from the four retransmissions of the first simulation.

Figure 48 shows the End-To-End delay measured at the mobile, showing the increased delay for the packets that were retransmitted. The peak delay is measured at 432sec, and has a value of 2.016sec, which is well below the delay of the QoS contract.

The number of corrected packets is measured at 866, higher than before (Figure 49).

The BCH codes used are shown in Figure 50. Here we see the most powerful code is used for a longer period of time, from 442sec to 528.6sec. If we compare this result with the change of the PLR, we see that the adaptation time of the MSEC for shifts to lower codes is ~0.6sec, a result consistent with the result for shifts to higher codes.

b) Parameters for the time-sensitive service

The cumulative achieved BER, and its comparison to the BER before the application of the RS codes are shown in figures 51 and 52. Both figures show that the achieved BER is well below the respective QoS parameter, so the switching algorithm depends on the value of the PLR parameter.

The number of lost packets is shown in Figure 53, with a maximum of 75 packets at simulation time 599sec.

Figure 54 shows the Packet Loss Rate achieved by the connection. The respective QoS parameter is violated for a time span of approximately 68 seconds, in the following intervals:

From 395sec to 404sec, from 406sec to 518sec, from 526 to 558 sec, and from 565sec to 570sec. If the connection is terminated in on of these intervals, the MSEC fails to maintain the QoS promised, otherwise it is guaranteed.

Figure 55 shows the number of corrected packets of the connection. It reaches a maximum of 604 packets.

The RS code used in the connection is shown in Figure 56. Here again we see that the MSEC is switching to various codes following the fluctuations of the PLR parameter (Figure 54), with respect to the QoS value of the PLR.

5.4.3 The third simulation run.

The results for the third simulation run are shown in figures 57 through 72.

In this simulation run, the capabilities of the MSEC scheme are stretched to their limits. In particular, the achieved PLR will be violated for hundreds of seconds, which is inappropriate in ATM standards.

As shown in Figure 57, the SNR drops from 19.4dB, to a very low 7.0dB at the time the jammer is at the closest point to the communicating nodes. The SNR drop is 12.4dB, the same drop as in the other simulations, but at a lower level. The corresponding BER is shown in Figure 58, and it peaks at time 430.3 seconds at a value of 0.087. Again we note the severity of the observed BER for a large time interval.

a) Parameters for the time-tolerant service

The cumulative achieved BER, and its comparison with the cumulative BER before the application of the error control scheme are shown in figures 59, and 60. As before, the achieved BER is well below the promised QoS value of the BER, and well below the value of the BER before the application of error correction. The switching algorithm will be based on the value of the Packet Loss Rate again.

The Packet Loss Rate of the connection is depicted in Figure 62. Here we see that the promised PLR of 10^{-3} is violated for a longer period of time. Specifically, the PLR is violated from time 358sec to 414sec, from 418sec to 504sec, and from 513sec to 560sec. From the time 560sec on to the end of the simulation, the PLR is below the promised

value of 10^{-3}. Apparently, this result suggests that we may not provide the QoS promised if the connection is terminated within these intervals.

The number of retransmissions in this case is up to 193, as shown in Figure 63, considerably increased from the retransmissions of the other simulations.

Figure 64 shows the End-To-End delay measured at the mobile, showing the increased delay for the packets that were retransmitted. The peak delay is measured at 408sec, and has a value of 2.99sec, which is well below the delay of the QoS contract. The difference in the maximum ETE delay between the three simulations is attributed to multiple retransmissions of the same packet(s).

The number of corrected packets is measured at 1005, higher than before (Figure 65).

The BCH codes used are shown in Figure 66. Here we see the most powerful code is used for a period of time, from 418sec to 508sec, while other codes are following the pattern of the PLR plot (Figure 62).

b) Parameters for the time-sensitive service

The cumulative achieved BER, and its comparison to the BER before the application of the RS codes are shown in figures 67 and 68. Both figures show that the achieved BER is well below the respective QoS parameter, so the switching algorithm depends on the value of the PLR parameter.

The number of lost packets is shown in Figure 69, with a maximum of 121 packets at simulation time 492sec.

Figure 70 shows the Packet Loss Rate achieved by the connection. The respective QoS parameter is violated for a time span of approximately 512 seconds, in the interval from 370sec to 842sec. If the connection is terminated in this interval, the MSEC fails to maintain the QoS promised, otherwise, it is guaranteed. Here we have to note that this is not a satisfying result, and due to the stochastic nature of the duration of the call, the probability of not meeting the QoS is really high, thus this is the "upper limit" of the performance of the MSEC.

Figure 71 shows the number of corrected packets of the connection. It reaches a maximum of 724 packets.

The RS code used in the connection is shown in Figure 72. Here again we see that the MSEC is switching to various codes following the fluctuations of the PLR parameter (Figure 70), with respect to the QoS value of the PLR.

5.5 Benefit from the use of adaptive codes

In the previous sections we presented the results we obtained from the computer simulation of the MSEC scheme. The MSEC uses a set of codes in order to select the appropriate ones at any given time and increase the average throughput. Figures 33, 50, and 66 show the BCH codes used in the time-tolerant service, and Figures 39, 56, and 72 show the RS codes used in the time-sensitive service throughout the simulation. To assess the usefulness of such a scheme, it is helpful to calculate analytically the achieved

throughput, and compare it to a scenario where we use non-adaptive codes. In this case we are interested in the data throughput, since the benefit of adaptivity lies in the use of the appropriate amount of error control bits. For this purpose we calculate the data throughput of all the RS and BCH codes used in the MSEC. The results are presented in the tables below:

d_{min}	Length	Parity Bits	Overhead	Total Overhead	Data Bits	Max. Throughput
3	402	21	42	63	339	0.84
5	402	35	42	77	325	0.81
7	402	49	42	91	311	0.77
9	402	63	42	105	297	0.74

Table 5: Maximum throughput of the BCH codes.

Correctable Symbols	Length	Parity Bits	Overhead	Total Overhead	Data Bits	Max. Throughput
1	402	12	42	54	348	0.87
2	402	24	42	66	336	0.84
3	402	36	42	78	324	0.81
4	402	48	42	90	312	0.78
5	402	60	42	102	300	0.75
6	402	72	42	114	288	0.72
7	402	84	42	126	276	0.69
8	402	96	42	138	264	0.66

Table 6: Maximum throughput of the RS codes.

In the above tables, the column labeled "Parity Bits", represents the bits used for error detection/correction, the column " Overhead" represents the bits of the header (not

containing information, the column "Total Overhead" represents the sum of the parity bits and the overhead bits (This is the total number of non-data bits). Finally, the column "Max. Throughput" represents the maximum throughput that can be achieved (That is all the errors can be corrected).

A fair comparison with non-adaptive codes must assume a code that can be as effective as the codes used in the MSEC in a given situation. This means that the code selected for comparison should be able to correct and/or detect the errors that the MSEC does. We will demonstrate a comparison, in the case of the second simulation run, for the time-sensitive service. The RS codes used in this case are shown in figure 56. From this figure, we find that during the 1000 seconds of the simulation, the MSEC operates as follows: For 795 seconds using the code 1 (Correctable symbol), for 38 seconds using the code 2, for 16 seconds using the code 3, for 30 seconds using the code 4, for 6 seconds using the code 5, for 7 seconds using the code 6, and for 108 seconds using the code 8. To estimate the average throughput for the entire duration of the connection, we use these times the contents of table 6. We also need to make the assumption that we measure the throughput on the correctly received packets only. This assumption is necessary for two reasons: First, the lost packets are within the acceptable limits of the QoS contract, and second it is hard to find a single code that will lose the exact same number of packets in identical conditions to get a fair comparison. Given this assumption the average data throughput for the entire duration of the connection can be estimated:

$$T = \frac{(38 * 0.84) + (16 * 0.81) + (30 * 0.78) + (6 * 0.75) + (7 * 0.72) + (108 * 0.66) + (795 * 0.87)}{1000} \cong 0.84 \qquad (14)$$

If we compare this value with the values presented in Table 6, we see that in essence we achieve the throughput of the code 2 for the duration of the connection. This result demonstrates the usefulness of the adaptation in the MSEC scheme. In contrast, if we used a non-adaptive code capable of maintaining the QoS parameters, we would select one of the more powerful ones for the entire duration of the connection, as can be seen from Figure 53: Even when we use more powerful codes such as 5, 6, or 8, we still lose packets. To guarantee the maintenance of the QoS contract we should employ one of the more powerful codes, such as 7, or 8. From Table 6 we can see that the respective maximum throughputs for these two codes are 0.69, and 0.66, which is a poor performance in comparison with the 0.84 achieved with the MSEC.

However, this comparison may be superficial since error control in wireless ATM must be able to support all types of services with diverse QoS requirements, including signaling and control channels with requirements for BER in the order of 10^{-8} (Table 1). A single non-adaptive code must use enough parity check bits to guarantee this requirement, as well as services with lower BER requirements (e.g. 10^{-4}), and in this case the throughput of the service would be significantly reduced.

Finally, similar results are obtained if we compare non-adaptive codes to the MSEC in other error conditions.

CHAPTER 6

CONCLUSIONS

6.1 Performance at various BER situations

The simulation results presented in the previous section demonstrated the performance of the MSEC scheme under harsh error conditions. For the most part, the results show that the MSEC is capable of handling severely errored packets, and of maintaining the QoS contract in most of the cases. The results presented refer to situations with very low SNR, as low as 7dB. Given the modulation scheme used (QPSK), the corresponding BER was as high as 8.7×10^{-2}, certainly representing extremely harsh operating environment. The MSEC was also tested for situations with higher SNR and corresponding lower BER, and the results proved to be very good. For all cases the codes employed were the lower of the code suite, and the achieved BER and PLR easily met the QoS contract. Problems with the QoS guarantees started when the power of transmission of the jammer was increased to 700mW, a situation corresponding to the first simulation run. From that point on several simulations were run, at various

levels of transmission power of the jammer, up to the value of 1200mW, where the MSEC scheme crashed, using the most powerful codes for both kinds of services, but yet was not able to meet the QoS standards of performance. In a pure ATM network, such situation would result in the breaking of the connection. However, the presented results suggest that the limits of the MSEC scheme are lower than that, in the form the MSEC is implemented now. In the last simulation, the PLR parameter was consistently violated for about 500 seconds, which is not acceptable given the stochastic nature of the duration of the call. For a reliable and persistent guarantee of QoS, the QoS parameters should not be exceeded for prolonged periods of time. However, for lower BER situations the MSEC performed up to the expectations, utilizing the available bandwidth efficiently and providing QoS guarantees.

Also from the presented results, we can conclude that time-tolerant services are easier to hold their QoS parameters than the time-sensitive ones. This phenomenon can be attributed to the BCH codes used in the MSEC. They are powerful codes, able to correct more errors per packet than the RS codes. In addition, for the time-tolerant services we can afford interleaving of the packets, thus spreading the errors in a number of packets, and thus "randomizing" them, so the BCH codes can be more effective. Furthermore, for time-tolerant services we use retransmissions for packets we are unable to correct. Time-sensitive services cannot benefit from these techniques, and only FEC is used. At the end, we see that the time-sensitive services unlikely to meet the QoS objectives.

6.2 Shortcomings

At the present time the switching algorithm of the MSEC scheme is quite simple: If one or more QoS parameters are violated, we move to a more powerful code, and if the QoS is met, we move to a less powerful code to increase the throughput. Although the MSEC performs satisfactorily even in harsh conditions, there are some problems associated with the switching algorithm, which are apparent for high BER situations. When the switching algorithm decides to switch to a more powerful code, the QoS has already been violated. In situations of manageable BER, the MSEC can recover relatively fast from the impairment, as can be seen in figures 29 and 37 in chapter 5. Therefore the QoS contract is not seriously jeopardized. However, for higher BER situations the recovery is not as fast (see figures 46 and 54), or even not possible (figures 62 and 70), and the QoS is nor guaranteed, depending on the duration of the call.

The MSEC could benefit from a more "intelligent" algorithm, which could take into consideration the ascending trends in the (cumulative) PLR and BER diagrams (see figures 76 and 70), and switch to more powerful codes before the QoS contract is violated. The same algorithm would be used in the decisions for switching to less powerful codes if the QoS contract has been kept for sometime. Such an algorithm could probably guarantee the QoS contract in situations where now the MSEC cannot, because now for the most part once the QoS is violated, the MSEC has the hard task to reverse a situation already severe.

6.3 Further work

The MSEC as described above works satisfactorily for the majority of the cases, and in conditions of high BER. However, as noted in the previous section, the scheme can benefit from a more efficient switching algorithm. The development and testing of such an algorithm could be an interesting topic of research, and it would be a definite enhancement for the MSEC scheme.

In addition, another important aspect of the implementation of wireless ATM is the correct identification of the VCI/VPI field of the air interface packet. Since the base station is responsible for the allocation of the VCI/VPI values, it would be useful to develop an algorithm which would allocate VCI/VPI values to connections to the same mobile station, such as their Hamming distance would be as large as possible. Thus, the mobile receiver would identify the connection easier, and in the case of the MSEC, the Golay code protecting the VCI/VPI field of the packet would have to choose only between a few codewords with large Hamming distance between them.

Further from the enhancements to the MSEC itself, it would be interesting to work towards the integration of the MSEC in other related work. For example, it would be interesting to measure the delay and PLR for networks in which we have handoffs (with an appropriate handoff mechanism suitable for wireless ATM, as in [1]).

In addition, since the work developed in this thesis is focusing on the second layer of the OSI protocol reference model (or the ATM layer in the ATM protocol stack), it would be interesting to assess the performance of the MSEC when another protocol is running above ATM, such as IP. Measurements could be made at the IP layer for the ETE delay, PLR, throughput, and other parameters of interest.

CHAPTER 7

REFERENCES

[1] A. Acampora, and M. Naghshineh, "An architecture and methodology for Mobile-Executed handoff in Cellular ATM Networks," *IEEE Journal on selected areas in communications*, Vol. 12, No.8, October 1994.

[2] A. Acampora, and M. Naghshineh, "QoS provisioning in micro-cellular networks supporting multiple classes of traffic," *Wireless Networks*, Vol. 2, No. 3, August 1996.

[3] ATM Forum, "ATM User-Network Interface (UNI) Specification Version 3.1," Prentice Hall 1994.

[4] E. Ayanoglou, S. Paul, F.T. LaPorta, K.K. Sabnani, and R.D. Gitlin, "AIRMAIL: A link-layer protocol for wireless networks," *Wireless Networks*, Vol. 1, No. I, 1995.

[5] M. Barton, and R. Hsing, "Architecture for wireless ATM networks," *6th Int. Symposium on Personal, Indoor, and Mobile Radio Comm. (PIMRC '95)* Toronto, Canada September 1995.

[6] G. Benneli, L. Favalli, and G. Filigheddu, "Error recovery for ATM transmission over wireless channels," *Electronics Letters*, Vol. 31, No. 16, 3rd, August 1995.

[7] D. Bertsekas, and R. Gallager, *Data Networks*, [Prentice-Hall, Second edition, 1992].

[8] J.C. Bibb, and D.N. McGregor, "A recommended Error Control Architecture for ATM Networks with Wireless Links," *IEEE Journal on selected areas in communications*, Vol. 15, No. 1, January 1997.

[9] Uyless Black, *Data Networks*, [Prentice-Hall 1989].

[10] Uyless Black, *ATM: Foundation for broadband networks*, [Prentice-Hall 1995].

[11] R.D. Gitlin, J.F. Hayes, and S.B. Weinstein, *Data Communication Principles*, [Plenum press, New York, 1992].

[12] R. Handel, and M. Huber, *Integrated Broadband Networks: An introduction to ATM-based networks*, [Addison-Wesley, 1991].

[13] International Telecommunications Union (ITU), *ISDN protocol reference model*, November 1993.

[14] K. Lee, "Supporting mobile multimedia in integrated services networks," *Wireless Networks*, Vol. 2, No. 3, August 1996.

[15] P. Lee, "A Mobile-Aware Transmission Control Protocol (TCP) for Wireless Communications," M.A.Sc. Thesis, Simon Fraser University, December 1996.

[16] S. Lin, and D.J. Costello, *Error control coding*, [Prentice-Hall, 1983].

[17] Z.Liu, M.J. Karol, M.El Zarki, and K.Y. Eng, "Channel access and interference issues in multi-code DS-CDMA wireless packet (ATM) networks," *Wireless Networks*, Vol. 2, No. 3, August 1996.

[18] Y. Lu, and R. Brodersen, "Unified power control, Error Correction Coding and Scheduling for a CDMA Downlink system," *INFOCOM '96* San Francisco CA USA, March 1996.

[19] Y. Nakayama, and S. Aikawa, : Cell Discard and TDMA Synchronization Using FEC in Wireless ATM Systems," *IEEE Journal on selected areas in communications*, Vol. 15, No. 1, January 1997.

[20] OPNET Manual, Mil3 Inc. 1994.

[21] K. Pahlavan, and A.H. Levesque, "Wireless data communications," *IEEE proceedings*, Vol. 82, No. 9, September 1994 pp. 1398-1430.

[22] B. Patel, and M. Schwartz, "Impact of mobility on resource allocation in ATM networks," *6th Int. Symposium on Personal, Indoor, and Mobile Radio Comm. (PIMRC '95)* Toronto, Canada September 1995.

[23] P. Polydorou, "Performance analysis and comparison of two wireless wide area networks: Mobile IP, and CDPD," M.A.Sc. Thesis, Simon Fraser University, December 1996.

[24] John G. Proakis, *Digital Communications*, [McGraw-Hill Series in Electrical Engineering, Second Edition, 1989].

[25] Martin de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*, [Ellis-Horwood Series in Computer Communications, 1993].

[26] D. Raychaudhuri, "Wireless ATM: An enabling technology for multimedia personal communication," *Wireless Networks*, Vol. 2, No. 3, August 1993.

[27] D. Raychaudhuri, and N.D. Wilson, "ATM-based Transport Architecture for Multiservices Wireless Personal Communication Networks," *IEEE Journal on selected areas in communications*, Vol. 12, No. 8, October 1992.

[28] M. Schwartz, *Telecommunication Networks, Protocols, Modeling and Analysis*, [Addison-Wesley, Series in Electrical and Computer Engineering, 1988].

[29] M.D. Yacoub, *Foundations of Mobile Radio Engineering*, [CRC Press, 1993].

[30] O. Yu, and V. Leung, "B-ISDN Architectures and protocols to support wireless personal communications internetworking," *6th Int. Symposium on Personal, Indoor, and Mobile Radio Comm. (PIMRC '95)* Toronto, Canada September 1995.

[31] N.D. Wilson, R. Ganesh, J. Kuriacose, and D. Raychaudhuri, "Packet CDMA versus Dynamic TDMA for Multiple Access in an Integrated Voice-Data PCN," *IEEE Journal on selected areas in communications*, Vol. 11, No. 6, August 1993.

APPENDIX A

LIST OF ABBREVIATIONS

AAL .....ATM Adaptation Layer

ARQ .....Automatic Repeat reQuest

ATM .....Asynchronous Transfer Mode

B-ISDN .....Broadband- Integrated Services Digital Network

BCH .....Bose-Chaudhuri-Hocquenghem

BER .....Bit Error Rate

BS .....Base Station

CATV .....CAble TV

CCITT ..... International Telegraph and Telephone Consultative Committee

CDPD .....Cellular Digital Packet Data

FEC .....Forward Error Correction

HDLC .....High-level Data Link Control

HEC .....Header Error Control

ITU .....International Telecommunications Union

MAC .....Medium Access Control

MS .....Mobile Station

MSC .....Mobile Switching Station

MSEC .....Mobile Specific Error Control

NNI .....Node-Network Interface

NGPCN .....Next Generation Personal Communications Network

OAM .....Operation And Management

OPNET .....OPtimized Network Engineering Tools

OSI .....Open Systems Interconnection

PC .....Personal Computer

PCN .....Personal Communications Network

QoS .....Quality of Service

SNR .....Signal-to-Noise Ratio

UNI .....User Network Interface

VC .....Virtual Channel

VCI .....Virtual Channel Identifier

VPI .....Virtual Path Identifier