Acknowledgments

I would like to acknowledge and express my appreciation to Dr Anthony Lo and Dr Winston Seah (Centre for Wireless Communications, National University of Singapore) for their invaluable advice and guidance throughout the course of this study. Also, especially to Mr Tan Tshun Kiat (Centre for Wireless Communications, National University of Singapore) for his patience and timely assistance to install a working version of Network Simulator (*ns*).

Next, I would like to thank Mr Duan Zhen Hai (duan@cs.umn.edu) for his advice on building the Makefile for *ns*, Mr Paul Ghobril (ghobri@enst.fr) for providing suggestions on TDMA simulation in *ns,* Mr Liu Liqun (Centre for Wireless Communications, National University of Singapore) and many others on the *ns* mailing list who have responded to my questions.

Last but not least, I would like to thank my parents for keeping up with my late nights as I work on my dissertation, and their emotional support throughout the course of this project.

Table of Contents

List of Abbreviations / Acronyms

2G	Second Generation
3G	Third Generation
8-PSK	8-Phase Shift Keying
ACK	Acknowledgment
ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
BLER	Block Error Rate
BSN	Block Sequence Number
BSS	Base Station Subsystem
C/I	Carrier-to-Interference ratio
CAS	Packet Channel Assignment
CDMA	Code Division Multiple Access
CMU	Carnegie-Mellon University
CRQ	Packet Channel Request
DECT	Digital Enhanced Cordless Telephone
DSSS	Direct Sequence Spread Spectrum
ECS	Enhanced Coding Scheme
ECSD	Enhanced Circuit Switched Data
EDGE	Enhanced Data rates for GSM Evolution
EGPRS	Enhanced GPRS
ETSI	European Telecommunications Standards Institute
FH	Frequency Hopping
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GMSK	Gaussian Minimum Shift Keying
GPRS	General Packet Radio Services
GSM	Global Systems for Mobile Communications
HSCSD	High Speed Circuit Switched Data
IFq	Interface Queue
IP	Internet Protocol
IMT	International Mobile Telecommunications
ITU	International Telecommunications Union
LA	Link Adaptation

LAN	Local Area Network
LL	Link Layer
LLC	Logical Link Control
MAC	Medium Access Control
MS	Mobile Station
NACK	Negative Acknowledgment
NetIF	Network Interface
ns	Network Simulator
O16QAM	Offset 16-Quadrature Amplitude Modulation
PACCH	Packet Access Common Channel
PACK	Pending Acknowledgment
PAGCH	Packet Access Grant Channel
PDC	Personal Data Communications
PDCH	Packet Data Channel
PPCH	Packet Paging Channel
PRACH	Packet Random Access Channel
Prop Model	Propagation Model
PRP	Packet Paging Response
PRQ	Packet Paging Request
QoS	Quality of Service
RLC	Radio Link Control
R_n	Radio Interface rate for coding scheme n
Rtagent	Routing Agent
RTO	Re-transmission Time Out
SGSN	Serving GPRS Support Node
S_n	Throughput for coding scheme n
SNDCP	Sub-Network Dependent Convergence Protocol
STD / std	Standard Deviation
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TS	Time Slot
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Systems
WWW	World Wide Web

Summary

Enhanced Data rates for GSM Evolution (EDGE) is a radio access technology capable of providing wireless Internet access within the framework of Third Generation (3G) mobile systems' requirements. It offers higher bit rates compared to current Second Generation (2G) cellular networks such as Global System for Mobile communications (GSM), General Packet Radio Service (GPRS).

Current research on EDGE focuses on providing non real-time data services using Enhanced General Packet Radio Services (EGPRS) and Enhanced Circuit Switched Data (ECSD). This study evaluates TCP performance of bulk data transfer between a Remote Host and a Mobile Station (MS) across an EDGE network using Network Simulator (*ns*). The top level simulation scenario is shown in Figure i. The simulation considers the packet mode, i.e., EGPRS, since it offers data rates up to 384 kbps compared to 64 kbps by ECSD.

From the top level, the level of abstraction for implementing the layered protocols is shown in Figure ii. A File Transfer Protocol (FTP) object is attached to the Remote Host via a TCP agent to support uni-directional data transfer. Another TCP agent is attached to the MS to simulate end-to-end TCP flow control. TCP packets are sent from the Remote Host to the Base Station Subsystem (BSS) via IP relays. Fragmentation at SubNetwork Dependent Convergence Protocol (SNDCP) and Link Layer Control (LLC) layers is collectively implemented, where IP packets are directly fragmented into Radio Link Control (RLC) blocks.

RLC flow control with selective Automatic Repeat reQuest (ARQ) is implemented for reliable data transfer across the radio link between the BSS and the MS. Time Division Multiple Access (TDMA) transmission is supported, with dynamic allocation of radio resources. Contention resolution in the up-link is unnecessary since only one MS is defined. Our simulation performs single-slot operation and multi-slot data rates can be computed according to the number of time slots used.

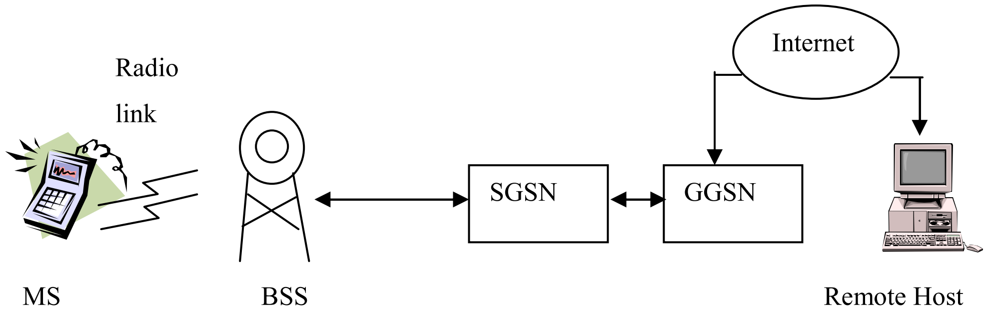


Figure i: EGPRS top level simulation environment

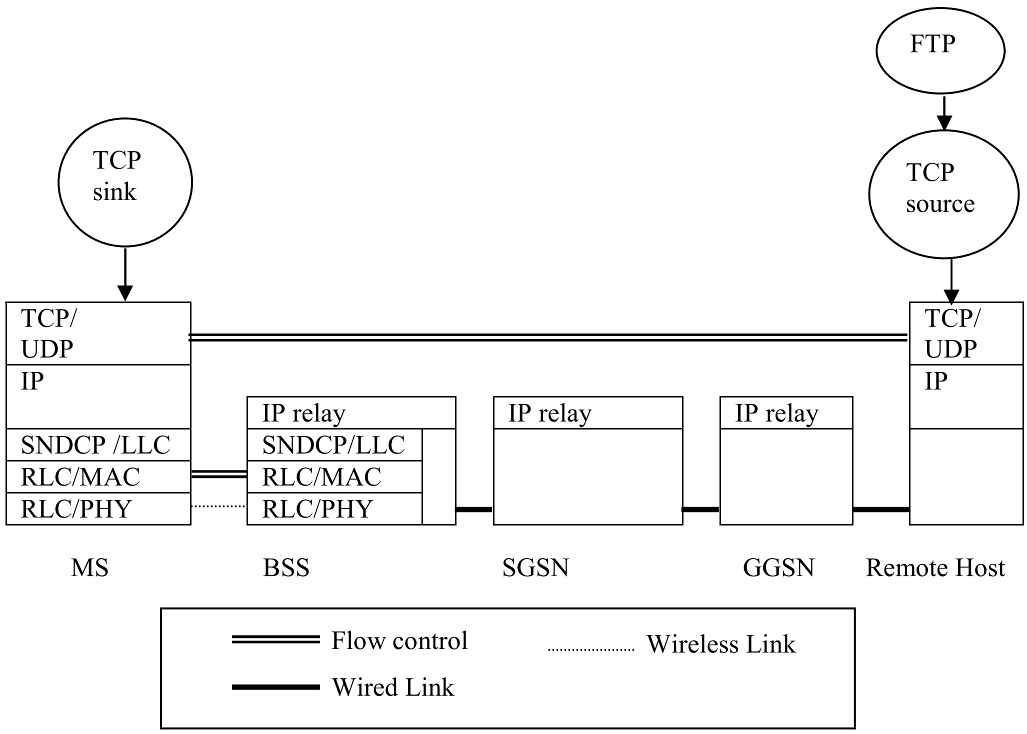


Figure ii: Implementation of EGPRS protocol structure in *ns*

All proposed Enhanced Coding Schemes, ECS-1 to ECS-8, which are defined in EGPRS, and Offset 16 Quadrature Amplitude Modulation (O16QAM) modulation, are supported. Both the Estimated Carrier-to-Interference ratio (C/I) and BLock Error Rate (BLER) algorithms for RLC link adaptation are implemented.

TCP performance is evaluated based on *throughput*, *average packet delay* and *delay variation* of received TCP packets. The formulation of theoretical throughput was modified to include effects of re-transmission, which is significant for high-level modulation. Enhanced data rates up to 240 kbps (with multi-slot operation) can be achieved under good channel conditions, but these levels fall rapidly as channel conditions deteriorate. Adaptive polling attempts to improve performance under poor channel conditions. Delay performance is improved, but the slight improvement in throughput under poor channel conditions is traded off with throughput degradation under good channel conditions.

Estimated C/I approximates ideal link adaptation performance for very good channel conditions; while the BLER approach performs well for poor channel conditions. A combined approach improves overall link adaptation performance.

Chapter 1

INTRODUCTION

This chapter explains the need for multi-media capable mobile communications for Internet access. It addresses the requirements of International Mobile Telecommunications – in Year 2000 (IMT-2000) and the evolution of Second to Third Generation mobile systems in the network and radio perspective to meet these requirements. The proposals for IMT-2000-capable radio access systems are next described. The importance of Enhanced Data rates for GSM Evolution (EDGE) is outlined, forming the basis of this study. A short description of the Internet follows. Several related works are addressed, followed by objectives of the project and the organization of the report.

1.1. Need for Multi-Media Capable Mobile Communications

There is a rising emergence of the Internet and Internet-based techniques to provide multi-media services to the mass market. In parallel, mobile telephony has also experienced a tremendous growth, with Second Generation (2G) systems, e.g., Global Systems for Mobile communications (GSM)[1], reaching high penetration rates in some markets. The expected penetration rate for mobile communications in developed countries is expected to hit 50 – 80 % within the next few years, serving more than 200 million subscribers worldwide by end of 2000. By the end of the year 2003, it is predicted that there will be some 830 million wireless subscribers and over 700 million Internet subscribers [2]. This translates into a huge market potential for mobile Internet solutions, and mobile extensions to all our basic desktop applications and data files, independent of where we are.

1.2. Mobile data evolution to Third Generation (3G) systems

2G mobile communications provide digital voice communications, as well as data services, mainly circuit-switched low to medium rate data communications (e.g., 9.6 kbps). Such systems include GSM, Personal Data Communications (PDC) and Code Division Multiple Access (CDMAone / IS-95).

Data services offered at present suffers from the following disadvantages: [2]
(a) restricted user data rate (9.6 kbps),
(b) call routing over the telephone network to data networks,
(c) higher price,
(d) connection time (and not usage) charge for subscriber and
(e) long call-setup time.

The data rates of 2G systems are not adequate for many Internet applications, especially those that have a large amount of traffic to send, such as a file transfer. To ameliorate this situation, new data services have evolved, taking the mobile data communications towards 3G mobile systems, as shown in Figure 1.1. Higher bit rate services have evolved from GSM systems, offering up to 100 kbps services in High Speed Circuit-Switched Data (HSCSD)[3]. Packet data services have been introduced into GSM in General Packet Radio Services (GPRS)[4] to extend the data rate capabilities of GSM systems, which commonly refer to as 2.5G systems. EDGE offers further enhancements by introducing new modulation schemes that result in higher data rates.

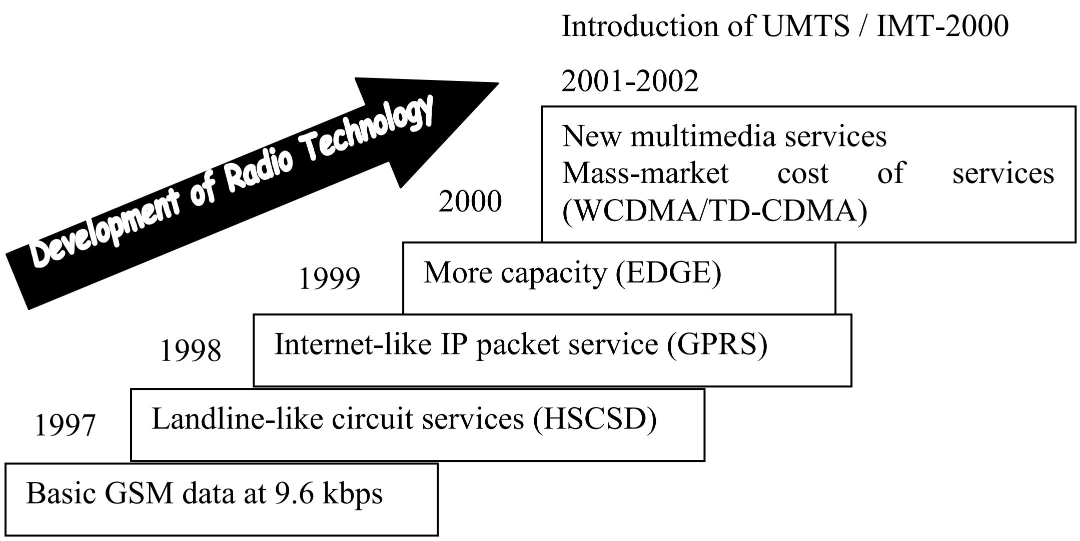


Figure 1.1: Mobile data evolution [2]

New innovative services, in particular broadband multi-media, will form the basis of true 3G systems. 3G systems will be based on the existing GSM due to existence of a mass market. GSM and its enhancements (HSCSD and GPRS) can fulfill the needs of speech and low-rate data services while Universal Mobile Telecommunications Systems (UMTS) and EDGE will concentrate on providing seamless access to high bit rate multi-media and packet data services.

1.3. IMT-2000 Requirements for 3G systems

The International Telecommunications Union (ITU) has defined a set of requirements for 3G systems under IMT-2000 [2], which are defined in the next subsections.

1.3.1. Multi-environment operation

3G systems expect to serve different radio environments ranging from vehicular, pedestrian and outdoor-to-indoor, indoor office and satellite. This necessitates classification into terrestrial- and satellite-based systems.

Terrestrial-based systems will offer local coverage with 2 Mbps data rates with maximum speed of 10 km/h, and wide area coverage with 384 kbps in suburban outdoor with maximum speed of 120 km/h. Satellite-based systems will offer data rates of 144 kbps in rural areas with maximum speed of 500 km/h.

1.3.2. Support of wide and changing range of services

3G systems are to offer service capabilities, rather than the services themselves, to the end user. Standardization of service capabilities includes standardization of bearers, Quality of Service (QoS) parameters and additional mechanisms to realize the required services. Examples of service capabilities include speech, video, multi-media, access to Internet, messaging, data and other tele-services. Services within the mobile networks and within the fixed networks shall be compatible.

1.3.3. Seamless wireless access to mass market multi-media

Deregulation of world wide market and rapid introduction of mobile services by 2G mobile services has led to the conclusion that one 'ultimate mobile solution', one radio access network and one single core network is not realistic. 3G system development takes into account opportunity of choice for users, networks and service operators and multiplicity of existing and future fixed and mobile telecommunication networks and services.

Some existing 2G systems do already incorporate multiple network interfaces and are called multi-homed systems. Examples of such systems in mobile applications include dual-mode handsets, for instance, GSM/Digital Enhanced Cordless Telephone (DECT), and portable computers with IrDA and Ethernet Interfaces.

1.4. IMT-2000 Capable Mobile Systems

The evolution of 2G systems to meet IMT-2000 requirements for 3G systems can be viewed from two perspectives, namely, network and radio perspectives [5].

(a) Network perspective

Adding 3G capabilities implies addition of packet switching, Internet access and IP connectivity capabilities. These capabilities have already been introduced into GSM in GPRS.

(b) Radio perspective

From a radio perspective, adding 3G capabilities to 2G systems mainly means supporting even higher data rates. Since HSCSD and GPRS are based on the original Gaussian Minimum Shift Keying (GMSK) modulation, the increase in bit rate is moderate and below the IMT-2000 data rate requirement. Two solutions have been identified based on evolution of GSM. EDGE operates on existing frequency bands at 800, 900 and 1800 MHz and it offers enhanced data rates with high-level modulation capabilities in 200kHz Time Division Multiple Access (TDMA) systems. UMTS operates on a new 2GHz frequency band as well as existing GSM bands and is based on new technologies optimized for new services and minimum costs.

The advantages offered by EDGE over UMTS include fast availability through reuse of existing GSM and TDMA infrastructure. The standardization roadmap for EDGE foresees two phases [5]. In the first phase, the emphasis has been placed on Enhanced GPRS (EGPRS) and Enhanced Circuit Switched Data (ECSD) [6]. Both were targeted in European Telecommunications Standards Institute (ETSI) for standards release in 1999 with products to follow shortly. The second phase of EDGE is targeted for release in year 2000 and is currently being defined with improvements for multi-media and real-time services as possible work items.

1.5. Technical Motivations for EDGE

The enhancements defined in EDGE are basically in the link and physical layers to provide high data bit rates efficiently.

1.5.1. Improvement in Spectral Efficiency

Different users tend to have different channel qualities in terms of Carrier-to-Interference Ratio (*C/I*). A traditional service such as speech requires a certain target *C/I* to give good quality: below the target, the quality is unacceptable, while above the target, the quality is good and practically independent of channel quality.

Radio network planning needs to ensure that a small fraction of users are below the *C/I* target and this results in a large part of the user population experiencing unnecessarily high *C/I*, from which they cannot reap any benefit. Data rates for such users can thus be traded for improvement in capacity, improving spectral efficiency. This is achieved by employing link quality control.

Link quality control adapts the protection of data to the channel quality so that for all channel qualities, an optimal bit rate is obtained.

1.5.2. Improvement in data rates to meet 3G requirements

EDGE achieves significantly higher information bit rates than those of standard GSM/GPRS with the introduction of new modulation schemes, which will be elaborated in Chapter 2.

1.6. The Internet

The Internet is a packet-switched network whereby the source message is split into packets, which are sent independently through the network one at a time. The network delivers the packets to the specified destination, where the whole message is reassembled. The network layer protocol used in the Internet is called the Internet Protocol (IP) [7] and is defined as an unreliable, connectionless delivery mechanism.

The transport layer protocol that is most widely used in the Internet is the Transmission Control Protocol (TCP) [7]. It is a connection-oriented protocol, which provides a reliable end-to-end data transfer. TCP includes complex algorithms (e.g., slow start) to adapt to a variety of combination of communication links, network topologies and computational speeds. Under normal circumstances, TCP adapts to transfer data reliably between any two hosts at speeds that approach those of the underlying links that comprise the path between two hosts. TCP also protects the state of the Internet by detecting congestion and quickly reducing the number of segments it will transmit when congestion has occurred.

TCP assumes that the overwhelming majority of lost segments and acknowledgments are due to congestion in the Internet. This assumption is quite valid for wired links. However, for wireless links, this assumption is false – most lost segments are due to errors that occur in the transmission of segments over the error-prone radio links. When these errors occur, TCP mistakenly assumes that the network is congested and dramatically reduces its transmission window size. This causes the performance over TCP over wireless links to be quite poor [8, 9]. However, most of these performance studies and experiments have been conducted over Wireless Local Area Networks (WLANs), which are different from cellular networks such as EDGE, particularly at the radio and link layers. Therefore, it is interesting to assess the performance of TCP over such a cellular network.

1.7. Related Work

A number of studies on the evaluation of TCP performance over cellular networks can be found in the literature (e.g., [10], [11]).

1.7.1. Performance evaluation of Internet Access via the GPRS of GSM [10]

A group from Ericsson Eurolab Deutschland has carried out a study on TCP/IP performance evaluation of Internet traffic over GPRS. The GPRS simulator is implemented using the simulation tool BONeS, which supports a block oriented hierarchical system modeling approach. The simulator models the whole GPRS node chain including the Internet.

World Wide Web (WWW) model is used to simulate applications and user behavior and TCP version Tahoe is used. The Internet is modeled as a self-similar delay model. The mobile is assumed to start in standby mode. The error model of the radio link is based on pre-simulated Block Error Rate (BLER) with the first and second moment of the *C/I* as user parameters. The mobile station is assumed to be stationary throughout the simulation.

Simulation results indicate a strong dependency between the throughput and the underlying channel conditions, i.e., each of the four coding schemes is optimal at different channel conditions. It is concluded that GPRS is a highly suitable bearer service for TCP/IP traffic.

1.7.2. TCP performance over GPRS

In this paper, Meyer [11] provided further insight into the behavior of TCP over GPRS. The study focused on bulk data transfer using TCP/IP over the GPRS interface to a single user in unacknowledged Link Layer Control (LLC) mode. It is assumed that four Packet Data Channels (PDCHs) are allocated to GPRS, i.e. that four timeslots on one GSM frequency can be used for GPRS. The mobile terminal has the capability to use 1 PDCH in up- and 4 PDCHs in down-link direction in parallel for increased data rates.

In the study of throughput versus channel conditions, it was verified that as channel conditions improve, it is highly desirable to switch to less robust coding schemes, that will finally result in higher throughputs. However, the theoretical throughput cannot be provided due to re-transmission of Radio Link Control (RLC) blocks, protocol overhead (up to 10 %) as well as signaling messages. Under optimal conditions, a throughput of 85 % of the theoretical value can be achieved.

In the analysis of TCP behavior over GPRS, even under bad channel conditions, the RLC Automatic Repeat Request (ARQ) mechanism works sufficiently fast to avoid spurious TCP timeouts. Link Layer problems are just recognizable by increased packet inter-arrival times of TCP segments. In addition, TCP adapts its protocol parameters, e.g., Re-transmission Time-Out (RTO) value. Both the given points contribute to the stability of TCP over GPRS.

It is also concluded that the RLC ARQ scheme is appropriately designed to ensure that TCP observes just packet delays rather than losses.

1.7.3. Link Adaptation algorithms in GPRS

Olav Queseth et al., [12] presented two algorithms for link adaptation (i.e., choosing the most suitable coding scheme for transmission). The first one is based on the Estimated C/I and the second one on BLER.

The objective of link adaptation is to maximize the throughput of a channel by using the most suitable coding scheme at any given moment. In conclusion, a stronger coding scheme is more suitable for bad channels and a weaker coding scheme is more suitable for good channels.

In GPRS data transmission, data is transmitted over one polling interval, which is a number of RLC blocks long. After the end of each polling interval, an Acknowledgment / Negative Acknowledgment (ACK/NACK) is expected from the receiver, followed by either further transmissions or re-transmissions.

In both link adaptation algorithms, transmission begins with the most robust coding scheme for one polling interval (e.g., 10 RLC blocks), and the average *C/I* and BLER are measured over the window. This is used to determine the coding scheme to be employed for the next polling interval. A stronger coding scheme will be applied to re-transmissions than to ordinary transmissions.

Algorithms based on Estimated C/I are superior to those based on BLER. The improvement in terms of throughput per user is in the order of 10 %.

1.8. Objective of the project

GPRS products will be available in the market in the near future, while EDGE standardization is still underway. As these products will be used mainly to support Internet applications, e.g., File Transfer Protocol (FTP), via wireless access, it is important to evaluate the expected TCP performance to verify the evolution towards 3G requirements. To date, none of the related works have evaluated the performance of TCP over EDGE.

This is the motivation behind this project, which evaluates TCP performance over EDGE for wireless Internet access using a simulation model, which is built on Network Simulator (*ns*)[13,14]. Since the first phase of EDGE will provide data services only, our study will focus on EGPRS, which offers much higher data rates (up to 384 kbps) than ECSD (up to 64 kbps).

1.9. Organization of the report

This chapter has provided introductory material on 3G mobile system requirements, EDGE and the Internet. Work in related areas as well as motivation and objective for this project have also been described. The rest of the report is organized as follows. Chapter 2 gives a more detailed description of EDGE, which forms the fundamentals to understand the design of the simulation model. Network Simulator, the software package from which the simulation model is built on, is described in Chapter 3. Chapter 4 describes the simulation model design and implementation, followed by simulation results in Chapter 5. Chapter 6 provides concluding remarks and recommendations.

Chapter 2

OVERVIEW OF EDGE

This chapter introduces the concept of EDGE, but the focus is on its packet mode, i.e., EGPRS, from which our simulation model is developed. The basic radio interface parameters as well as the radio link protocol design of EGPRS are first described. This is followed by a description of the basic EGPRS / GPRS operation.

2.1. Motivation for EDGE development

As speech was envisioned as the main service to be provided in 2G mobile communications systems (e.g., GSM), data services offered are limited to 9.6 kbps. However, the explosive growth of the Internet coupled with the increased mobility of its users are fueling the demand for wireless Internet access services (as illustrated in Figure 2.1) with the same QoS as perceived via wireline access. This drives the need for advanced radio technology to provide high radio access rates. In view of this, standardization of 3G mobile communications services is now rapidly progressing in all regions of the world, aimed at enhancing the data rates and services provided by 2G systems.

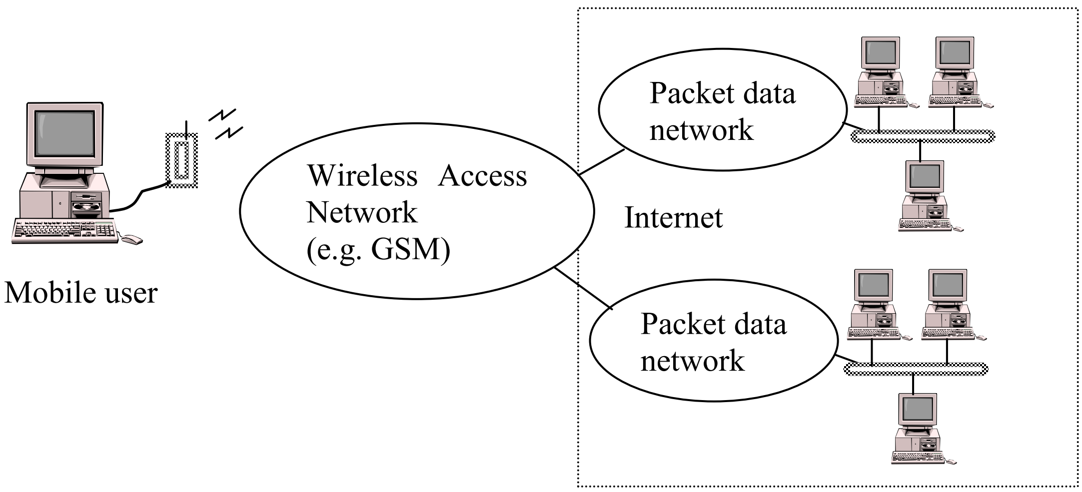


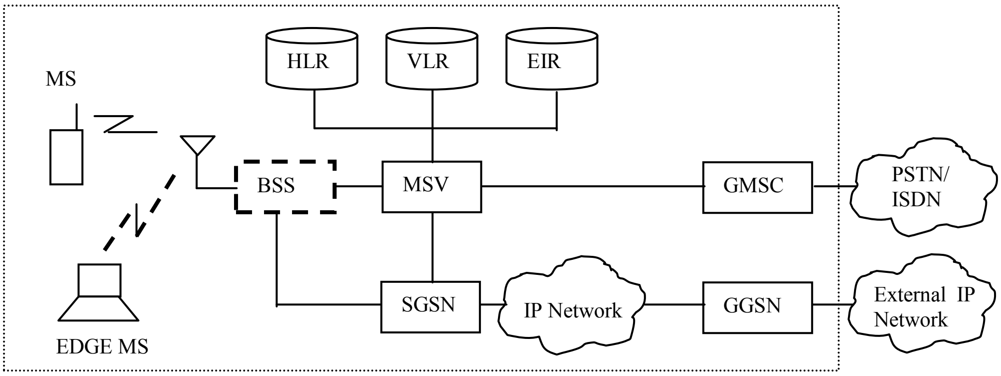
Figure 2.1 Illustration of wireless Internet access

ETSI took the first step by initiating standardization activities for two additional high data rate services that evolved from GSM in order to achieve fast availability using existing infrastructure. HSCSD is the circuit-switched extension of GSM that offers up to 64 kbps of user bit rate.

By introducing additional network elements to GSM, packet-switched transmission can be supported, where many users can share the scarce radio resource. GPRS is the packet-oriented extension of GSM, which supports data transmission up to rates more than 100 kbps. Both HSCSD and GPRS achieve high data rates over GSM by using multi-slot operation, while retaining the use of GMSK modulation of GSM.

However, data rates achievable are still a far cry from the IMT-2000 requirement of 384 kbps, defined for 3G systems to provide high data rate and multi-media services over a wide coverage area to moderate mobility users.

This spurred further development efforts that eventually led to the concept of EDGE. EDGE is seen as a complement to UMTS (for CDMA systems) and is currently being specified to provide a 3G evolution for TDMA systems, e.g, GSM. EDGE uses high-level modulation and higher symbol rates to offer additional enhancements to current GSM systems. As such, it only defines a new radio access technology and reuses the infrastructure of GSM/GPRS. The parts in the GSM/GPRS system, which are affected by EDGE introduction, are the Mobile Station (MS) and the Base Station Subsystem (BSS), as illustrated in Figure 2.2.



Legend: – – – – – Affected by EDGE

BSS – Base Station Subsystem
EIR – Equipment Identity Register
GGSN – Gateway GPRS Support Node
HLR – Home Location Register
MS – Mobile Station
SGSN – Serving GPRS Support Node
MSC – Mobile Switching Center
GMSC – Gateway Mobile Switching Center

Figure 2.2 EDGE Architecture

EDGE offers enhancements to the circuit-switched services of HSCSD as well as the packet switched services of GPRS. The enhanced circuit-switched services, ECSD, offer bit rates similar to HSCSD (up to 64 kbps). The enhanced packet data service, EGPRS, gives bit rates up to 384 kbps, thereby meeting the IMT-2000 requirements. Hence, EGPRS will be the main focus of the descriptions that follow.

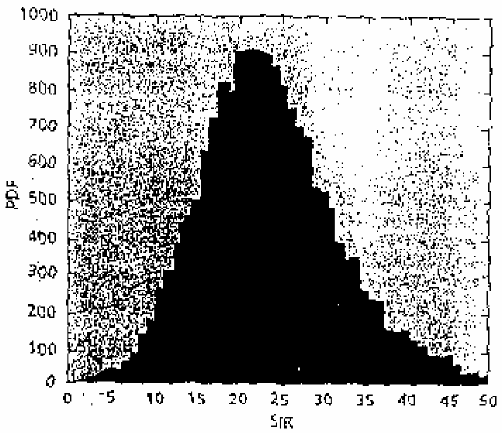
2.2. ECSD

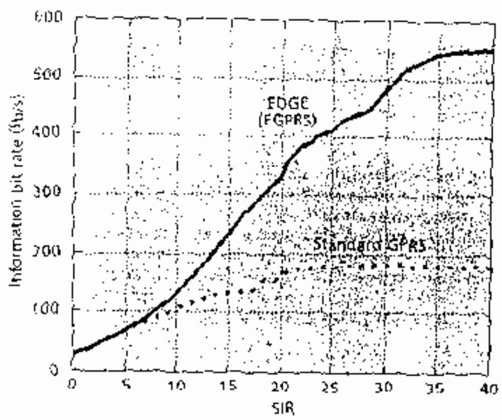
ECSD is being developed using current HSCSD as a basis. The new data rates are not increased compared to HSCSD (up to 64 kbps), but these rates can be achieved with less number of Time Slots (TSs). The excess TSs can be allocated to another user for data transmission and this implies an improvement in capacity. The ECSD architecture is largely based on HSCSD transmission and signaling so that there will be a minimal impact on existing specifications and infrastructure.

2.3. EGPRS

As mentioned, EGPRS is the packet-switched extension of GSM. Besides offering higher data rates compared to GPRS, EGPRS is also more spectral efficient.

Figure 2.3(a) shows the typical distribution of channel quality. In order to assure most users enjoy the minimum required SIR (or equivalently C/I) for good voice quality, radio network planning sets this value to, e.g., 10 dB. As depicted in Figure 2.3(b), GPRS information bit rate saturates at rather low SIR, thus 'wasting' the SIR in excess of the minimum requirement. On the other hand, EGPRS benefits from this excess SIR by 'converting' them into additional data rate capabilities. To achieve these data rates, high-level modulation is introduced into EGPRS, as in the case of ECSD.





(a) (b)

Figure 2.3(a) SIR distribution and (b) EGPRS/GPRS data rates vs SIR [5]

2.4. Basic Radio Interface Parameters of EGPRS

In order to facilitate higher bit rates than those currently achievable, high-level modulation schemes are introduced to complement GMSK modulation. Offset-16 Quadrature Amplitude Modulation (O16QAM) and 8-Phase Shift Keying (8-PSK) were proposed as potential candidates, and their characteristics are described in terms of *modulation* and *burst format*.

2.4.1. Modulation

GMSK is the existing modulation scheme deployed in GSM / GPRS. It replaces the sinusoidal pulse shape in MSK with a Gaussian pulse, resulting in a signal with low side lobes and narrower main lobe. The resulting good spectral and power efficiency in GMSK is traded off with increased Inter-Symbol-Interference (ISI) compared with MSK. However, as long as the GMSK-induced error rate is less significant compared to mobile channel-induced error rate, there is no penalty in using GMSK. In GMSK, each symbol contains 1 bit of encoded information.

O16QAM is a linear high-level modulation scheme. Each symbol carries 2 bits of encoded information. The symbol rate and pulse shaping are similar to GSM, and thus it fits into the GSM spectrum mask and will coexist with GSM in its existing frequency plan. The signal constellation diagram is shown in Figure 2.4(a).

8-PSK is another linear high-level modulation scheme. Each symbol carries 3 bits of encoded information. The pulse shape of 8-PSK is linearised GMSK, allowing it to fit into the GSM spectrum mask. 8-PSK is less sensitive to noise than O16QAM, but O16QAM offers higher spectral efficiency. The signal constellation diagram is shown in Figure 2.4(b).

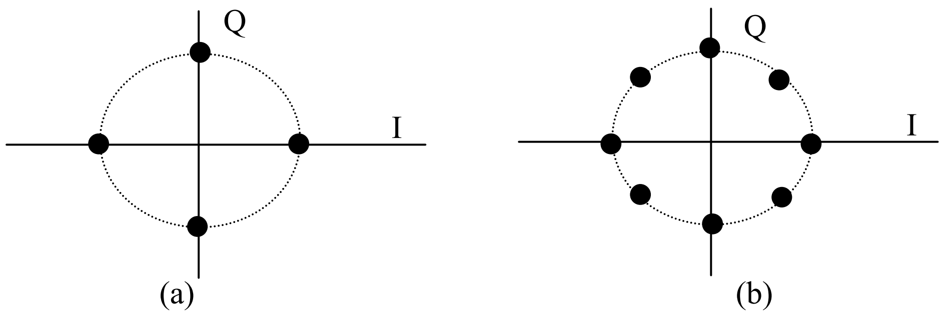


Figure 2.4: Signal constellation diagrams of (a) O16QAM and (b) 8-PSK

2.4.2. Burst Format

One requirement for EDGE is to reuse as much as possible the existing GSM standard. Hence, the original GSM burst structure (one burst is equivalent to one Time Slot (TS)) with duration of 576.92 μs is retained.

Figure 2.5(a) shows the burst format of GSM. Each burst contains a training sequence, which comprises 26 symbols, 3 tail symbols at either end, and 8.25 guard symbols at one end. Each burst carries 2 x 58 data symbols, including 2 stealing bits. In total, the payload comprises 114 bits (2 x 58 – 2), giving rise to a gross bit rate per TS of 22.8 kbps.

In O16QAM, each burst contains a training sequence of 28 symbols, 2 tail symbols at either end, and 12.3 guard symbols at one end. Each burst carries 2 x 82 data symbols, including 2 stealing bits. In total, since each symbol carries 2 bits of information, the payload comprises 326 bits (2 x 82 x 2 –2), which results in gross bit rate per TS of 65.2 kbps. The burst structure is illustrated in Figure 2.5(b).

In 8-PSK, the original burst format of GSM is used, as shown in Figure 2.5(a). However, since each symbol carries 3 bits of information, the payload comprises 346 bits (2 x 58 x 3 – 2). Hence, gross bit rates per TS of 69.2 kbps can be achieved.

3	58	26	58	3	8.25

(a) GSM / GPRS and EGPRS (8-PSK)

2	82	28	82	2	12.3

(b) EGPRS (O16QAM)

Figure 2.5: Burst format for (a) GSM(GPRS)/EGPRS(8-PSK) and (b) EGPRS(O16QAM) in terms of symbols

2.5. Radio Link Protocol Design of EGPRS

2.5.1. Definition of coding schemes

High-level modulation schemes are known to be more susceptible to noise and interference compared to GMSK, and hence, throughput may diminish under poor channel conditions, where noise and interference levels become significant. Therefore, in order to achieve maximum throughput under all channel conditions, there is a need to adapt the transmission scheme used to the interference situation.

Hence, different coding schemes are defined for each modulation type by setting different code rates relative to full-rate transmission, such that low-rate schemes are more robust against noise and interference. However, under very poor channel conditions, high-level modulation does not give sufficient robustness even if very low code rates are employed. Hence, the more robust GMSK modulation is retained. Four coding schemes (CS-1 to CS-4) are defined for GMSK using various code rates, and they correspond to GPRS coding schemes. Their characteristics are shown in Table 2.1. Also shown in the last column of Table 2.1 are the radio interface rates (kbps) for each coding scheme. As radio interface rate increases, the coding scheme becomes less robust against noise and interference.

Coding Scheme	Code Rate	Modulation Scheme	Radio Interface Rate (kbps)
CS-1	0.49	GMSK	11.2
CS-2	0.64	GMSK	14.5
CS-3	0.73	GMSK	16.7
CS-4	1	GMSK	22.8

Table 2.1: Characteristics of coding schemes of GPRS

Four coding schemes (ECS-1 to ECS-4) based on O16QAM are defined for EGPRS. Since GMSK and O16QAM have different symbol rates, Offset Quadrature Phase Shift Keying (OQPSK) is defined to replace GMSK. This is to ensure that when link adaptation switches from high to low data rate coding schemes, it can be carried out more easily without switching symbol frequency, while retaining the robustness of GMSK. Four other coding schemes (ECS-5 to ECS-8) are defined based on OQPSK. The parameters of ECS-1 to ECS-8 are shown in Table 2.2.

For 8-PSK, six additional coding schemes (PCS-1 to PCS-6) are defined on top of the original GPRS coding schemes (CS-1 to CS-4). The parameters of these schemes are shown in Table 2.3.

Coding Scheme	Code Rate	Modulation Scheme	Radio Interface Rate (kbps)
ECS-5	0.35	OQPSK	11.2
ECS-6	0.45	OQPSK	14.5
ECS-7	0.52	OQPSK	16.7
ECS-8	0.70	OQPSK	22.8
ECS-1	0.51	O16QAM	33.0
ECS-2	0.63	O16QAM	41.0
ECS-3	0.74	O16QAM	48.0
ECS-4	1	O16QAM	65.2

Table 2.2: Characteristics of coding schemes of EGPRS (O16QAM)

Coding Scheme	Code Rate	Modulation Scheme	Radio Interface Rate (kbps)
CS-1	0.49	GMSK	11.2
CS-2	0.64	GMSK	14.5
CS-3	0.73	GMSK	16.7
CS-4	1	GMSK	22.8
PCS-1	0.33	8-PSK	22.8
PCS-2	0.5	8-PSK	34.3
PCS-3	0.6	8-PSK	41.25
PCS-4	0.75	8-PSK	51.6
PCS-5	0.83	8-PSK	57.35
PCS-6	1	8-PSK	69.2

Table 2.3: Characteristics of coding schemes of EGPRS (8-PSK)

2.5.2. Link quality control

In GPRS, four coding schemes have been defined to allow for variations in radio channel conditions. Link adaptation algorithms [12] have been proposed to select the most suitable coding scheme based on the channel condition at that moment. A link adaptation scheme regularly estimates the link quality and subsequently selects the most appropriate modulation and coding scheme for coming transmissions in order to maximize the user bit rate.

However, owing to the higher bit rate and higher sensitivity to noise in EGPRS coding schemes, incremental redundancy has been proposed for EGPRS to complement the capabilities of link adaptation.

In an incremental redundancy scheme [19], information is first sent with very little coding, yielding a high bit rate if decoding is immediately successful. If decoding fails, additional coded bits (redundancy) are sent until decoding succeeds. The more coding that has to be sent, the lower the resulting bit rate and the higher the delay. In addition, substantial memory is required at the receiver to store all coded bits belonging to a block till it is successfully received and decoded.

A flexible link quality control solution, Two Burst Based Link Quality Control (2BB LQC) [20], has been proposed for EGPRS that will support a combined link adaptation and incremental redundancy scheme. This approach is envisioned to achieve robustness and high throughput of the incremental redundancy approach, along with lower delays and lower memory requirements using link adaptation.

However, due to lack of sufficient information on 2BB LQC to carry out the simulation, this study will focus on link adaptation algorithms.

2.5.3. Radio Link flow control with selective ARQ

A link adaptation scheme selects the most appropriate modulation and coding scheme for coming transmissions based on the link quality at that moment in order to maximize the user bit rate. However, the link quality is not available beforehand, and so can only be estimated via feedback from the receiver, which rides on ARQ Status Feedback messages, which is part of the RLC flow control mechanism. Hence, this will be described before describing the link adaptation mechanism.

The RLC protocol for GPRS/EGPRS is designed to provide a reliable link across the radio interface. Data is transmitted as discrete RLC blocks, and each block is indexed. Data transmission takes place in polling intervals based on these indices.

For example, as illustrated in Figure 2.6, 10 blocks with index 1 to 10 are transmitted in the current polling interval. At the end of the interval, the sender sets the polling flag of the last block, i.e., block 10, to request for transmission status of all blocks belonging to the current interval, i.e., blocks 1 to 10. It waits for an interval of two RLC blocks to receive the acknowledgment status (ACK message) from the receiver. In this case, blocks 4, 7 and 8 are transmitted in error, and hence, they will be transmitted first in the next polling interval. This will be followed by blocks 11, 12, etc. In this way, the next polling interval is updated, thereby implementing flow control.

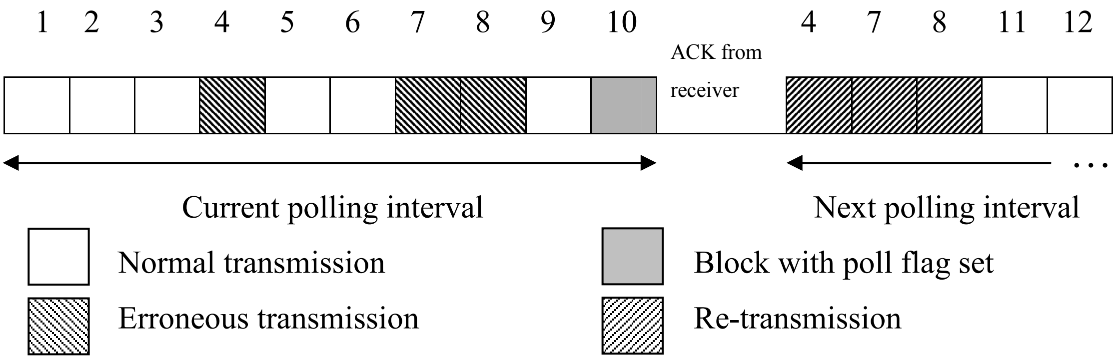


Figure 2.6: Illustration of RLC flow control with selective ARQ

2.5.4. Link Adaptation

As mentioned in Section 2.5.3, the estimate of the link quality is computed at the receiver and is fed back to the sender via the ARQ Status Feedback messages once every polling interval. Two parameters are useful to indicate the link quality, namely, *C/I* and *average error rate*, and link adaptation algorithms based on each parameter are described next.

2.5.4.1.Link Adaptation based on Estimated C/I

C/I is a useful metric to represent the channel quality where co-channel interference dominates over thermal noise, i.e., in an interference-limited system. A large value of C/I indicates good channel quality and vice versa.

It has been mentioned in [12] that the C/I can be estimated at the receiver with a mean error of 1 dB from the actual value. In our simulations, where the *Actual_C/I* is known and is constant over the simulation duration, the *Estimated_Mean_C/I* over the polling interval is obtained as follows:

$$Estimated\_Mean\_C/I\ (dB) = Actual\_C/I\ (dB) + normal(0,1) \qquad (2.1)$$

where normal(0,1) is a normal random variable with mean 0 and standard deviation of 1 dB

The *Estimated_mean_C/I* computed in eqn. (2.1) is transmitted to the sender and is used to compute the radio interface throughput, *S(CS),* for each coding scheme, *CS*, based on the following formula [18]:

$$S(CS) = R(CS)\times(1-BLER_{CS}(Estimated\_mean\_C/I)) \qquad (2.2)$$

where *R(CS)* is the radio interface rate corresponding to coding scheme *CS*, and $BLER_{CS}(Estimated\_mean\_C/I)$ is the *BLER* at the estimated channel quality, *Estimated_mean_C/I*, for each coding scheme, *CS*, obtained via link simulations [15]. Figure 2.7 shows the *BLER* vs *C/I* for the case of ideal frequency hopping using O16QAM.

Link adaptation then selects the coding scheme that maximizes the radio interface throughput according to eqn. (2.2).

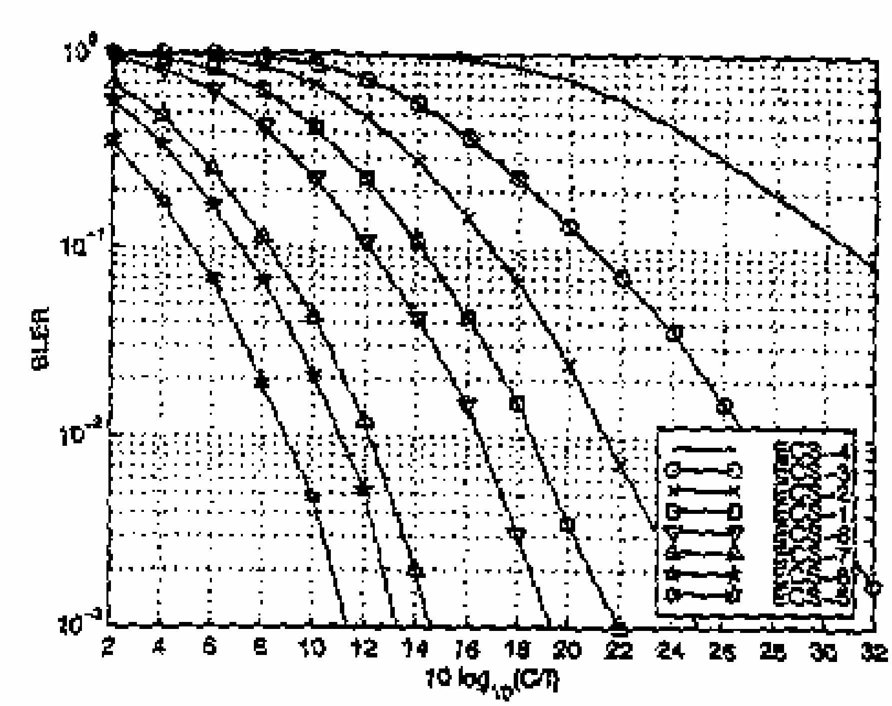


Figure 2.7: BLER vs *C/I* for EGPRS (O16QAM) with ideal FH [15]

Due to the unavailability of similar performance characteristics for 8-PSK (i.e., BLER vs C/I for PCS-1 to PCS-8), this study will focus on EGPRS using O16QAM as the high-level modulation scheme.

2.5.4.2.Link Adaptation based on Average error rate

The *average error rate* of transmitted blocks can be used to estimate the channel quality. However, since the polling interval is relatively short (e.g., 10 blocks), the *average error rate* can only be obtained as a coarse estimate.

Data blocks in each polling interval are transmitted with an initial coding scheme, e.g., with the highest data rate [12]. At the receiver, the *Actual_C/I* is used to determine the BLER of each block, based on Figure 2.4. A uniform random variable, x, is defined and the error flag of each block is set according to the following rule:

if $x > BLER$ $error\_flag$ = FALSE
else $error\_flag$ = TRUE

The average error rate is then computed based on the following equation:

$$Average\_error\_rate = \frac{Number\_of\_blocks\_with\_error\_flag\_set}{Total\_number\_of\_blocks\_per\_polling\_interval} \quad (2.3)$$

The value computed in eqn. (2.3) is sent back to the sender via the ARQ Status Feedback message. If this value is within 10%, a scheme with higher data rate is selected. If it is beyond 20%, a scheme with lower data rate is selected for the next polling interval. Otherwise, the current coding scheme is retained for the next interval.

2.6. Basic EGPRS/GPRS Operation

Consider the transmission of packet data across a EGPRS/GPRS network (comprising Gateway GPRS Support Nodes (GGSN), Serving GPRS Support Nodes (SGSN), and Base Station Subsystems (BSS)) between a Remote Host, and a Mobile Station (MS). Figure 2.8 illustrates the protocol structure for the scenario described.

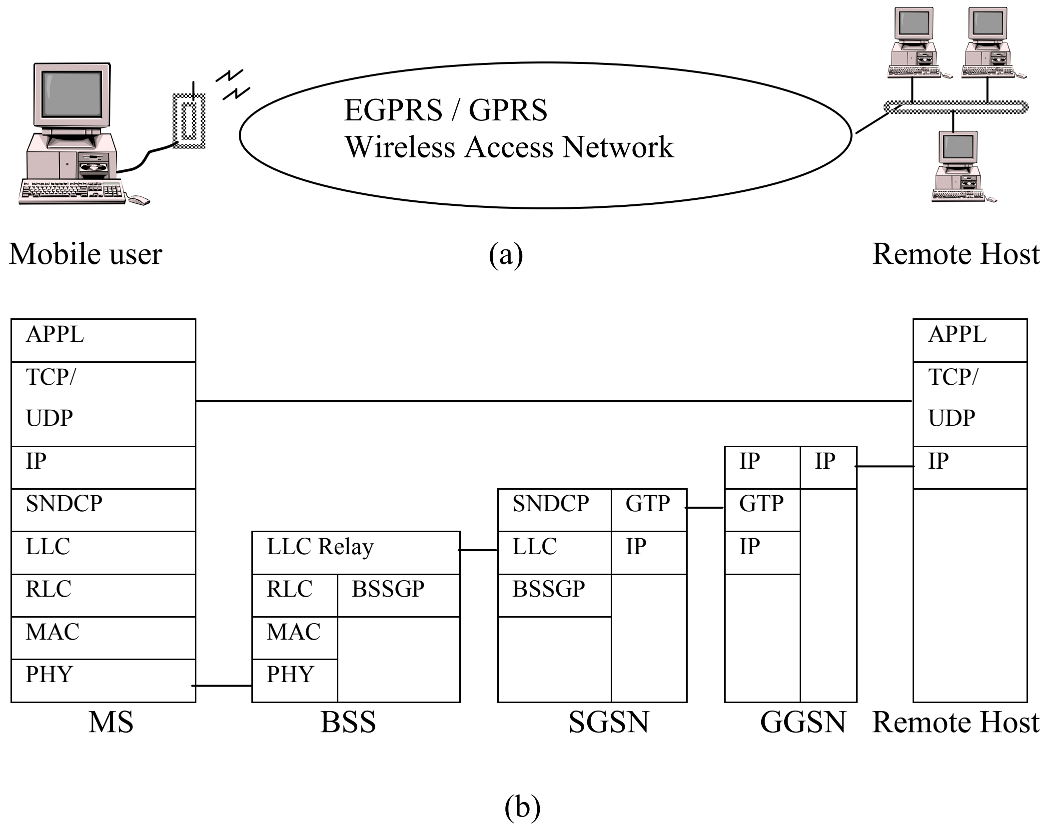


Figure 2.8: (a) Packet data transmission scenario via EGPRS / GPRS network between Mobile Station and Remote Host and (b) Protocol Structure for given scenario

2.6.1. Packet transformation data flow

The packet transformation data flow is shown in Figure 2.9.

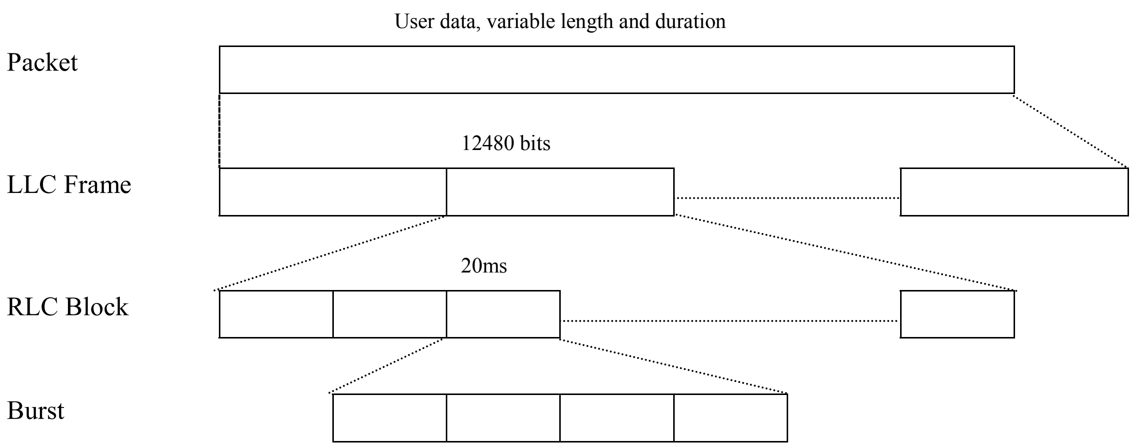


Figure 2.9: Packet transformation data flow in EGPRS/GPRS

The network-layer protocol data units (packets) received from the network layer (i.e., IP layer) are transmitted across the air interface between the MS and SGSN using the LLC protocol. First, the Sub-Network Dependent Convergence Protocol (SNDCP) transforms packets into LLC frames. The process includes optional header/data compression, segmentation and encryption. The maximum size of an LLC frame has been specified to be 1560 bytes.

The LLC layer then segments each LLC frame into RLC data blocks that are formatted into the physical layer based on the radio interface parameters described previously. The RLC flow control protocol with selective ARQ described is implemented at the RLC layer to provide a reliable wireless link between the BSS and the MS, while the link quality control strategies adapts the transmission format to the channel quality to ensure optimal throughput levels.

The various layers described thus far do not distinguish amongst the users to which the data blocks belong. The MAC layer is responsible for allocating radio resources to transmit RLC blocks from the BSS and destined for different destination MS or vice versa, and is described next.

2.6.2. EGPRS/GPRS MAC protocol

As a hybrid frequency-division / time-division system, GSM organizes radio transmissions by assigning carriers and time slots to logical channels. The frame duration is 4.615ms, and this is divided into eight TSs, as shown in Figure 2.10.

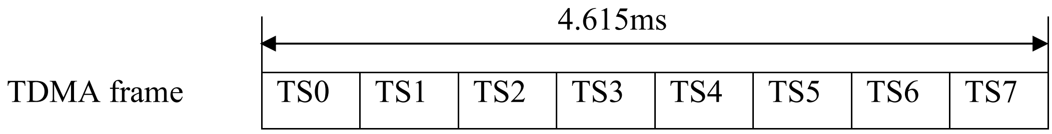


Figure 2.10 GSM TDMA frame structure

A special multi-frame structure has been defined for GPRS/EGPRS, where each frame is a GSM frame. A PDCH is defined by specifying (TS, GSM frequency) and each RLC block undergoes rectangular interleaving over four bursts in consecutive TDMA frames. Figure 2.11 illustrates the definition of a PDCH based on the multi-frame structure using TS1. RLC block *m* is interleaved over four bursts in TDMA frames n, n+1, n+2 and n+3.

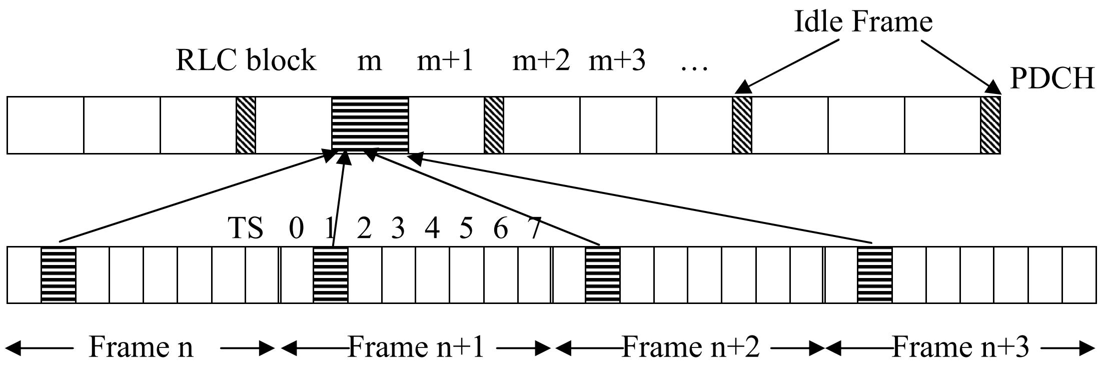


Figure 2.11: Illustration of 52 Multi-frame structure for EGPRS/GPRS

In GPRS, the radio subsystem is required to support a number of logical channels that can be categorized as traffic and control channels. Different packet data logical channels can occur on the same PDCH. These logical channels and their functionality are described in more detail in [17].

To illustrate the concept of logical channels, let's consider a mobile terminated packet transfer. Prior to any data transmission, a connection set-up procedure has to be established between each sender-receiver pair. Assuming the MS is in Standby mode, the BSS sends a Packet Paging Request (PRQ) on the down-link Packet Paging Channel (PPCH or PCH). The MS responds to the PRQ by initiating a procedure for Packet Paging Response (PRP).

This procedure comprises making a Packet Channel Request (CRQ) on Packet Random Access Channel (PRACH). The BSS responses by sending a Packet Channel Assignment (CAS) message on the Packet Access Grant Channel (PAGCH). The down-link assignment message includes the list of PDCH(s) that will be used for down-link transfer. The MS then sends a PRP on the Packet Access Common Channel (PACCH) to complete the paging process. The connection set-up procedure is illustrated in Figure 2.12

The BSS then begins to send RLC blocks on the assigned PDCHs, as shown in Figure 2.13. Multiplexing of RLC blocks destined for different MSs on the same

PDCH is enabled using a flow identifier. Periodically, the MS sends Packet Downlink ACK/NACK messages on the PACCH with BSS-initiated polling.

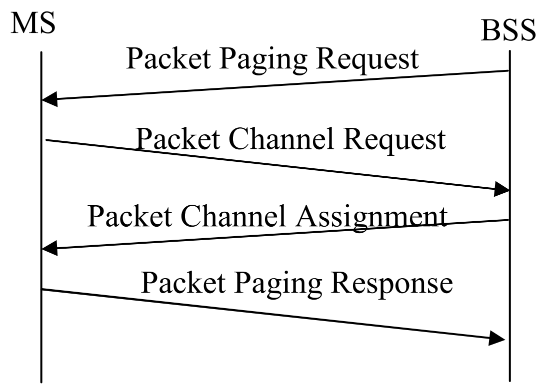


Figure 2.12: Connection set-up exchange for down-link transfer

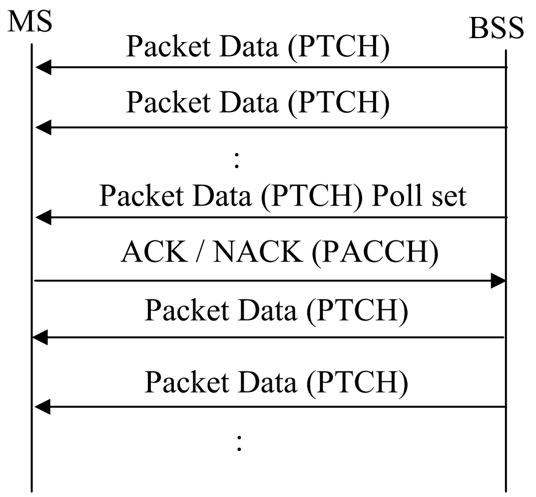


Figure 2.13 Down-link data transfer

Chapter 3

Network Simulator (*ns*)

This chapter provides an introduction to Network Simulator (*ns*) as well as a description of main components. The advantages offered by *ns* over other simulation packages are explored. Next, the wireless extension and its current implementation are described.

3.1. Introduction to *ns*

ns is a discrete-event simulator tool developed under the collaboration between researchers at University of California, Berkeley, Lawrence Berkeley Laboratories, University of Southern California and Xerox PARC. The simulator is written in *C++* and uses *Otcl* as a command and configuration interface [13].

Figure 3.1 shows an example of a simple network topology, which comprises nodes, links and traffic sources. These form part of the simulation object space in *ns*, and are defined in a hierarchical manner in *C++*. The user realizes this network simulation scenario in *ns* by creating these objects through the *Otcl* Interpreter using a simulation script in *Tcl*. Plotting and visualization tools can be used to display the output from the simulation. An example *Tcl* script can be found in [13].

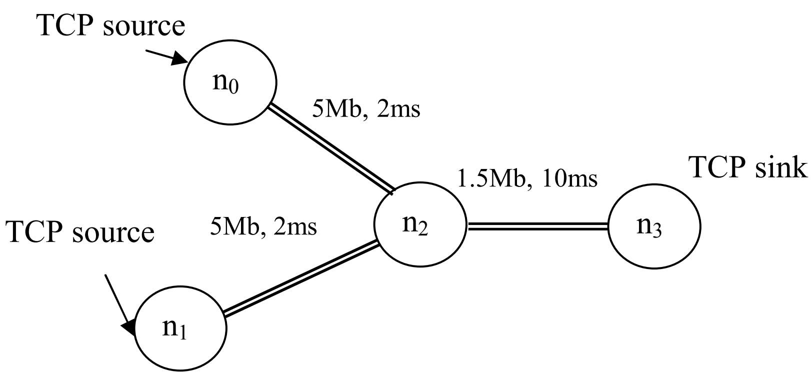


Figure 3.1: Example of a simple network topology

3.2. Description of Main Components of *ns*

Each simulation written in a *Tcl* script requires an instance of the Simulator class in *ns* to control and operate that simulation. This class provides instance procedures to create and manage the topology, and internally, store references to each element of the topology. Several important object classes in *ns* used in our simulation will be described in the following subsections.

3.2.1. Scheduler

ns is an event-driven simulator. The *ns* scheduler runs by selecting the next earliest event, executing it to completion and returning to execute the next event.

3.2.2. Timers and Timer-handlers

Timers are important objects in *ns* and they can be used to trigger periodic events, e.g., slotted TDMA transmissions at the MAC layer to the channel. A timer is scheduled to expire after a given lapse of time, and upon expiry, some function can be triggered, e.g., packet transmission and the timer can be rescheduled for the process to be repeated.

3.2.3. Nodes

One aspect of creating a network topology in *ns* is creating the nodes. The function of a node when it receives a packet is to examine its field, and then maps the values to an outgoing interface object that is the next downstream recipient of this packet.

3.2.4. Links

Once the nodes have been created, they have to be connected to form links to complete the topology. *ns* supports various kinds of links, including point-to-point wired link, multi-access Local Area Network (LAN), wireless and other broadcast media.

3.2.5. Error models

These are objects attached to links to introduce packet losses into a simulation.

3.2.6. Agents

Agents represent endpoints where network layer packets are constructed or consumed, and are used to implement protocols at various layers. An example of a transport agent is a TCP source / sink.

3.2.7. Applications

Applications sit on top of transport agents, e.g., traffic-generators, which generate data to be sent across the end-to-end connection.

3.2.8. Trace and Monitoring Support

Trace agents can be attached to each layer to record each individual packet as it arrives, departs or is dropped at that layer. The simulation output thus collected can then be displayed graphically / visually.

3.3. Wireless Extensions to *ns*

ns is often growing to include new protocols. In 1998, Carnegie-Mellon University (CMU)'s Monarch Group's mobility extension to *ns* provided new elements at the physical, link and routing layers of the simulation environment. With these elements, it is possible to construct detailed and accurate simulations of wireless subnets, LANs or multi-hop networks. The original CMU model has been recently extended [13] to allow combined simulation of wired and wireless networks.

3.4. Why *ns*?

ns is selected amongst the many simulation packages available, e.g., Opnet, BoNeS simulator etc because of the following reasons. As it is distributed as a freeware on the network, it is freely downloadable and it boasts a large pool of users, mainly students and researchers. The associated mailing list also has a subscriber volume exceeding 400, where active discussions take place and innovative ideas are exchanged. In this way, on-line help is easily available, which shortens the learning curve to the new user and facilitates research activities using this software.

In addition, with the wireless extensions available, it becomes a particularly attractive tool to simulate EGPRS since redundant re-engineering efforts can be avoided and development efforts will be efficiently employed for modification of existing codes and addition of new modules.

3.5. Description of *ns* Wireless Node Structure

Figure 3.2 gives an overview of how wireless nodes are connected together. Each wireless node is an independent entity responsible for computing its own position and velocity as a function of time. Each wireless node can have one or more network interfaces, each of which is attached to a channel. Conceptually, a channel represents a particular radio frequency with a particular modulation and coding scheme.

When a wireless node transmits a packet into the channel, the channel distributes a copy of the packet to other network interfaces on the channel. These interfaces then use a radio propagation model to determine if they are actually able to receive the packet.

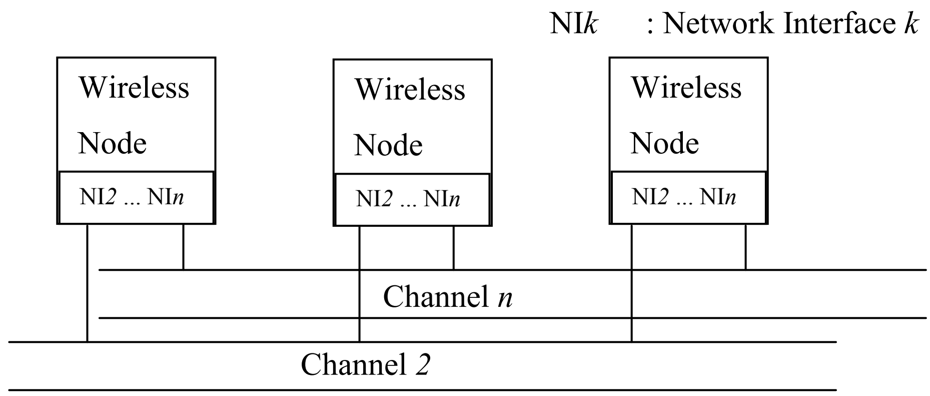


Figure 3.2: Illustration of how wireless nodes are connected together

Figure 3.3 shows the basic schematic layout of a typical wireless node. Section 3.5.1 and 3.5.2 describe the packet flow in a wireless node.

3.5.1. Outgoing packets (Into channel)

Packets sent by a source on the wireless node are handed to the node's entry point, which passes them to the address demultiplexer (addr demux). The addr demux tests the destination IP address of the packet, and if it matches the node's own address, passes the packet up to the port demultiplexer (port demux). Since packets sent by a source are typically destined to another node, most packets will match the default target of the addr demux and be handed down to the Routing Agent (Rtagent).

The packet is next passed to the Link Layer (LL), which queries the Address Resolution Protocol (ARP) object to translate the IP address to a hardware address. The packet is then inserted into the Interface queue (IFq). The MAC object takes packets from the head of the IFq and sends them to the Networking Interface

(NetIF) when appropriate, depending on the MAC protocol employed. The NetIF stamps the common header of the packet with properties such as power and position of the transmitting interface, and then passes the packet to the channel, where a copy is made for all the other interfaces on the channel.

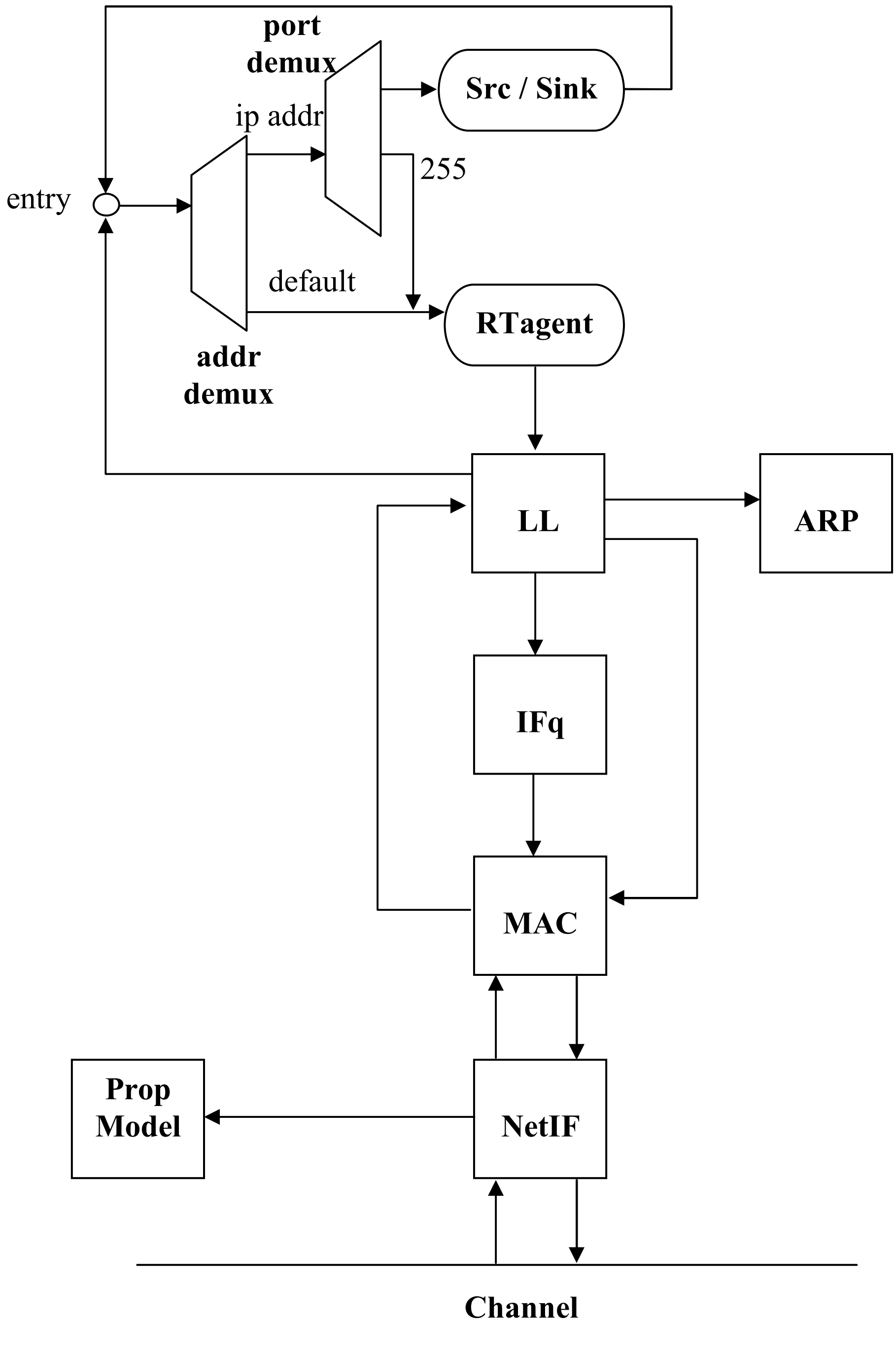


Figure 3.3: Schematic of a wireless node [14]

3.5.2. Incoming Packets (From channel)

Each NetIF stamps the packet with the receiving interface's properties and then invokes the Propagation Model (Prop Model). The Prop Model uses the transmit- and receive- stamps and the properties of the receiving interface to determine the power with which the interface will receive the packet. The receiving network interfaces then use their properties to determine if they can successfully receive the packet, and hand the packet to their MAC layer if appropriate. If the MAC layer receives the packet collision- and error-free, it passes the packet to the wireless node's entry point.

If this is the final destination of the packet, the addr demux will pass the packet to the port demux, which will hand the packet over to the default target of the addr demux. Otherwise, the Rtagent at this node will be called to assign the packet a next hop and pass the packet back to the LL.

3.6. Current Implementation of various layers in a Wireless Node

3.6.1. Channel

The radio propagation model uses Friss free-space attenuation ($1/r^2$) at near distances and an approximation to 2-Ray Ground ($1/r^4$) at far distances. The approximation assumes specular reflection off a flat ground plane. A unity gain omni-directional antenna is assumed.

3.6.2. NetIF

A shared media model is implemented, where, subject to collisions and the Prop Model, each node can overhear packets transmitted by the others. The default parameters are taken to approximate the Lucent WaveLAN Direct-Sequence-Spread-Spectrum (DSSS) radio interface [14].

3.6.3. MAC protocols

Wireless simulations in *ns* currently support Null-MAC, IEEE 802.11 and a few other protocols. Null-MAC does not perform any form of collision detection, and it just schedules the next packet immediately following the end of the current transmission. Collided packets will have their error flags set and will not be received correctly.

The IEEE 802.11 standard [21] specifies a Carrier Sense Multiple Access with Collision Avoidance (CSMA / CA). In this protocol, when a node receives a packet to be transmitted, it first listens to ensure that no other node is transmitting. If the channel is clear, it then transmits its packet. Otherwise, it chooses a random back-off factor, which determines the amount of time the node must wait till it is allowed to transmit its packet. During periods in which the channel is free, the transmitting node decrements its back-off counter. When the back-off counter reaches zero, the node transmits the packet. Since the probability that two nodes will choose the same back-off factor is small, collisions between packets are minimized.

Chapter 4

EGPRS SIMULATION DESIGN AND IMPLEMENTATION

This chapter describes the simulation environment and the level of abstraction and simplification in the design of the simulation model. The mapping of EGPRS protocol structure into *ns*-wireless node structure is next described, followed by detailed descriptions of the design and implementation in terms of new / modified functions to existing *ns* modules.

4.1. Simulation Network Configuration

This study focuses on the end-to-end performance of Internet access via the EGPRS network. The simplest configuration for such a scenario is shown in Figure 4.1, which shows end-to-end communications between a single MS and a Remote Host over an EGPRS network, comprising the GGSN, SGSN and the BSS. These elements have been described in Section 2.3.

In *ns*, each of the logical entities in the top level configuration shown in Figure 4.1 can be realized as nodes and links are defined to connect these nodes, as shown in Figure 4.2. The Remote Host, GGSN, and SGSN are defined as wired nodes. The (Remote Host, GGSN), (GGSN, SGSN) and (SGSN, BSS) are connected by duplex links. The BSS and MS are defined as wireless nodes and they are linked via a wireless channel. Delays and packet losses introduced by the IP Network can be modeled in *ns* by attaching an Error Model on the link between the Remote Host and GGSN.

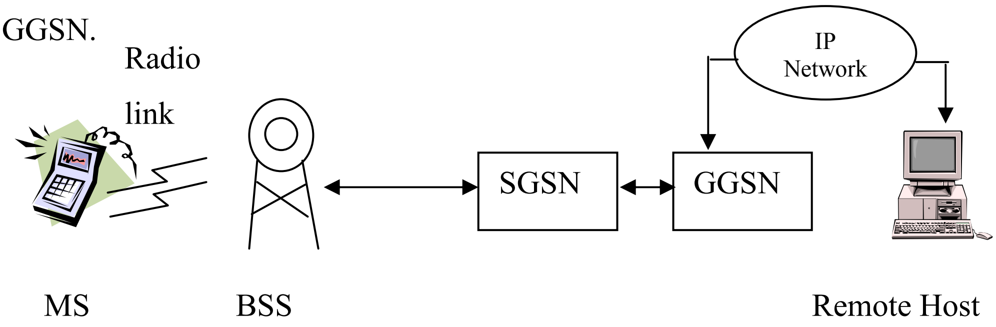


Figure 4.1: EGPRS top level simulation environment

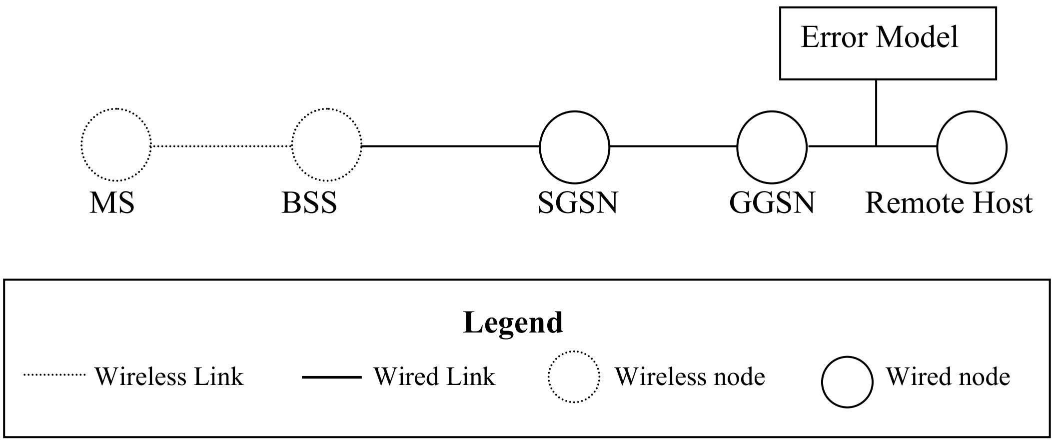


Figure 4.2: Realization of top level simulation environment in *ns*

From this level, the configuration is decomposed into blocks representing the internal structure of each node, i.e., the EGPRS protocol stacks and the corresponding peer-to-peer communication link, as shown in Figure 4.3.

4.2. Level of Abstraction for EGPRS Protocol Implementation

The level of abstraction for implementing the EGPRS protocol structure in *ns* is shown in Figure 4.4.

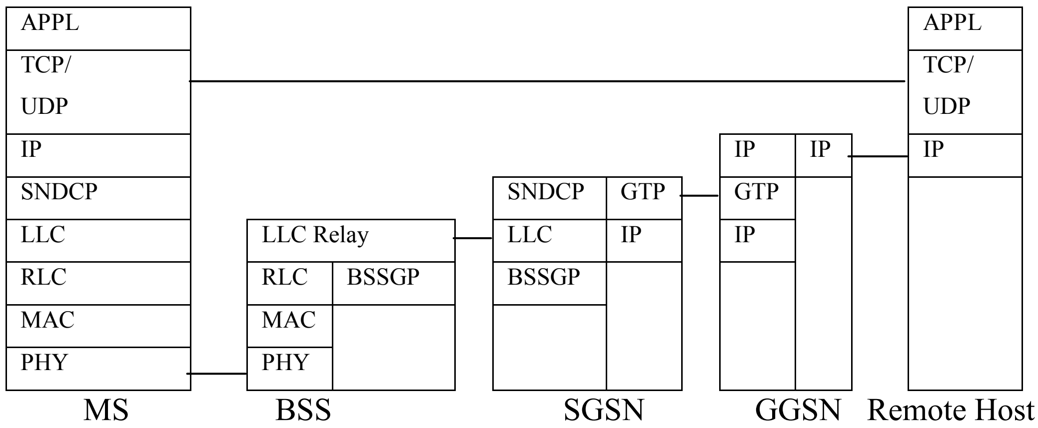


Figure 4.3: EGPRS protocol structure

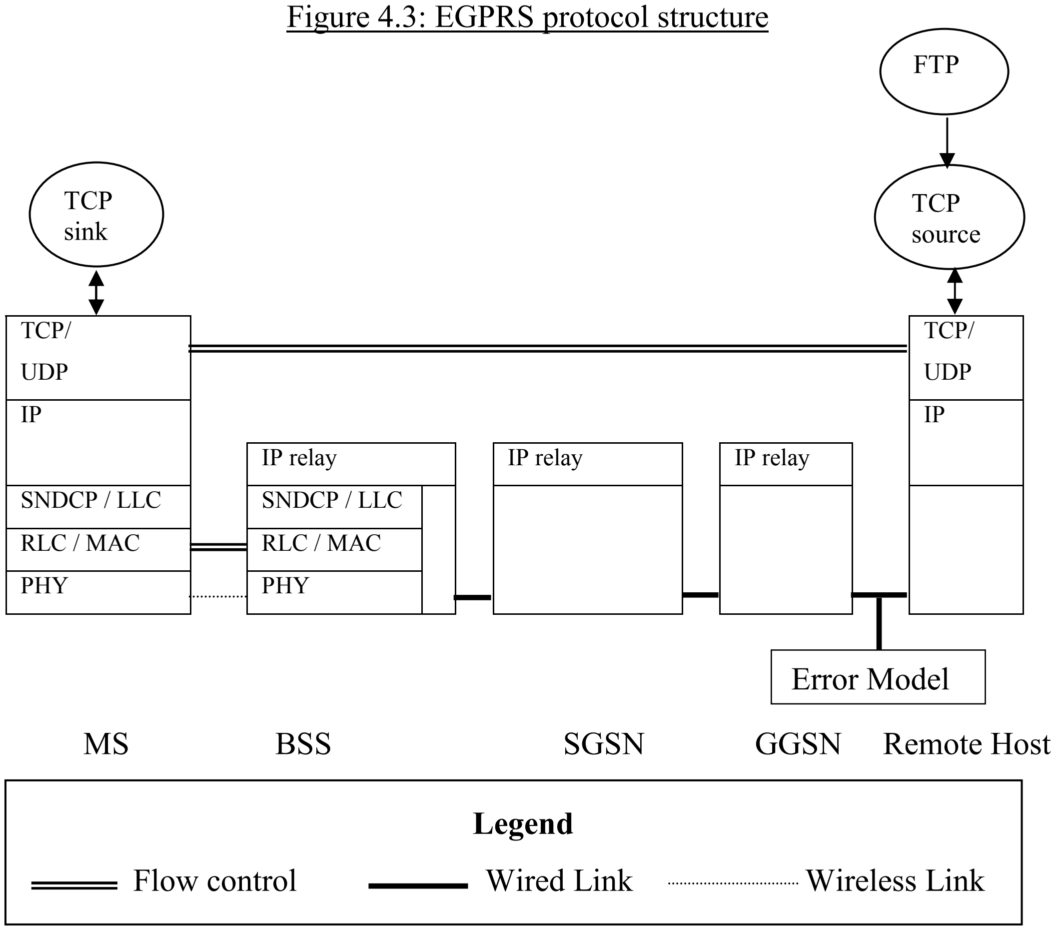


Figure 4.4: Implementation of EGPRS protocol structure in *ns*

4.2.1. Application and User behavior

The application layer is implemented by attaching an application object, e.g., FTP, to the TCP agent defined for the end nodes, i.e., Remote Host or MS. In our simulation, the FTP object is defined at the Remote Host to support uni-directional transfer of bulk data from the Remote Host to the MS.

4.2.2. TCP

TCP agents (source or sink) are defined at the end nodes to implement end-to-end flow control. In our simulation, one-way TCP version Tahoe is used. TCP segment size is set to 512 bytes and a window size of 16 segments is used.

4.2.3. IP

For uni-directional transfer of data from the Remote Host to the MS, TCP segments at the Remote Host are sent across to the GGSN via IP. These IP packets are sent across to the SGSN via IP. They are fragmented by the SNDCP at the SGSN and sent to the BSS via LLC relay. In our simulation, IP relays are defined for all the mentioned wired links for simplification, and each link is characterized by its bandwidth and processing delay.

4.2.4. SNDCP

Fragmentation of IP packets into LLC frames (with maximum size of 1560 bytes) is supported, while TCP / IP header compression is not supported in our simulation.

4.2.5. LLC protocol

The EGPRS attach procedure is not implemented and hence, the MS starts in stand-by mode. This implies that data transfer can commence once the connection is set–up (as described in Section 2.4.4). In addition, it is assumed that LLC operates in UNACKnowledged (UNACK) mode, i.e., it does not take care of packet losses. Fragmentation of LLC frames into RLC blocks is supported. However, since the maximum TCP segment size (512 bytes) is less than the maximum LLC frame size (1560 bytes), IP packets are directly fragmented into RLC blocks. The RLC block size used to carry out fragmentation depends on the coding scheme selected by the link adaptation algorithm.

4.2.6. RLC / MAC protocol

TDMA transmission using the 52-Multiframe structure for EGPRS as described in Section 2.4.4 is implemented, with some simplifications that will be described in Section 4.3.2.4. The sliding window mechanism for RLC selective ARQ and the link adaptation algorithms are implemented.

4.2.7. PHY layer protocol

All proposed coding schemes (ECS-1 to ECS-8) of EGPRS using O16QAM (Table 2.2) are supported.

4.3. Mapping of EGPRS protocols into *ns* Wireless Node

The lower layers that specify communication over the wireless link (i.e., the link between the MS and the BSS) are EGPRS-specific. Figure 4.5 shows how the EGPRS lower layer protocols (SNDCP, LLC, RLC, MAC and PHY) are implemented in *ns*. The current implementation of IFq in *ns* is used in the model.

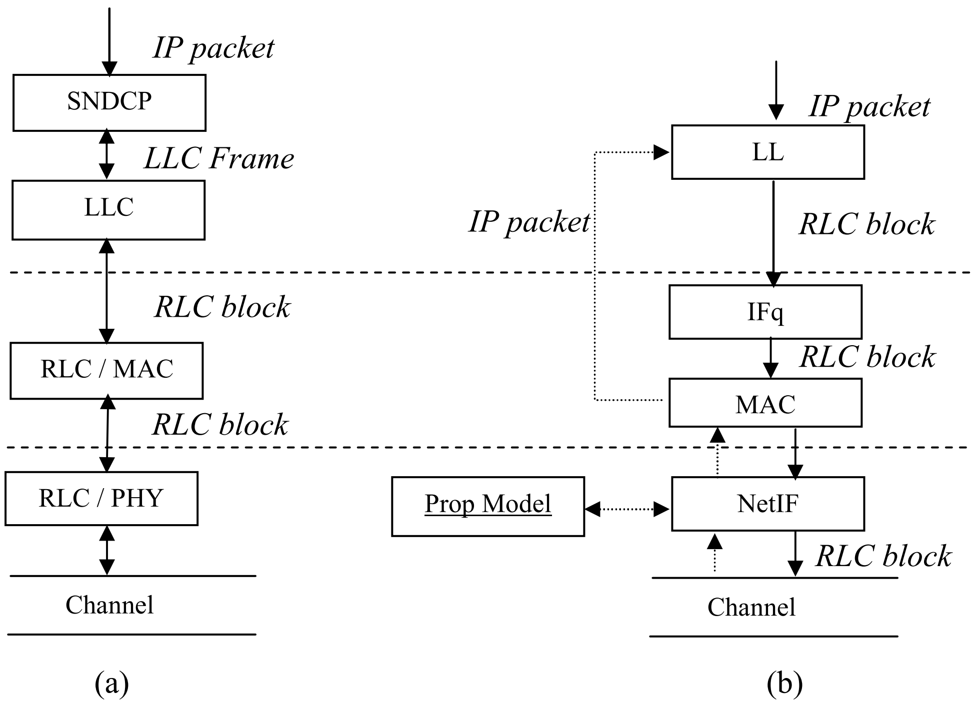


Figure 4.5: Comparison between (a) EGPRS and (b) *ns* wireless node structure

4.3.1. LL

The LL receives packets (*recv*) from either the RTagent or from the MAC Layer. It updates the sequence number and schedules the packet received from the RTagent to be sent down (*send down*) to the IFq. For an incoming packet, it is dropped if the error flag is set; otherwise, the packet is scheduled to reach the node after some delay (*send up*). The current implementation of these functions in *ns* is shown in Figure 4.6.

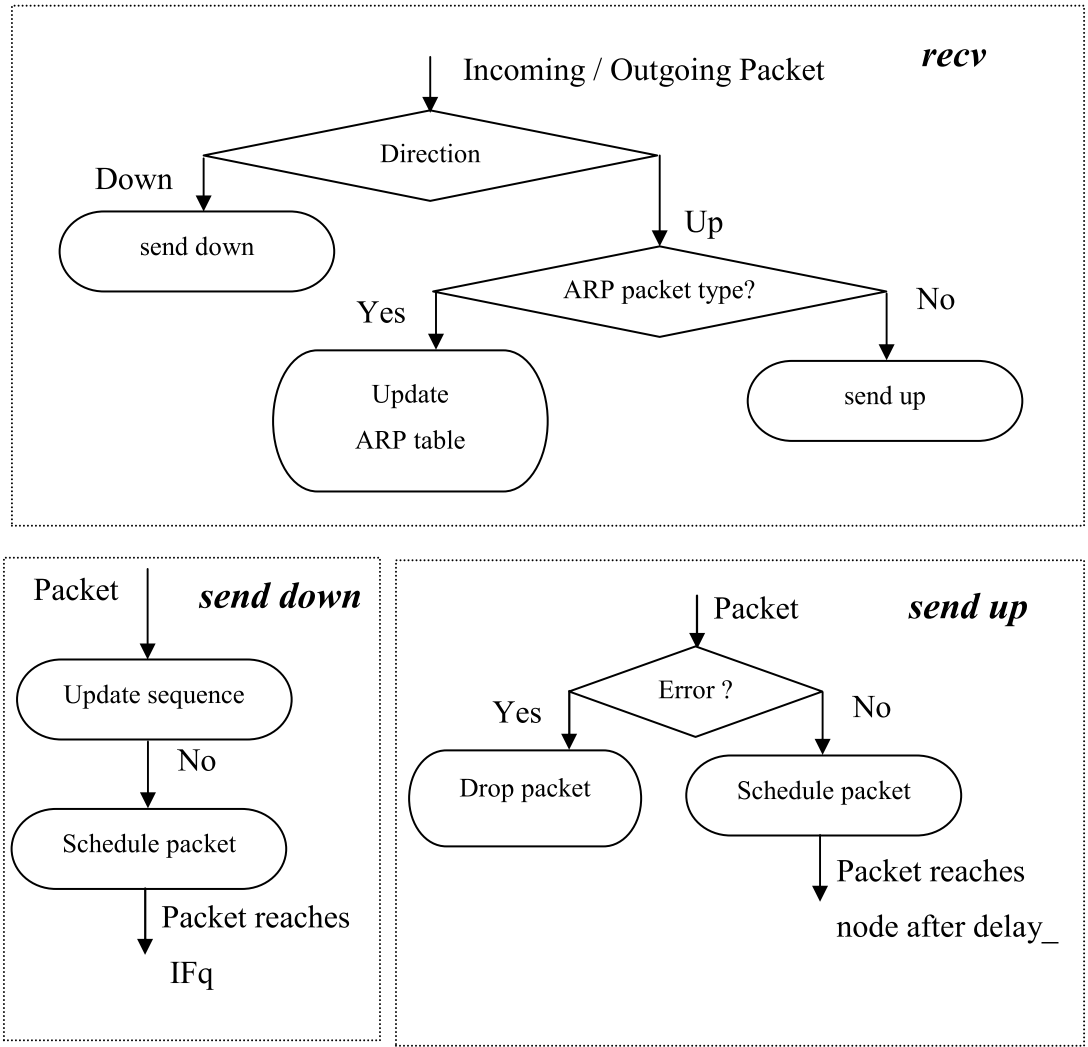


Figure 4.6: Block diagram of current LL implementation in *ns*

To implement EGPRS SNDCP and LLC functionality, the following functions are added to the LL:

4.3.1.1. Fragmentation of IP packets into RLC blocks

As described in subsection 4.2.4, IP packets are directly segmented into RLC blocks, the size of which is determined by the coding scheme selected by link adaptation. Besides the sequence number, each RLC block is also assigned a BSN. The coding scheme is also embedded in the MAC layer header. In addition, the first and last BSN of the current packet are also noted as they will be used to determine if a complete packet has been received at the receiving MAC layer. All these are implemented in the *send down* function.

4.3.1.2. Update of RLC block size via link adaptation

C/I estimates are obtained at the receiver's channel and the average value for the previous polling interval is used to compute the optimal coding scheme for the next polling interval. This is sent via an ACK block to the sender to update the RLC block size accordingly. The block size is set to the worst coding scheme at the beginning of the simulation. These are implemented in the *recv* function.

The above functions are implemented at the LL in our simulation and summarized in Figure 4.7. The *send up* function remains unchanged.

4.3.2. MAC Layer

Current implementation of MAC in *ns* supports several MAC protocols, including Null-MAC, IEEE 802.11, etc., which have been briefly described in Section 3.6.3. After evaluating the code, it is found to be more efficient to add-on to the Null-MAC to include EGPRS MAC and RLC protocols than to modify the IEEE 802.11, which is complex.

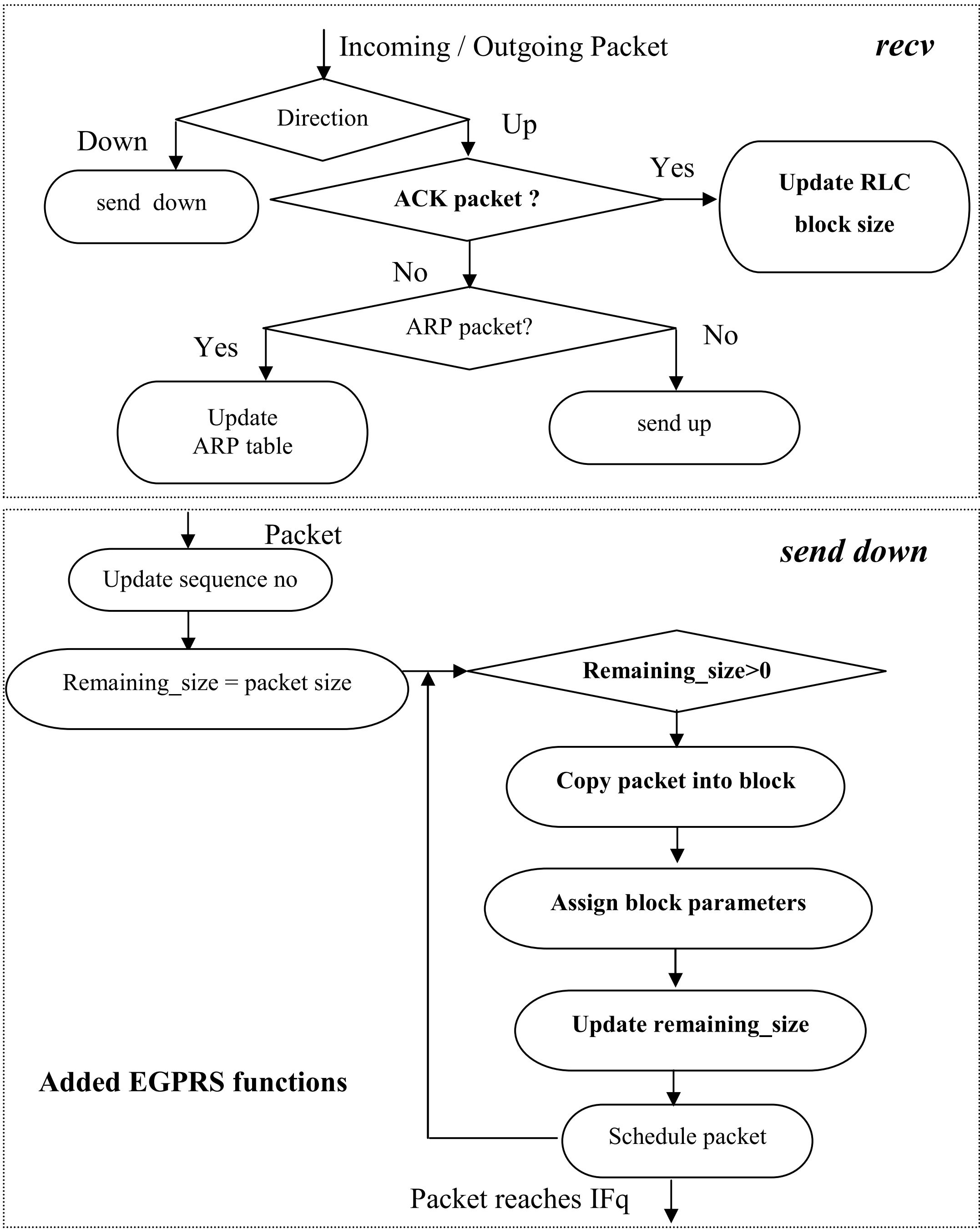


Figure 4.7: Block diagram of LL implementation with EGPRS

In the existing implementation, the MAC Layer receives packets (*recv*) from either the IFq or from the NetIF. For an outgoing packet, it schedules reception of the packet by the NetIF (*send down*) and subsequently resumes dequeuing (*resume*) the next packet from the IFq to be received by the MAC layer. For an incoming packet, if its destination address matches the node's MAC address or it is a broadcast address, the MAC layer will schedule reception of the packet at the LL (*send up*). The current implementation of these functions in *ns* is shown in Figure 4.8.

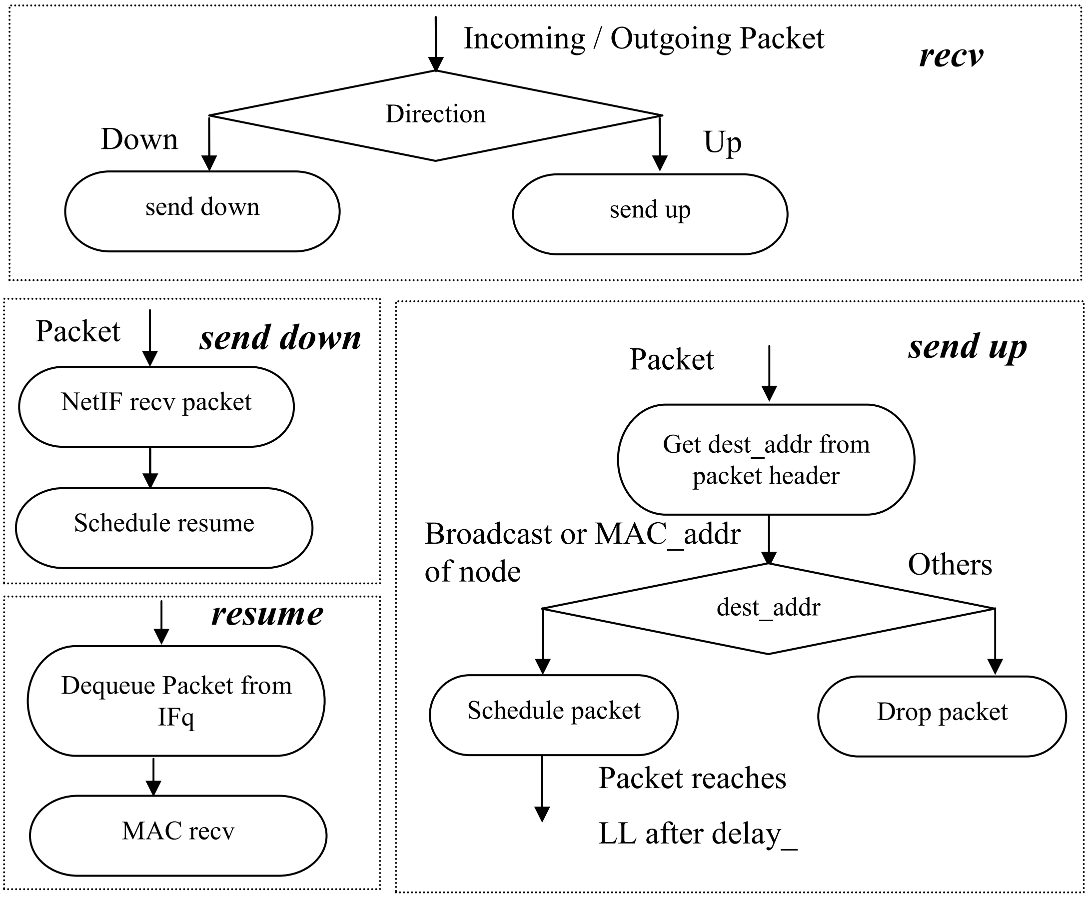


Figure 4.8: Block diagram of current MAC implementation in *ns*

In EGPRS, prior to down-link data transfer, an end-to-end connection has to be established. This is implemented at the MAC layer as described below.

4.3.2.1. Connection set-up message exchange

When the sending MAC layer receives its first data block from the LL, it holds the block, and initiates the connection set-up exchange by sending the PRQ (*sendPRQ*). The receiver responds with a CRQ message (*sendCRQ*). The sender then sends the CAS message (*sendCAS*), and the receiver sends the PRP message (*sendPRP*) to complete the connection set-up. The held block is then transmitted (*check(pktTX_)*), which sets off the down-link data transfer. This is mainly implemented in the *recv* function, as illustrated in Figure 4.9.

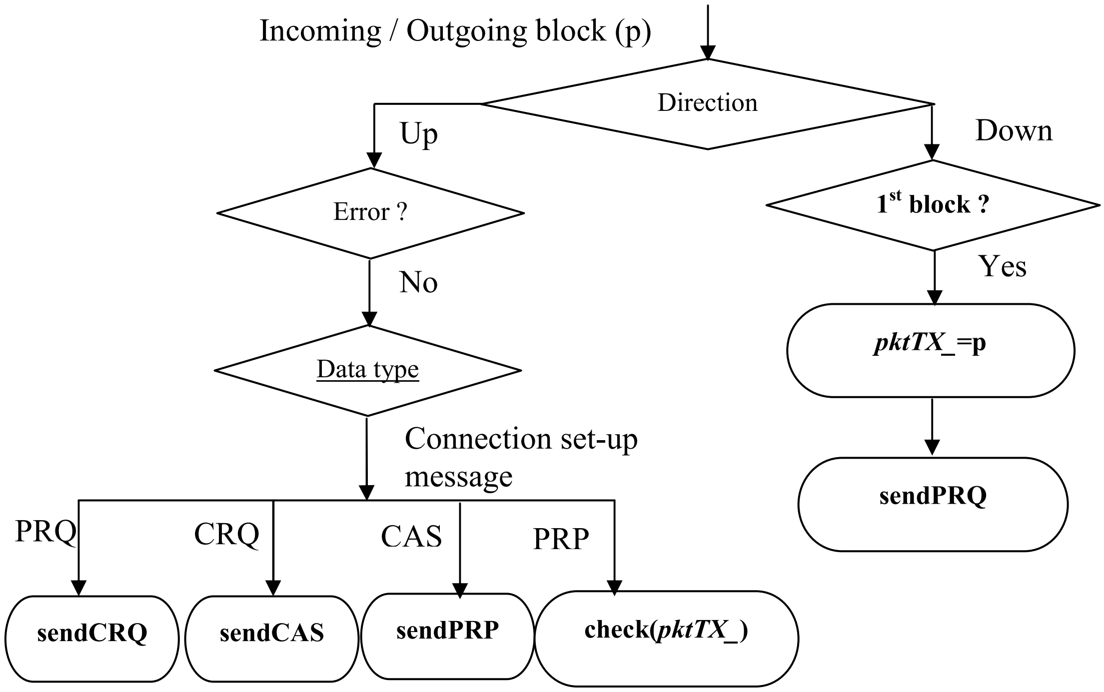


Figure 4.9: Block diagram of connection set-up exchange in MAC *recv*

4.3.2.2. RLC flow control with selective ARQ

Implementation of RLC sliding window flow control with selective ARQ in *ns* requires RLC blocks to be categorized into DATA and ACK blocks since they are processed differently.

(a) DATA transmission

Upon completion of connection set-up, transmission of data blocks commences. Each block sent into the channel assumes the status of *RLC_PACK*. The receiving node periodically sends the ACK message to update the transmission status of sent blocks into *RLC_ACK / RLC_NACK* accordingly. The send window is also updated.

As more data blocks arrive at the MAC layer from IFq, the situation may arise where blocks with status *RLC_NACK* exists from a previous polling interval. The oldest block with status RLC_NACK is assigned to the variable *NACK*. In this case, the outgoing block will be held and the block with BSN *NACK* will be transmitted. This is defined as CASE I transmission in our simulation. *NACK* will then be updated since the transmitted block status will be changed to *RLC_PACK*.

If no blocks with status *RLC_NACK* exist, but the BSN of the outgoing block exceeds the send window, the block will be held and the block with BSN *lastPACK* will be sent. This is termed CASE II transmission. *lastPACK* is then updated as the transmitted block status will be changed to *RLC_PACK*.

Otherwise, the outgoing data block is transmitted normally and this is termed CASE III transmission.

(b) ACK transmission

RLC DATA block transmission takes place in polling intervals, and the poll flag is set for the last block of each interval. The receiver, upon receiving such a polled data block, will schedule an ACK block to the sender to indicate the transmission status of blocks belonging to the last interval. This is termed CASE IV transmission, and has the highest priority over all other cases.

In addition to the definition of types of RLC blocks, additional status attributes are defined in the MAC object for the sending and receiving node [18].

(c) Sending node status attributes

An array, *TX_array*, is defined to store the status of each transmitted data block and is indexed by the BSN. All entries are initialized to NULL, and updated to *RLC_PACK* when they are transmitted. They are then updated to *RLC_ACK/RLC_NACK* accordingly when the ACK block is received. In addition, Table 4.1 lists a set of parameters to determine what type of data block to send next.

Parameter	Description	Remarks
NACK	Oldest NACK BSN	Updated every polling interval; BSN for next CASE I transmission.
lastPACK	Un-re-trans PACK BSN	BSN for CASE II transmission.
next	Next in-sequence BSN	BSN for CASE III transmission.
va	Oldest NACK / PACK BSN	Updated every polling interval; Send window = (*va* : *va* + 64)

Table 4.1: Parameters to determine type of DATA block to send next

In addition, four boolean flags and block variables are used to indicate which type of block will be scheduled next or is being scheduled, and the respective block variable which stores them, as defined in Table 4.2.

Flag	Status indication	Block variable
HOLD	A data block is being held	*pktTx_*
CTRL	An ACK block is scheduled for transmission	*pktCtrl_*
NORMAL	A data block is scheduled for transmission	*pktRx_*
POLL	A polled data block is scheduled for transmission	

Table 4.2: Status indicators type of block that will be scheduled / is being scheduled

(d) Receiving node status attributes

An array, *RX_array*, is defined to store the status of each received data block and is indexed by the BSN. All entries are initialized to *RLC_NREC*, and updated to *RLC_REC* when they are received. Another array, *ACK_array*, is updated along with *RX_array*, and is copied into the ACK message header to update the *TX_array* at the sending node for the current polling interval, after which they will be reset to RLC_NREC. The entries in *RX_array* are reset to *RLC_NREC* when all the blocks belonging to the current packet are received.

Figure 4.10 illustrates the use of the send and receive status parameters, using a polling interval of 5 blocks. Figure 4.11 shows the block diagram of the *recv* function in the MAC layer excluding the functions for connection setup message exchange.

(e) *check* function at sending node

The *check* function replaces the *send down* function in the original MAC implementation. This function is called when a block is received from the upper layer or when a held packet exists (*HOLD* = 1) at the end of a block transmission. It uses the logic in 4.3.2.2(a) to determine the type of data block to be sent next and to schedule it accordingly using the *sendDownPkt* function. The block diagram for the *check* function is shown in Figure 4.12.

Transmitted BSN	1	2	3	4	5			4	6	7	8	9			4
Received BSN		1	2	3		5				6		8	9		

				va			*NACK*			*next*					
TX_array	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	A	A	A	P	A	A	Na	P	P	0	0	0	0	0	0
RX_array	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	R	R	R	Nr	R	R	Nr	R	R	Nr	Nr	Nr	Nr	Nr	Nr

P : RLC_PACK; A : RLC_ACK; Na : RLC_NACK; R : RLC_REC; Nr : RLC_NREC

Figure 4.10: Illustration of status parameters defined at MAC layer

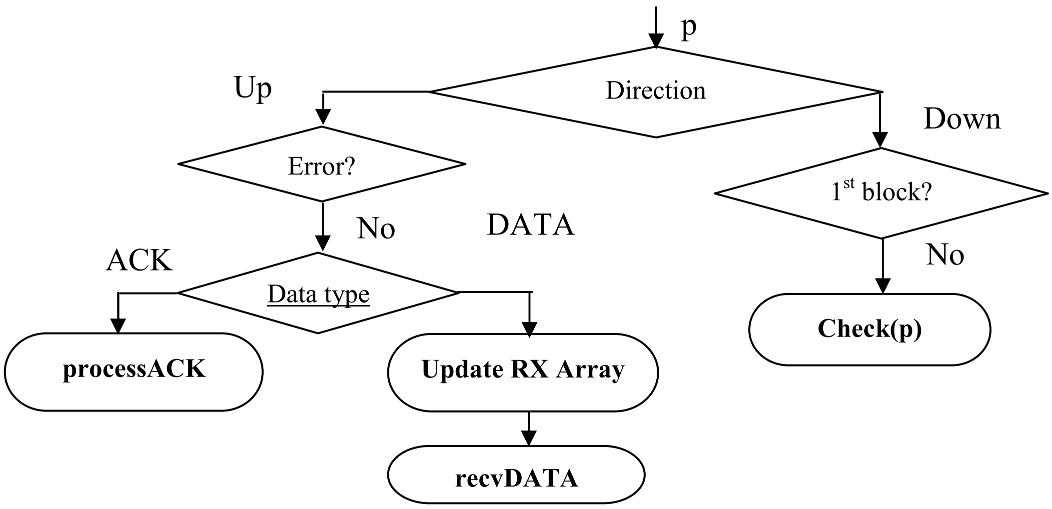
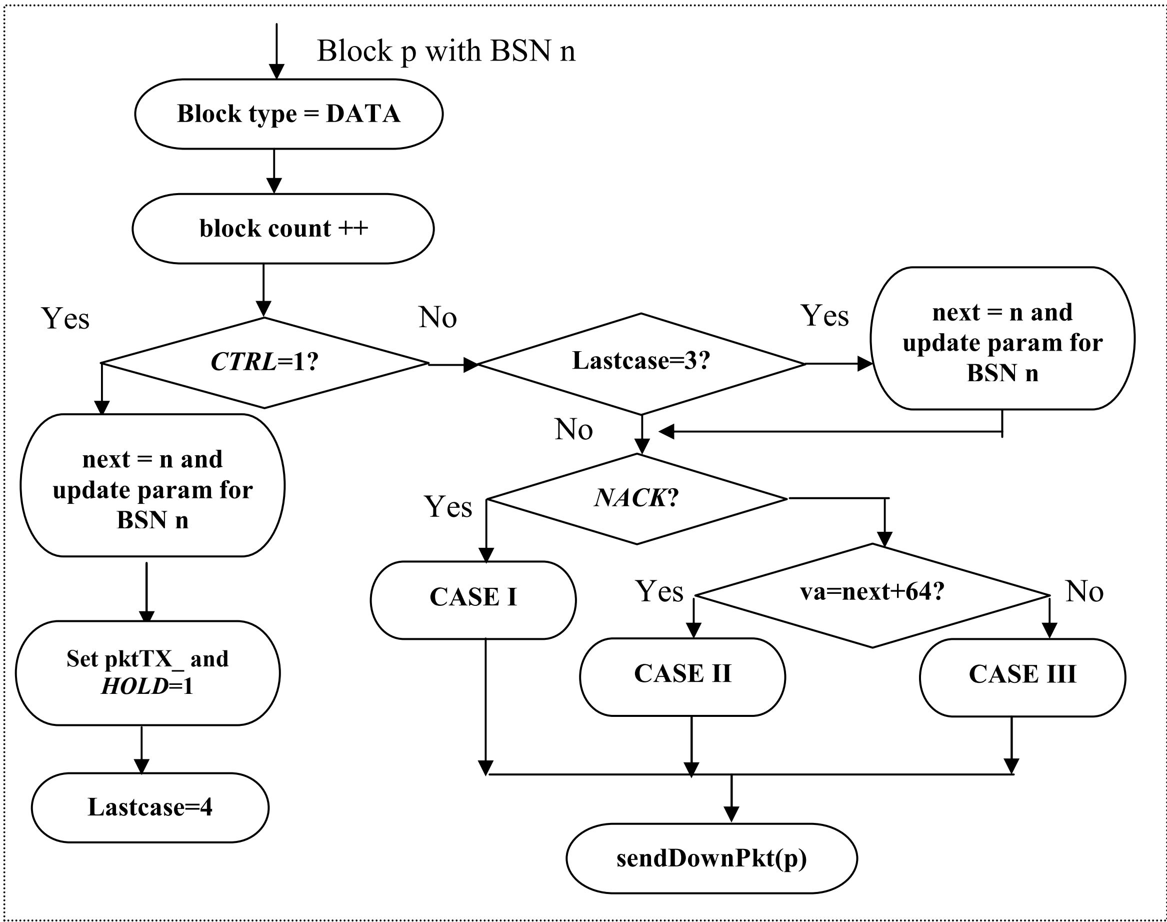


Figure 4.11: Block diagram of RLC algorithm in MAC *recv* function

(f) *recvDATA* and *sendACK* at receiving node

The *recvDATA* and *sendACK* function replaces the *send up* function in the original MAC implementation, which is now embedded in the *recvDATA* function. In *recvDATA*, data blocks are assembled into their respective IP packets and the packets are sent up to the LL (*send up*) in the order that was sent from the sending node. If the poll flag is set, *sendACK* is called which schedules an ACK message back to the sending node. The block diagram for these functions is shown in Figure 4.13.



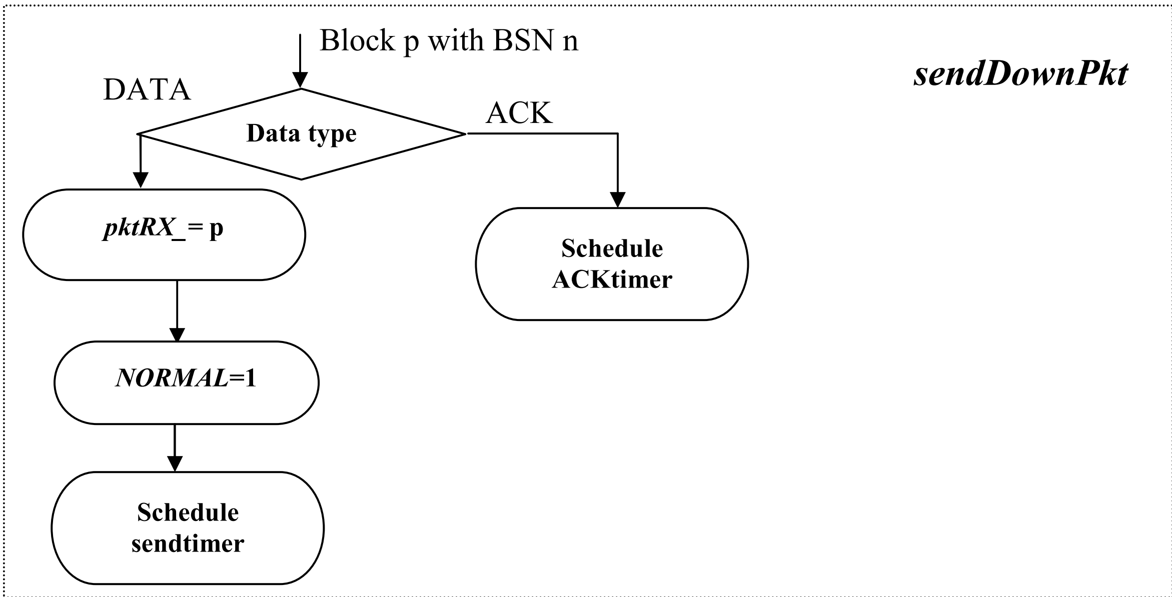


Figure 4.12(a): Block diagram of MAC *check* and *sendDownPkt* function

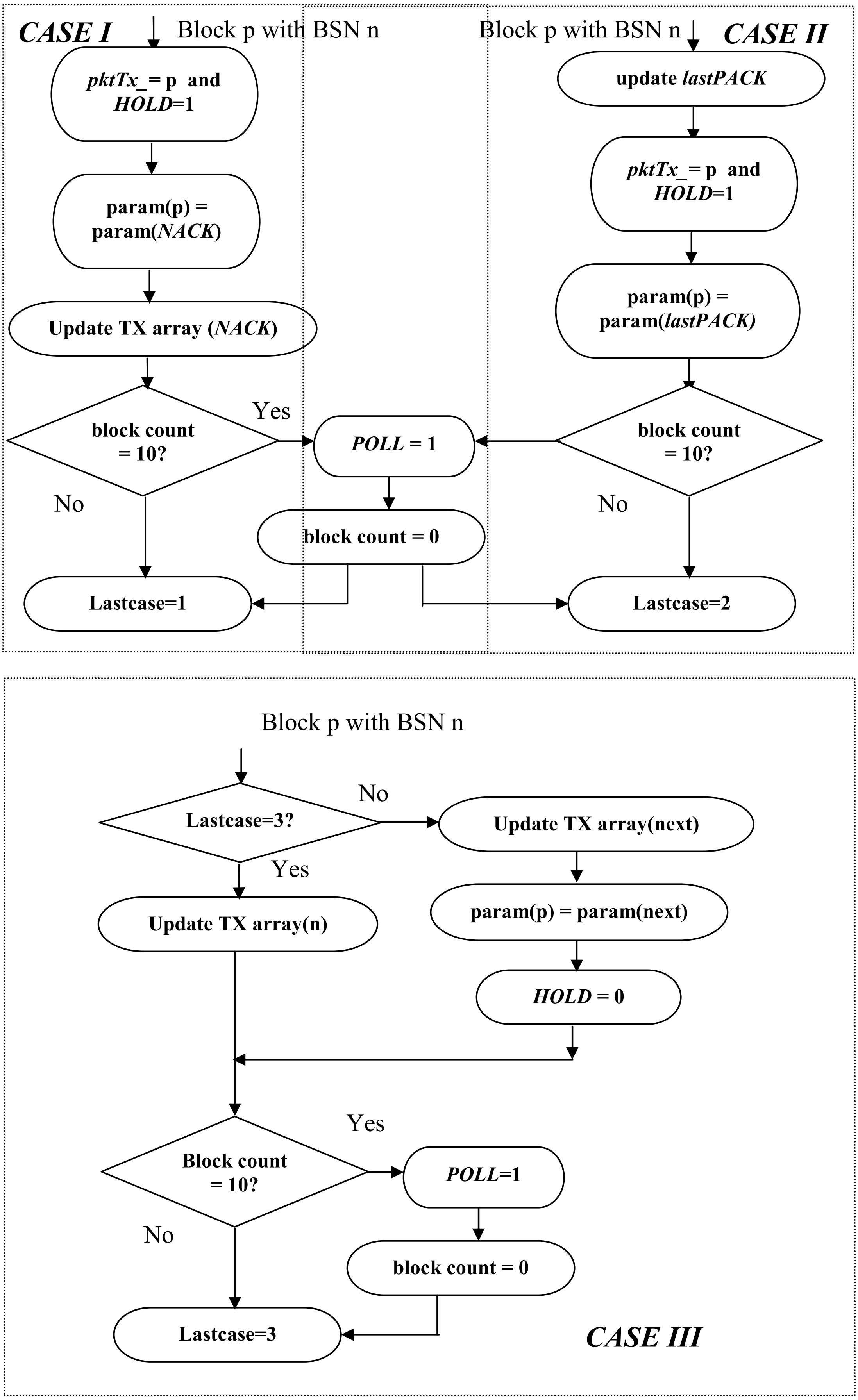


Figure 4.12(b): Block diagram of CASE I, II and III functions in MAC::*check*

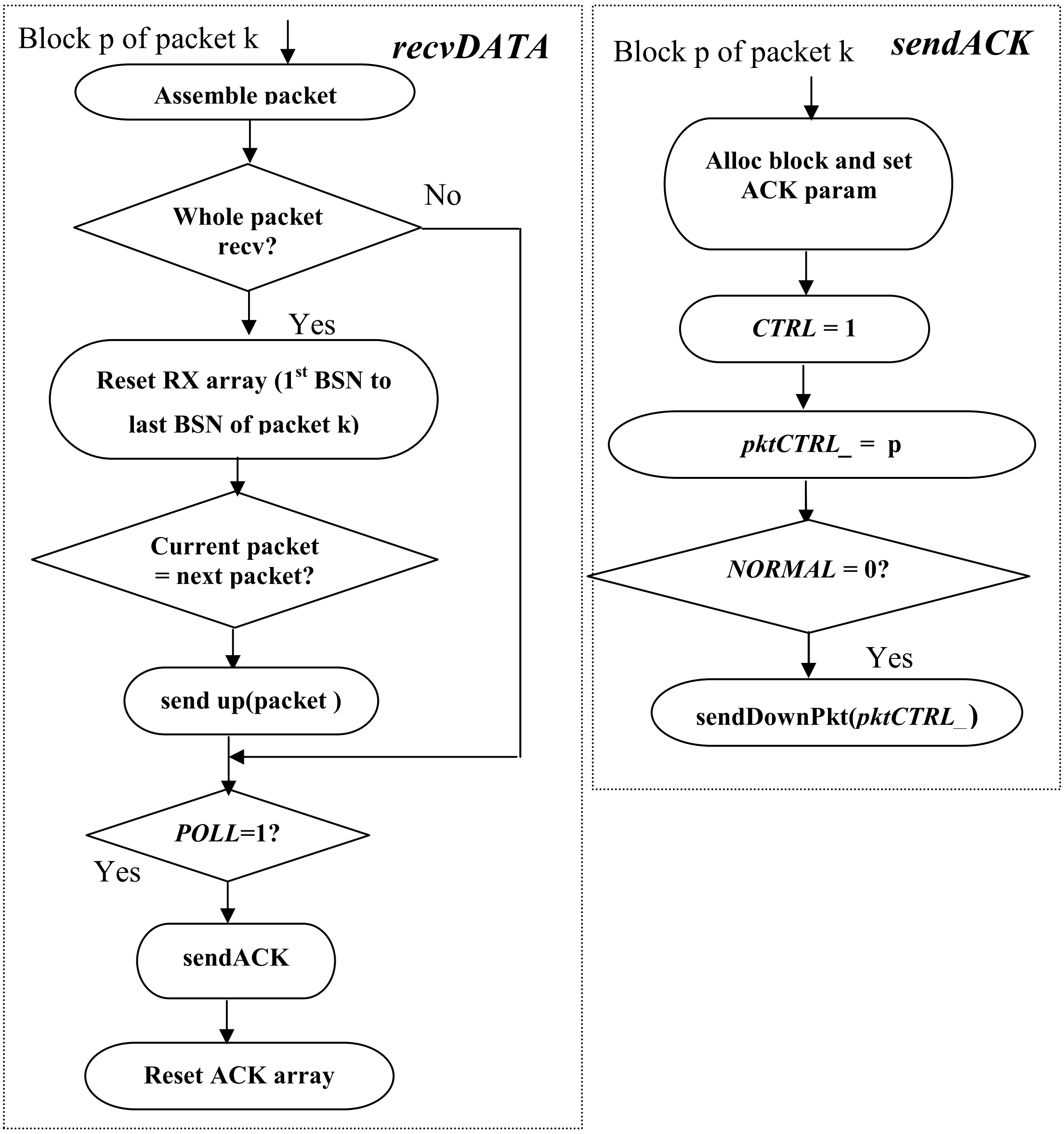


Figure 4.13: Block diagram of *recvDATA* and *sendACK* in MAC

(g) *processACK* at sending node

When the MAC layer receives an ACK block from the channel, it updates the *TX_array*, *va* as well as *NACK* for the previous polling interval. It also resets the poll flag so that transmission can resume. In addition, if link adaptation is employed, this block is sent up to the LL. These functions are shown in Figure 4.14.

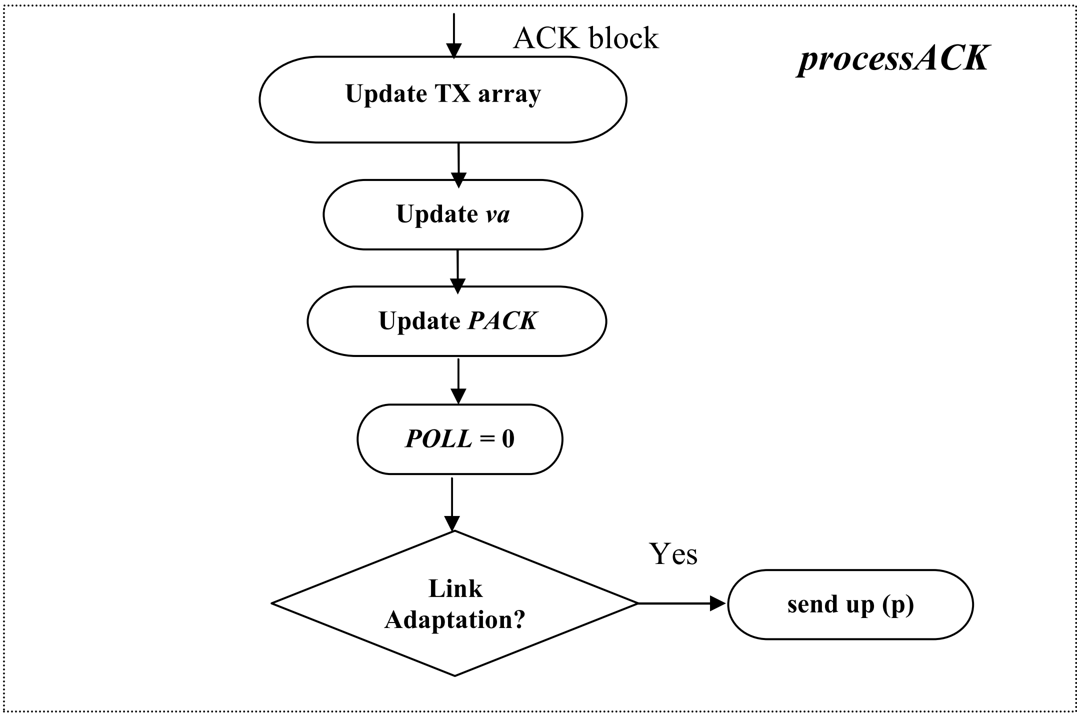


Figure 4.14 : Block diagram of *processACK* in MAC

4.3.2.3. TDMA implementation of MAC - 52 Multi-frame structure

Since only one MS is specified in our simulation scenario, there is no need for contention resolution in the up-link, i.e., CRQ messages are treated like other messages and are transmitted on PDTCH / PACCH [16]. Multi-slot operation is not considered here even though EGPRS supports this mode of operation. The throughput achievable can be theoretically computed by multiplying the single-slot throughput obtained here with the appropriate number of TS.

With dynamic allocation of radio resources, PDTCH / PACCH / PAGCH / PPCH [16] are allocated as and when needed. Hence, there is no distinction of logical channel type and all TDMA frames are treated as identical.

Several modifications were made to the scheduler as well as MAC layer to implement the TDMA frame transmission.

(a) Scheduler modifications

A new attribute, *start_frame*, is defined in the scheduler class to represent the start of the next available TDMA frame. The start of the Multi-frame structure is specified in the main *tcl* file for which the simulation is specified and *start_frame* is updated at every frame interval, *frame_width*, (20ms in our case) with the use of timer-handlers.

(b) MAC Modifications

Currently, each block which arrives from the IFq to the MAC layer is sent down to the channel immediately, and reaches the channel after some transmission delay. With the TDMA Multi-frame structure implementation, blocks can only be sent down at the start of each frame.

This is implemented in MAC by defining timer-handler objects to handle several types of blocks. As seen in Figure 4.11(a), the *sendDownPkt* function schedules the block transmission to the start of the next frame using either *sendtimer* or *ACKtimer* for DATA and ACK blocks respectively. The send functions defined in Section 4.3.2.1 for connection set-up messages schedules these messages using *ACKtimer*.

When *sendtimer* expires, the block is sent to the NetIF. If an ACK block is pending, i.e., *CTRL* = 1, it will be scheduled next. Otherwise, if a held block exists, i.e., *HOLD* = 1, it will be scheduled next; else, the next block will be dequeued from IFq. When *ACKtimer* expires, the ACK block is sent to the NetIF, and the status flag, *CTRL*, will be reset, i.e., *CTRL* = 0. If *HOLD* = 1, it will be scheduled next; else, the next block will be dequeued from IFq. These functions are illustrated in Figure 4.15.

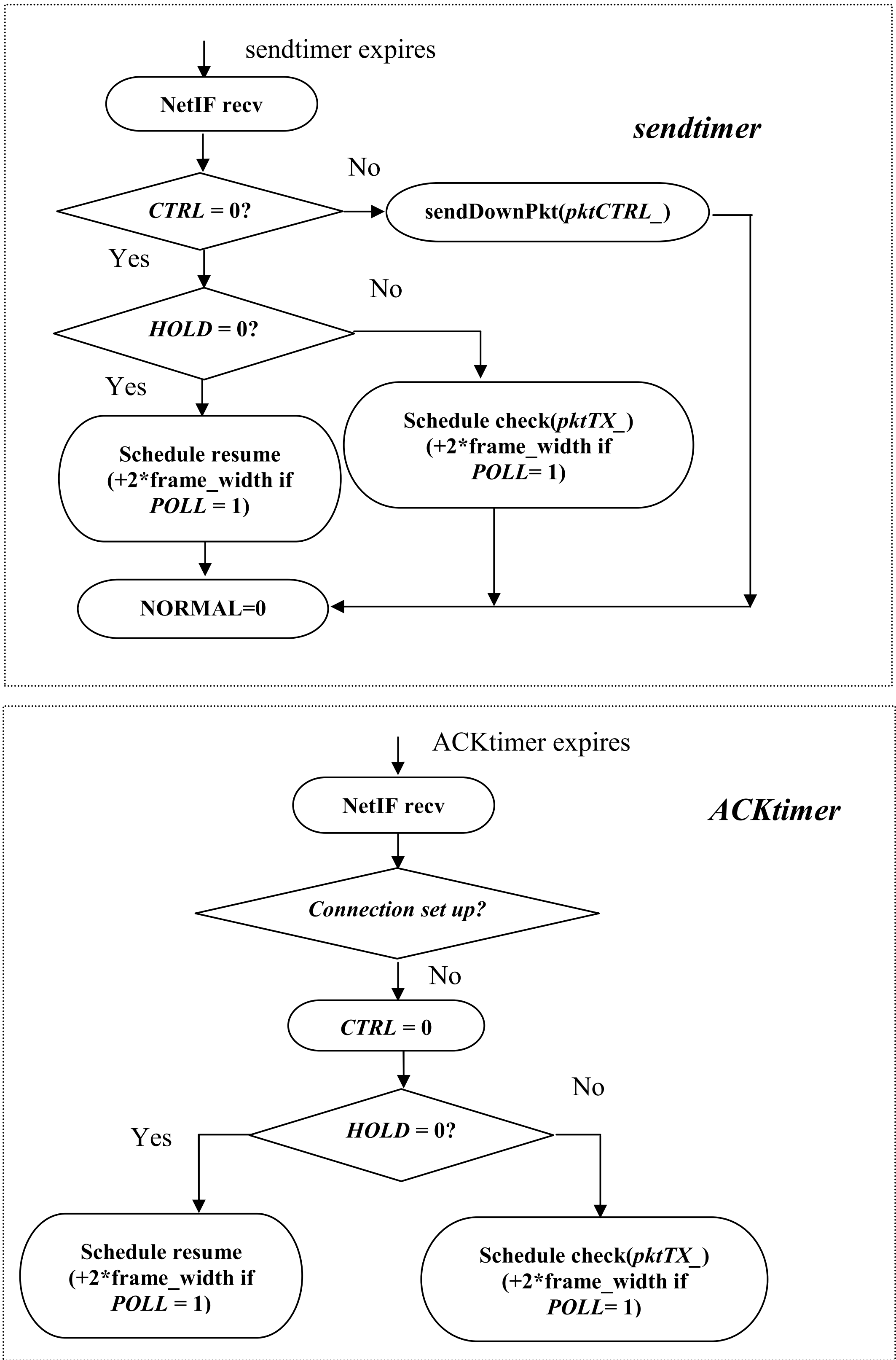


Figure 4.15: Block diagram of *sendtimer* and *ACKtimer* of MAC

4.3.3. NetIF and Channel

Each wireless node is connected to each Channel via a NetIF. The NetIF receives a packet either from the MAC layer or the Channel (*recv*). When it sends a packet into the Channel (*send down*), transmission information is stamped on the packet and the Channel distributes a copy of the packet to other NetIFs on the channel.

Firstly, an energy model is used to determine if the packet can be detected; otherwise, the packet is dropped. If so, a radio propagation model can be used to determine if the packet can be received; otherwise, the packet is dropped. For packets that are detected and received, a modulation model can be used to determine the BLER according to the modulation scheme, before the packet is sent up (*send up*) to the MAC layer.

However, current implementation of NetIF and Channel in *ns* supports the shared media channel using Lucent's WaveLAN DSSS radio interface, which does not make use of the radio propagation as well as modulation model. The functions of NetIF are illustrated in Figure 4.16.

In our simulation, the NetIF performs link adaptation at the receiving node as described in Section 2.5.3.3. The channel quality is quantified by the *C/I* and varies between 6-30dB in a typical urban environment [10]. This value is fixed for each simulation run as the MS is assumed to be stationary and hence, the channel quality will not vary substantially during the simulation period.

4.3.3.1. Computation of BLER and updating of error flag

For each coding scheme, graphs of BLER versus *C/I* are available from actual measurement [15] and an example is shown in Figure 2.8 for ideal FH. A 2-dimensional array is defined for BLER indexed by *C/I* and coding scheme, *n*. The BLER value will determine the status of the error flag of the block.

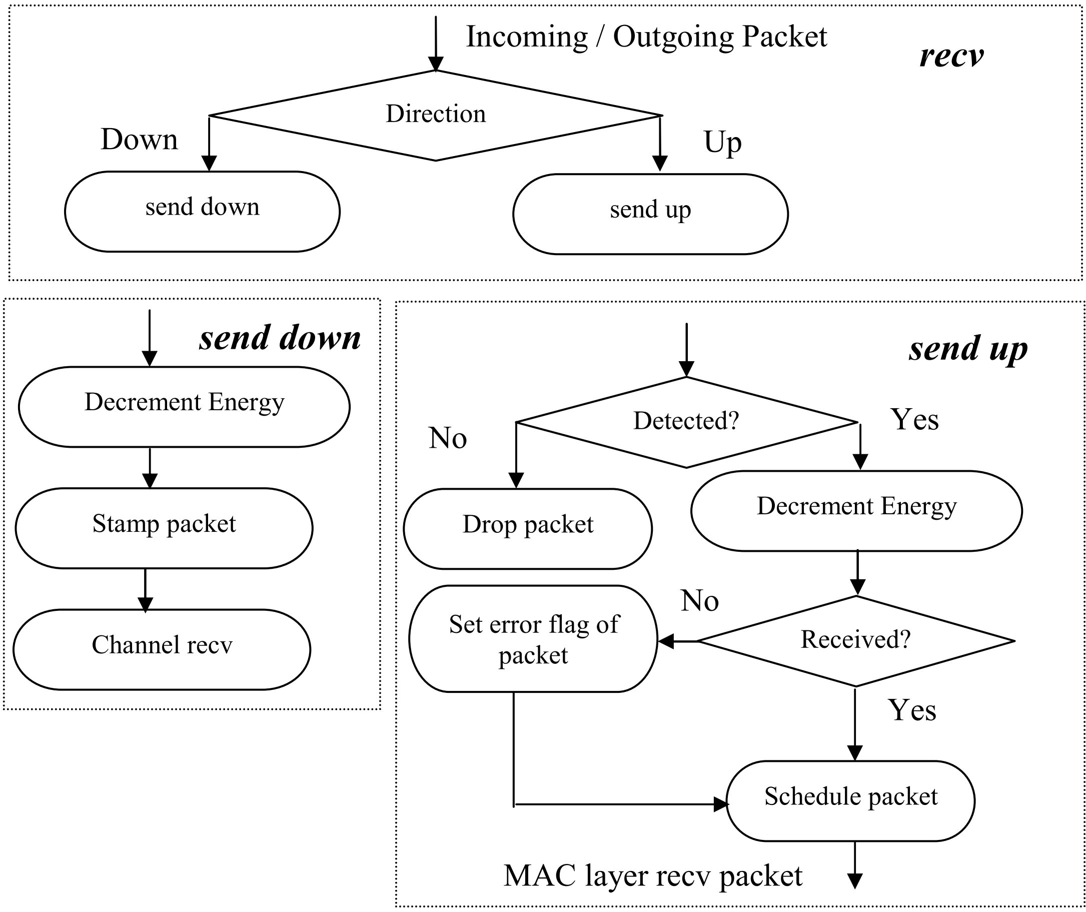


Figure 4.16: Block diagram of current NetIF implementation in *ns*

4.3.3.2. *Estimation of average C/I*

According to [12], an estimate of the average *C/I* over a polling interval can be obtained with an error variance of 1 dB from the actual *C/I*.

4.3.3.3. *Selection of optimal coding scheme for next polling interval*

If the poll flag of the current block is set, the average *C/I* value is computed over the last polling interval as in subsection 4.3.3.2. This is used to compute the throughput, S_n, for various coding schemes, n, and that which maximizes the throughput (eqn. (2.1) of Chapter 2) is selected and coded into the header of the block. This information will be sent via the ACK message to the sender to update the coding scheme.

These modifications are made in the NetIF *send up* function, and illustrated in Figure 4.17.

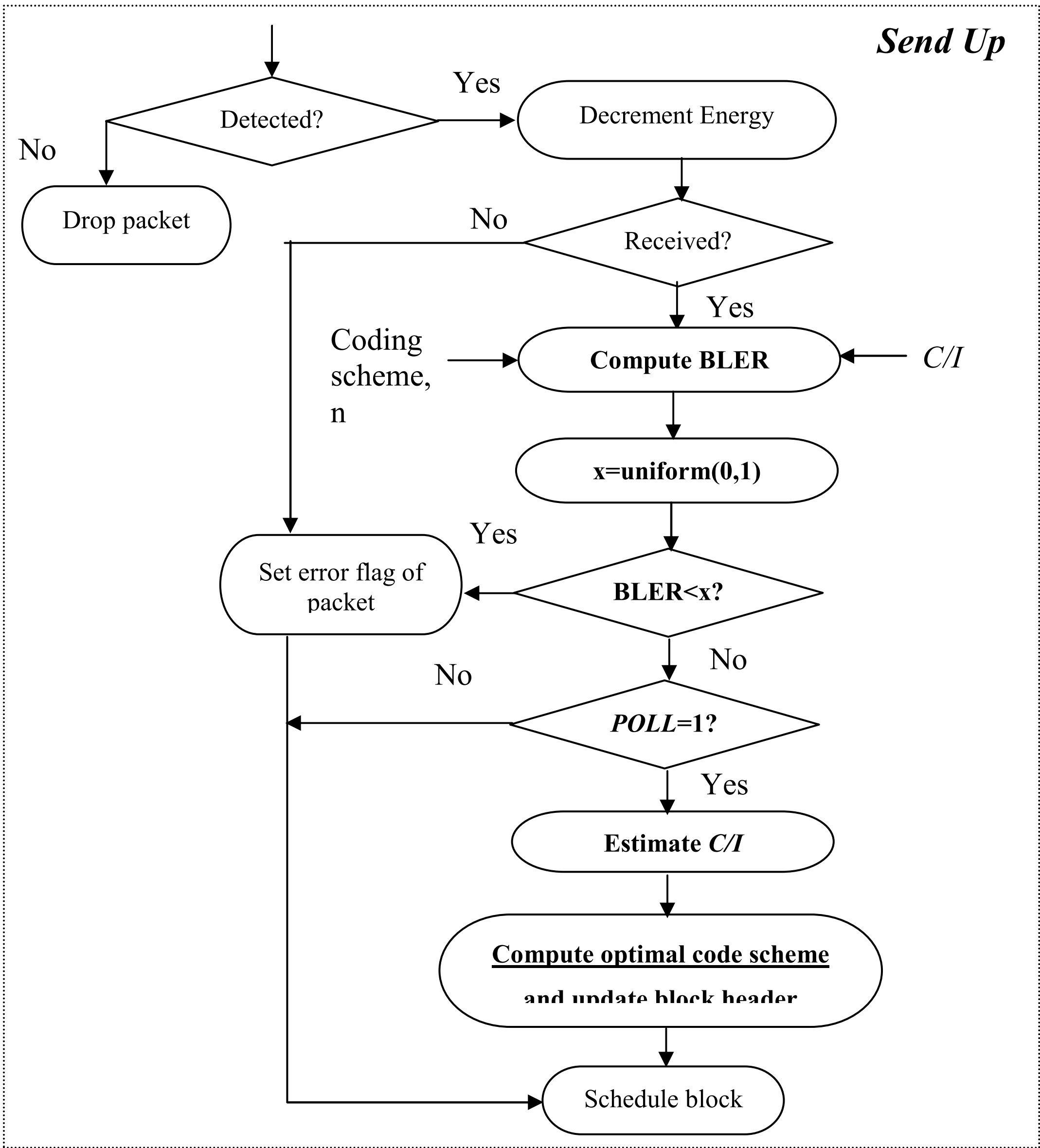


Figure 4.17: Block diagram of NetIF *send up* function

Chapter 5

SIMULATION RESULTS

This chapter presents the results of TCP performance over EGPRS via simulation. The simulation network configuration and important simulation parameters are first specified. Results without link adaptation are first presented to illustrate the enhanced data rates achievable with EGPRS for good channel conditions. Results using various link adaptation techniques are shown and compared to illustrate optimal throughput capabilities at all channel conditions. A hybrid link adaptation method is proposed and some results are shown, which indicate better performance than the individual link adaptation approaches.

5.1. Simulation Network Configuration

End-to-end TCP performance over EGPRS is evaluated for bulk data transfer from the Remote Host to the MS based on the scenario in Figure 5.1. Important simulation parameters and the assigned values are shown in Table 5.1.

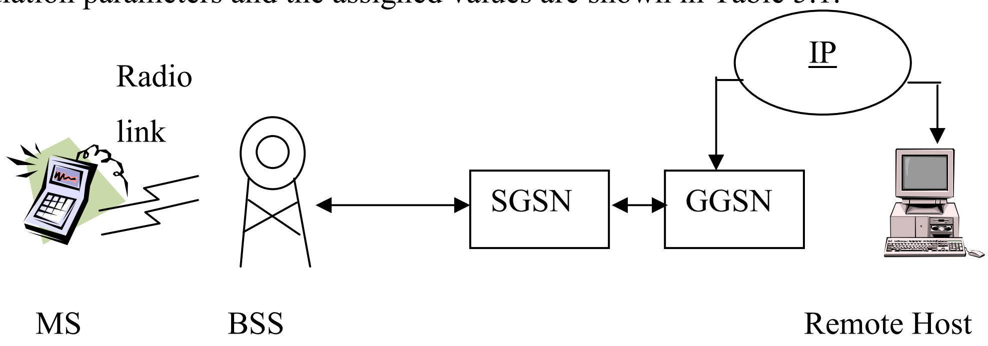


Figure 5.1: EGPRS top-level simulation environment

Parameter	Value
Simulation time	2000 seconds
TCP version	Tahoe
TCP Packet Size	512 bytes
TCP Window Size	16 packets
RLC Polling Interval	10 blocks
Number of TS	1
Number of MS	1
C/I range	6-30 dB
Coding schemes	ECS1 - 8
Frequency Hopping	Ideal
Modulation scheme	O16QAM
Mean Internet Delay	100ms
Mean Internet Loss	0.01

Table 5.1: EGPRS simulation parameters

The MS is assumed to remain stationary during the simulation, and for each simulation run, a fixed *C/I* value is used.

5.2. Performance parameters

Trace agents are attached to the TCP source and sink so that the arrival and departure time of each packet is recorded in the packet header. *Throughput, delay* and *delay variations* can then be computed from the trace output, which are then used to evaluate TCP performance over EGPRS.

(a) Throughput

Throughput can be approximated as the number of packets successfully received by the MS over the simulation duration (i.e., 2000 seconds).

(b) Delay

Delay indicates the mean end-to-end delay experienced by each packet received at the MS. Each packet is stamped with the time of arrival at the Remote Host, *t_start*, when the packet is generated. Each packet is acknowledged as it is received at the destination and the packet is stamped with the time of arrival, *t_end*. The end-to-end duration of each packet is recorded as *t_end – t_start*, and the average for all received packets is calculated.

(c) Delay variation

Delay variation is measured by the standard deviation of the end-to-end delay of packets received at the MS, which is computed as in (b).

Throughput is the data rate perceived by the user and is important in all kinds of services including FTP, web browsing, or email retrieval. Delay is important for real-time services such as voice over IP or video telecommunications over IP. Interactive applications have stringent requirements on both delay and delay variations.

In our simulation, TCP performance is measured for each EGPRS coding scheme under various channel conditions. Firstly, we consider the TCP performance without link adaptation in Section 5.3 to demonstrate EGPRS data rate capabilities over GPRS. Section 5.4 and 5.5 shows the TCP performance using link adaptation algorithms, Estimated *C/I* and BLER, while Section 5.6 presents the results for the hybrid link adaptation approach.

5.3. TCP Performance without Link Adaptation

5.3.1. Theoretical throughput performance for EGPRS

In [15], link simulations were performed to evaluate the performance of EGPRS in terms of BLER under typical urban channel conditions. From the experiments, BLER results for each coding scheme (ECS-1-to ECS-8) are obtained (Figure 2.5, Chapter 2).

With these results, together with given LLC data rates, R_n, for each coding scheme, n, (Table 2.2 of Chapter 2), the theoretical throughput, S_n, for each coding scheme n as a function of link quality, C/I can be calculated according to [18] as:

$$S_n = R_n \times (1\text{-}BLER_n(C/I)) \qquad (5.1)$$

S_n obtained using eqn. (5.1) is plotted against C/I as shown in Figure 5.2.

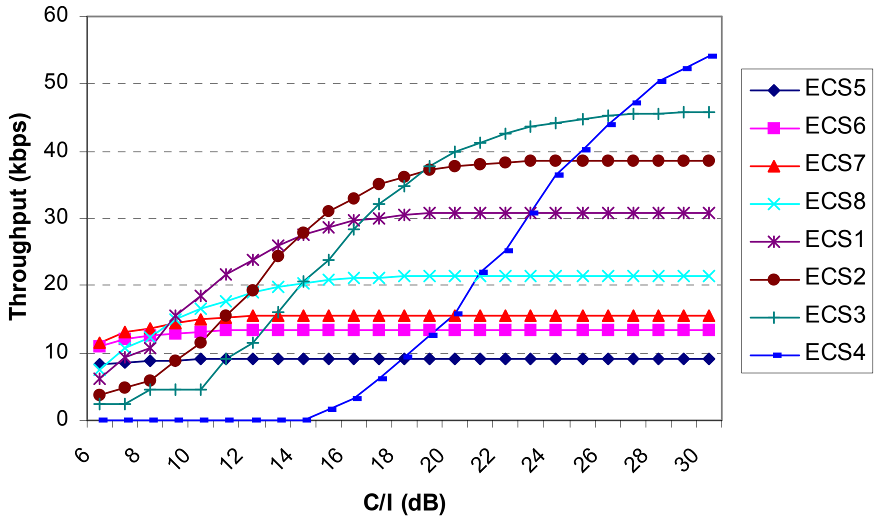


Figure 5.2: Theoretical throughput performance for EGPRS

It is shown that with ECS-5 to ECS-8, theoretical throughput levels up to 22.8 kbps can be achieved under good channel conditions. With the enhanced coding schemes, i.e., ECS-1 to ECS-4, theoretical throughput levels up to 55 kbps can be achieved under good channel conditions.

5.3.2. Simulated throughput performance for EGPRS

The simulated throughput obtained for each *C/I* is plotted in Figure 5.3.

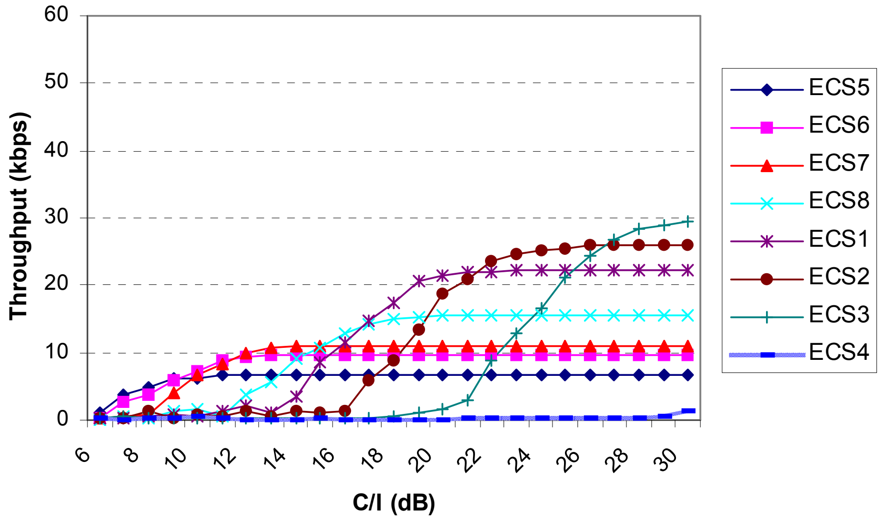


Figure 5.3: Simulated throughput performance for EGPRS

It is observed that the throughput levels are significantly lower than the corresponding theoretical values depicted in Figure 5.2.

This can be explained as follows. The throughput computed using eqn. (5.1) assumes that most of the RLC blocks are successfully transmitted on the first attempt. This may be true for GPRS coding schemes (i.e., ECS-5 to ECS-8). However, for high-level coding schemes (i.e., ECS-1 to ECS-4), which are highly sensitive to noise, significant re-transmissions occur that can contribute to additional latency. This implies that on average, each packet takes a longer time to reach the destination, which eventually results in throughput degradation. A new formulation of throughput that considers this latency is described in Section 5.3.3.

5.3.3. Derivation of throughput formula

Consider transmission of P packets (each of fixed size M bits) over an EGPRS network with coding scheme n over a channel, with link quality of C/I. Theoretically, out of these P packets, $(1\text{-}BLER_n(C/I)) \times P$ packets will be transmitted successfully with bandwidth R_n bps. This is equivalent to sending P packets with a reduced bandwidth of $(1\text{-}BLER_n(C/I)) \times R_n$. Therefore, the time needed to send these P packets is computed as follows:

$$T\_normal = \frac{P \times M}{(1 - BLER_n(\frac{C}{I})) \times R_n} \tag{5.2}$$

Based on eqn. (5.2), the throughput can be computed as follows:

$$S_n = (P \times M) / T\_normal = (1 - BLER_n(\frac{C}{I})) \times R_n \tag{5.3}$$

This is in agreement with eqn. (5.1), which assumes that re-transmission latency is insignificant compared to the time taken to transmit the packets on the first attempt.

To take into consideration re-transmission latency, consider a re-transmission of lost packets using the best coding scheme (ECS-5) and assume that these re-transmissions will be successful. Therefore, additional time required to send these $BLER_n(C/I) \times P$ packets is computed as follows:

$$T\_retrans = \frac{BLER_n(\frac{C}{I}) \times P \times M}{(1 - BLER_{ECS-5}(\frac{C}{I})) \times R_{ECS-5}} \tag{5.4}$$

Hence, the total time to send all P packets successfully is computed as follows:

$$\begin{aligned} T\_total &= T\_normal + T\_retrans \\ &= \frac{P \times M}{(1 - BLER_n(\frac{C}{I})) \times R_n} + \frac{BLER_n(\frac{C}{I}) \times P \times M}{(1 - BLER_{ECS-5}(\frac{C}{I})) \times R_{ECS-5}} \end{aligned} \tag{5.5}$$

The throughput can then be obtained as follows:

$$\begin{aligned} S_n &= (P \times M) / T\_total \\ &= \frac{R_{ECS-5} \times R_n \times (1 - BLER_{ECS-5}(C/I)) \times (1 - BLER_n(C/I))}{(1 - BLER_{ECS-5}(C/I)) \times R_{ECS-5} + R_n \times BLER_n(C/I) \times (1 - BLER_n(C/I))} \end{aligned} \tag{5.6}$$

In addition, at the end of each polling interval, the transmitter stops transmission for two block-intervals while waiting for the ACK message from the receiver. This is manifested in the throughput computation by re-computing each bandwidth, R_n, as follows:

$$R_{n\_new} = \frac{N\_blocks\_per\_interval}{N\_blocks\_per\_interval + 2} \times R_{n\_old} \tag{5.7}$$

where *N_blocks_per_interval* is the number of RLC blocks per polling interval (default simulation value = 10)

5.3.4. TCP Performance using Fixed Polling Interval

5.3.4.1. Throughput results

Figure 5.4 shows the theoretical throughput is re-computed using eqn (5.5) and (5.6). Figure 5.3 is reproduced and is shown in Figure 5.5 for better comparison.

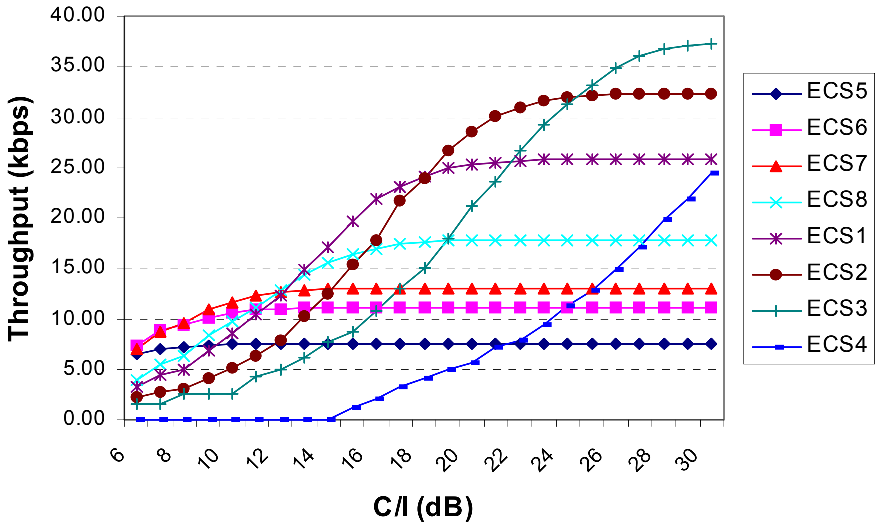


Figure 5.4: Theoretical throughput performance for EGPRS

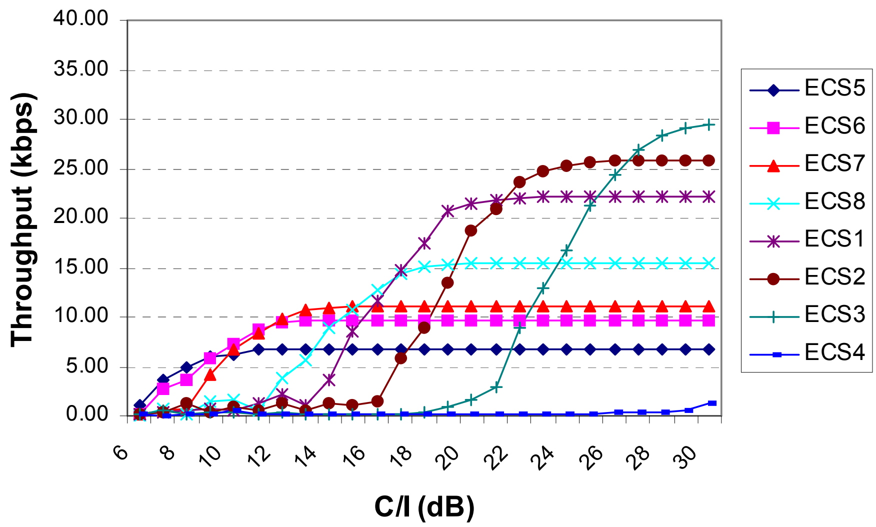


Figure 5.5: Simulated throughput performance for EGPRS

It is observed that the maximum throughput achievable with each coding scheme is within 8 kbps from the theoretical throughput as computed in Figure 5.4. The discrepancy could be due to the bandwidth needed to send non-data packets, e.g., ACK packets between the Remote Host and the MS.

Under very good channel conditions, enhanced coding schemes of EGPRS (ECS-1, ECS-2 and ECS-3) saturates at significantly higher data rates (between 22 to 30 kbps) than GPRS schemes (ECS-5 to ECS-8), which saturates between 6 to 15 kbps. ECS-4, the enhanced scheme with the highest data rate, fails to deliver useful throughput under all channel conditions.

5.3.4.2.Delay results

The average end-to-end delay for each packet is obtained and shown in Figure 5.6.

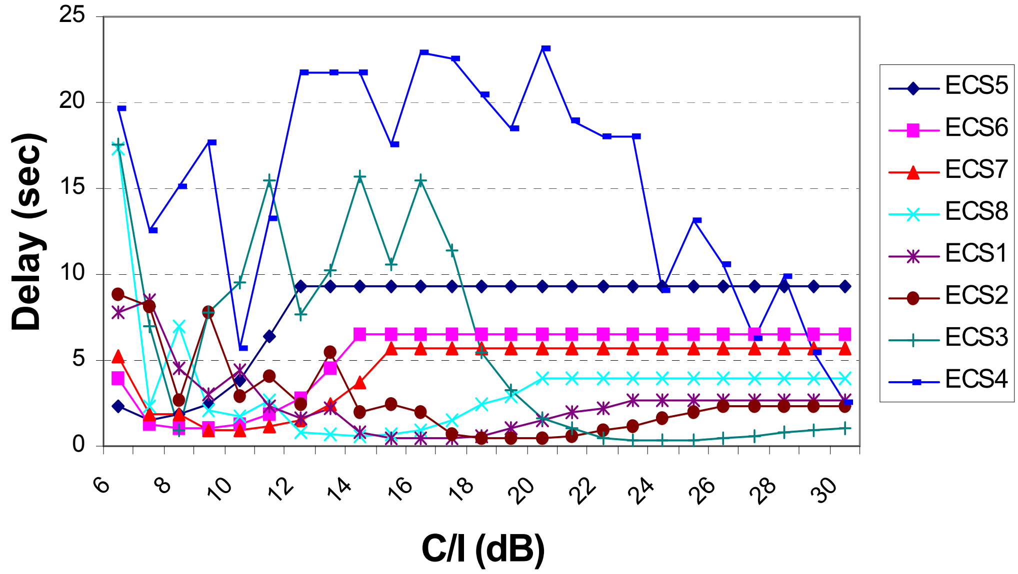


Figure 5.6: Delay performance for EGPRS

Under good channel conditions (e.g., C/I larger than 20 dB), it is observed that the average delay per packet is reduced as we switch from GPRS schemes to enhanced coding schemes of EGPRS, except ECS-4.

It is also observed that the data protection level in ECS-4 is inadequate to produce useful throughput and delay characteristics under most channel conditions. Hence, this scheme is not considered when evaluating TCP performance with link adaptation.

5.3.4.3.Delay variation results

Figure 5.7 depicts the standard deviation of end-to-end delay of all packets.

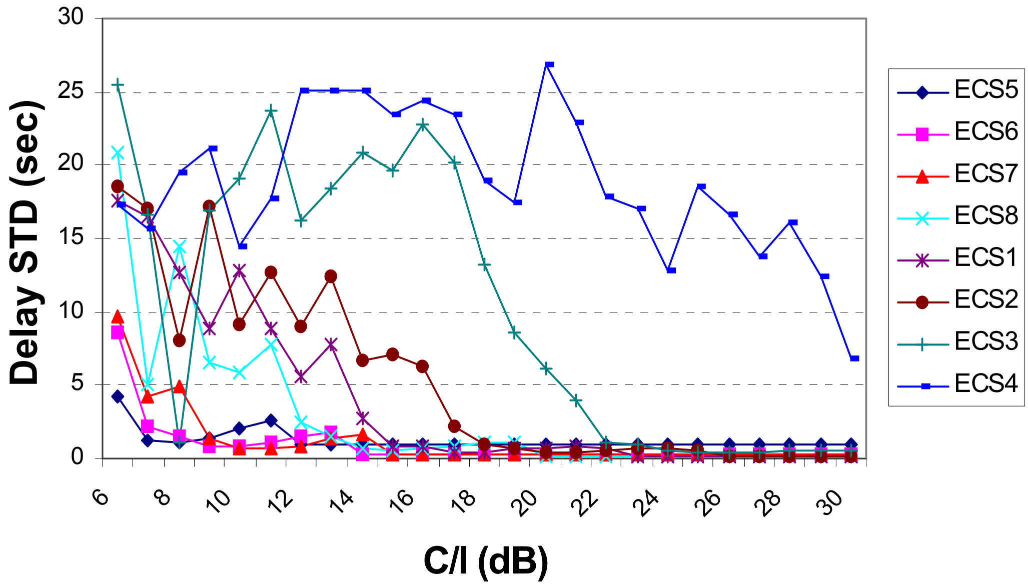


Figure 5.7: Delay variation performance for EGPRS

Under very good channel conditions (e.g., C/I larger than 23 dB), it is observed that the delay standard deviation (STD) is within 0.5 sec for all coding schemes except ECS-4. Under other channel conditions, the delay STD is too large for applications that are sensitive to delay variations, e.g. voice or interactive services.

It is observed from Figure 5.6 and 5.7 that delay and delay variation are bounded for coding schemes ECS-5 to ECS-7, but are very high for the other coding schemes, especially under bad channel conditions. For the remaining schemes, each IP packet is fragmented into < 10 RLC blocks (per polling interval), and hence, unnecessary delay may be incurred to receive the complete IP packet while waiting for ACK messages.

A suggestion is to use an adaptive polling interval to supplement the fixed scheme. In this scheme, when the MAC layer at the sending node receives a block from the LL that corresponds to the last block of the packet, the poll flag in the block header is also set (in addition to the original condition). This will immediately trigger an ACK message from the receiving node, thus reducing unnecessary latency.

5.3.5. TCP Performance using Adaptive Polling Interval

5.3.5.1. Throughput results

When an adaptive polling interval is used, the parameter, *N_block_per_interval*, in eqn. (5.7) is replaced by *N_block_per_packet(n)* (i.e., No. of blocks per IP packet for coding scheme *n*) which varies with each coding scheme *n*. This is to take into account transmission overhead due to ACK messages, which do not contribute to useful throughput. The theoretical throughput is plotted in Figure 5.8, and simulated throughput is illustrated in Figure 5.9.

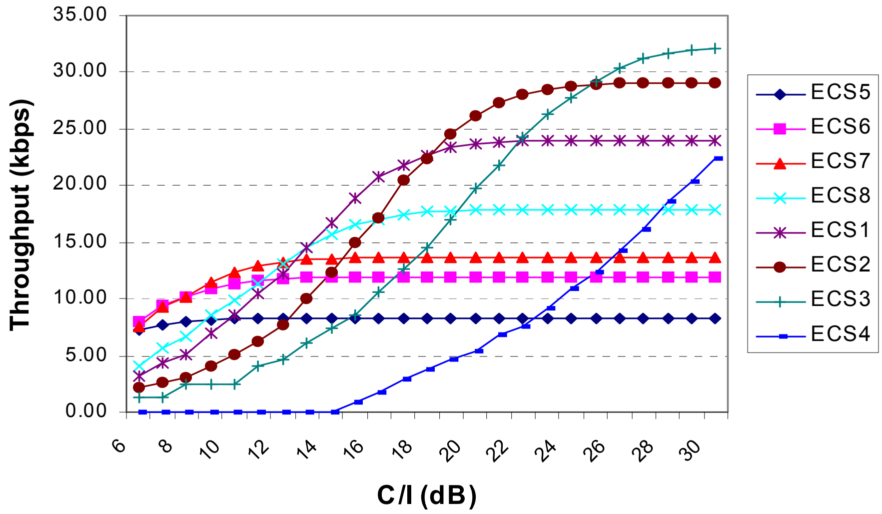


Figure 5.8: Theoretical throughput performance for EGPRS for adaptive polling

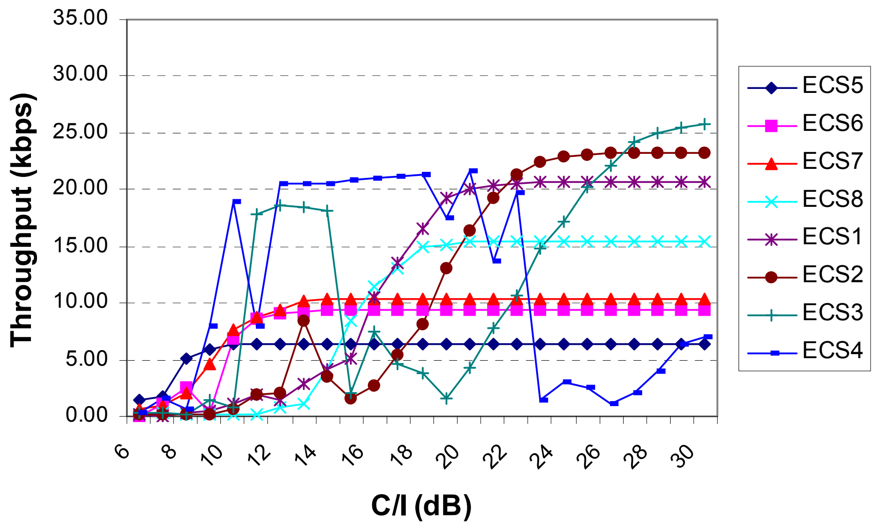


Figure 5.9: Simulated throughput performance for EGPRS for adaptive polling

It is observed that the maximum throughput achievable with each coding scheme is within 8 kbps from the theoretical throughput as computed in Figure 5.8. The discrepancy could be due to the delay incurred for sending non-data packets, e.g., ACK packets between the Remote Host and the MS.

With adaptive polling, substantial throughput levels can be achieved with enhanced coding schemes (ECS-2 to ECS-4) under poor channel conditions.

5.3.5.2. Delay results

The average end-to-end delay for each packet is obtained and shown in Figure 5.10.

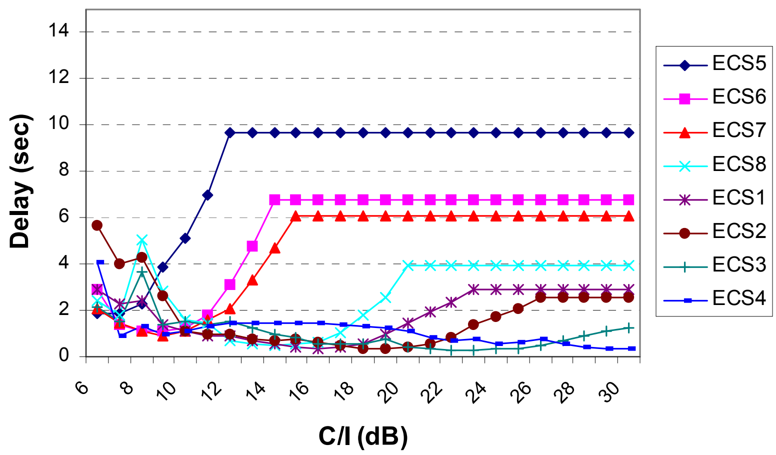


Figure 5.10: Delay performance for EGPRS for adaptive polling

Under good channel conditions, it is observed that the average delay per packet is reduced as we transit from GPRS coding schemes to EGPRS coding schemes.

5.3.5.3. Delay variation results

Figure 5.11 illustrates the STD of end-to-end delay of all received packets.

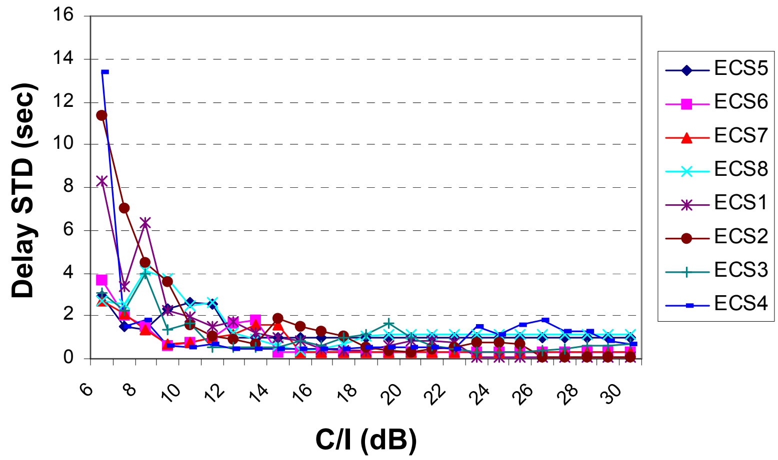


Figure 5.11: Delay variation performance for EGPRS for adaptive polling

It is observed that packet delay variation for all coding schemes is reduced as channel conditions improve.

5.3.6. Comparison of TCP performance using fixed and adaptive polling

5.3.6.1. Throughput comparison

The throughput obtained for fixed and adaptive polling is shown in Figure 5.12 and 5.13, respectively. ECS-4 is not considered as it is extremely sensitive to noise, and does not yield predictable throughput.

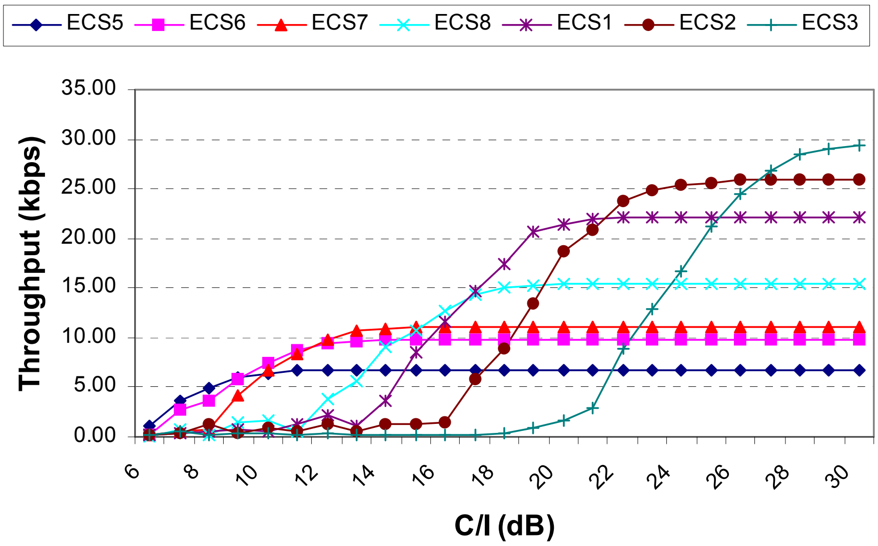


Figure 5.12: Throughput performance for fixed polling interval

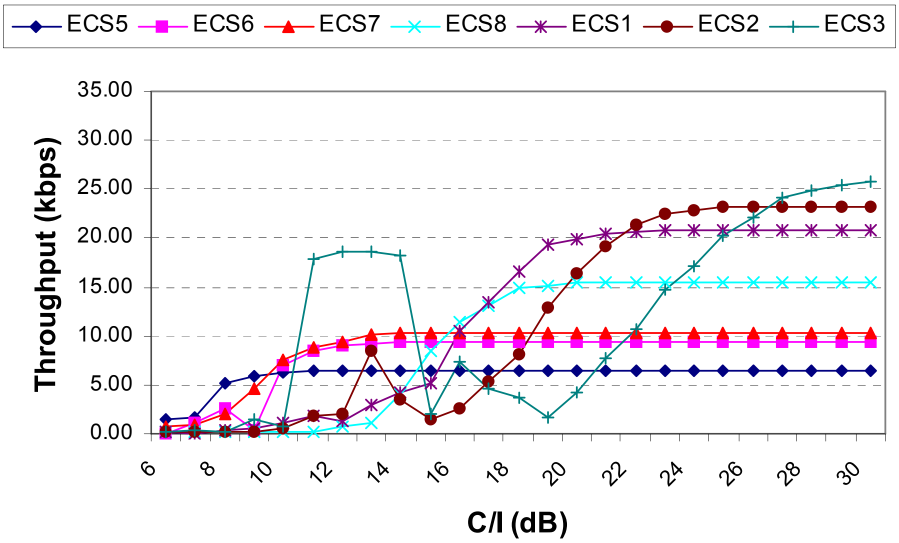


Figure 5.13: Throughput performance for adaptive polling interval

The performance for GPRS schemes (i.e., ECS-5 to ECS-8) is similar with both fixed and adaptive polling intervals. For these schemes, the number of blocks per packet is greater than 10, and hence, the two approaches are similar i.e. the poll flag is set at every fixed polling interval for both cases.

Under poor channel conditions, re-transmissions dominate normal transmissions for enhanced coding schemes (ECS-1 to ECS-3). Since transmission failures are detected earlier with adaptive polling, re-transmissions take place sooner and hence, throughput levels are much higher than fixed polling ("Hump" observed for ECS-3 for C/I between 11 to 15 dB). Under very good channel conditions, most transmissions are successful on the first attempt, and hence adaptive polling consumes unnecessary bandwidth, and results in lower throughput.

5.3.6.2.Delay comparison

The average end-to-end delay of each packet using fixed and adaptive polling is shown in Figure 5.14 and 5.15 respectively.

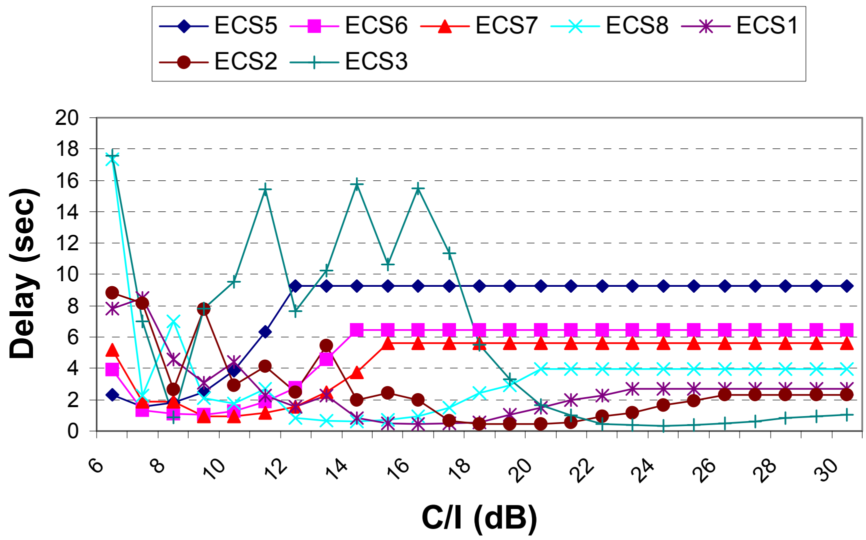


Figure 5.14: Delay performance for fixed polling interval

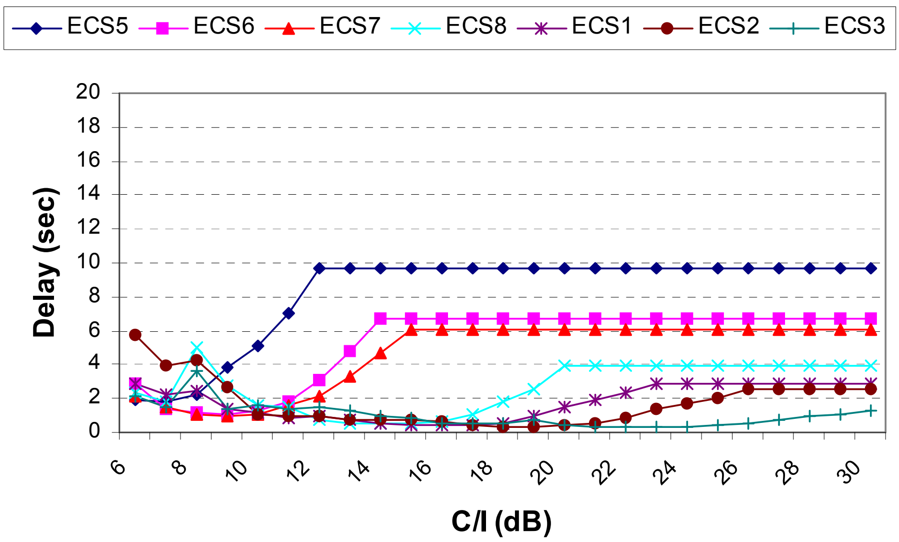


Figure 5.15: Delay performance for adaptive polling interval

The delay performance for ECS-5 to ECS-8 is similar for both fixed and adaptive poll intervals. For ECS-1 to ECS-3, the delay performance is superior with adaptive polling under poor channel conditions.

The delay variation for fixed and adaptive polling is shown in Figure 5.16 and 5.17 respectively.

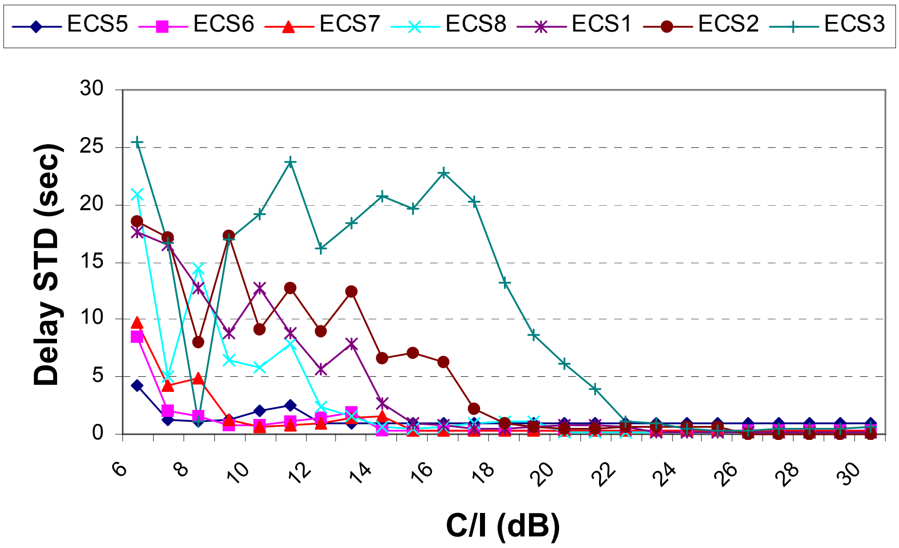


Figure 5.16: Delay variation performance for fixed polling interval

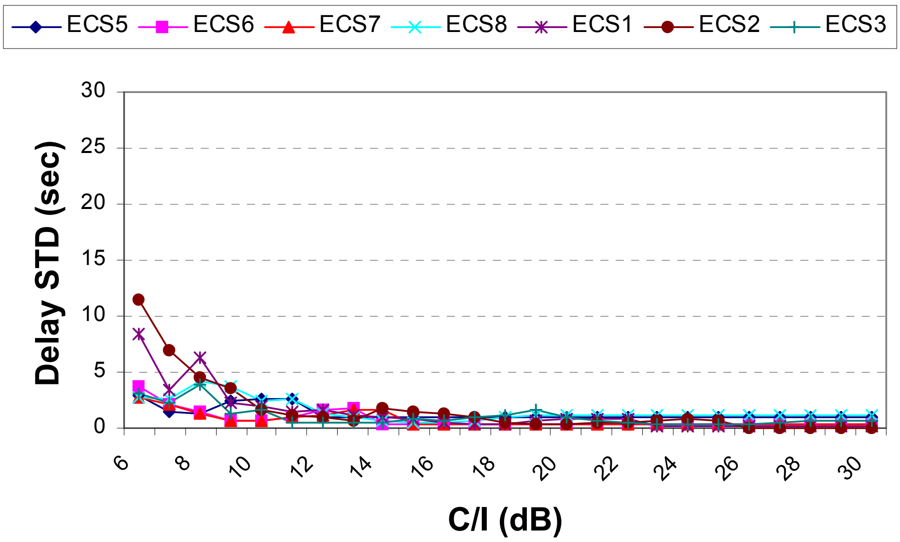


Figure 5.17: Delay variation performance for adaptive polling interval

The delay variation performance for ECS-5 to ECS-8 is similar for both fixed and adaptive polling under good channel conditions. The delay variation performance is superior with adaptive polling under poor channel conditions.

Based on the observations, it is apparent and logical that adaptive polling produces superior TCP performance when enhanced coding schemes (i.e., ECS-1 to ECS-3) are used under poor channel conditions over fixed polling. On the other hand, fixed polling achieves better TCP performance for high-level modulation schemes under good channel conditions. Hence, a fusion of fixed and adaptive polling should be employed to yield optimal TCP performance with high-level modulation under all channel conditions.

5.4. TCP Performance with Link Adaptation (Estimated C/I Approach)

5.4.1. Throughput performance

Using fixed polling, the throughput performance using the Estimated C/I algorithm for link adaptation is plotted in Figure 5.18, denoted by *Actual LA*. Based on eqn. (5.6), the coding scheme that would be selected by link adaptation for each C/I can be determined theoretically. The throughput levels obtained at each C/I using this set of coding schemes is plotted alongside the simulated throughput, and is denoted as *Theoretical LA*. In addition, the figure shows the ideal link adaptation throughput (denoted as *Ideal LA*). The ideal link adaptation throughput is determined by selecting the maximum simulated throughput at each C/I (Figure 5.4) over all the coding schemes.

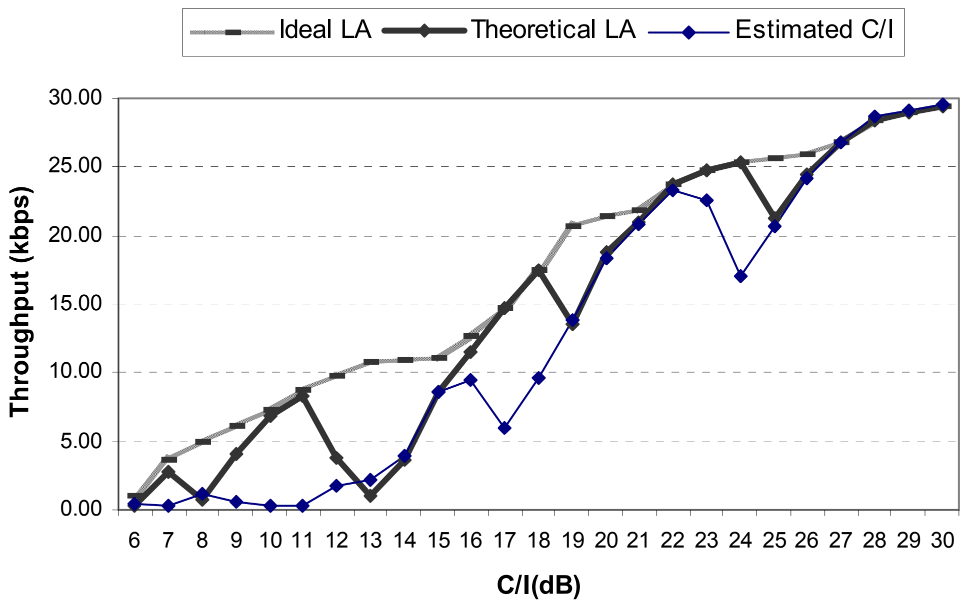


Figure 5.18 Throughput performance with link adaptation for fixed polling interval

Comparing the Theoretical LA throughput with the Ideal LA throughput, the throughput levels improve with increase in C/I in both cases. However, some ‘dips’ are observed in the Theoretical LA throughput, where it fails to keep up with the ideal throughput levels. This could be because eqn. (5.6) does not model the throughput behavior accurately, and hence, it does not select the optimal coding scheme at the C/I where these ‘dips’ appear.

Comparing the Actual LA throughput with the Theoretical LA throughput, a similar trend is observed in both cases, except for a shift along the C/I axis. This could be due to the 1 dB error variance in the C/I estimation.

The corresponding graph for adaptive polling is shown in Figure 5.19. For this case, link adaptation only selects amongst ECS-1, ECS-2 and ECS-5 to ECS-8.

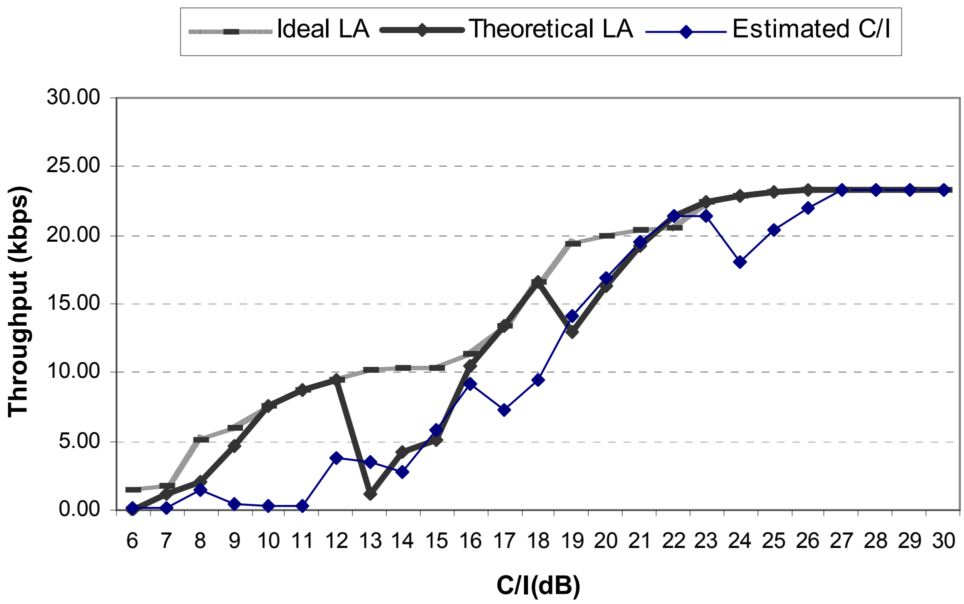


Figure 5.19 Throughput performance with link adaptation for adaptive polling interval

Similar observations are found in this case as for the case of using fixed polling intervals.

5.4.2. Delay performance

Using fixed polling, the average packet delay is plotted in Figure 5.20. In addition, the figure shows the minimum simulated packet delay at each C/I (Figure 5.6) over all coding schemes.

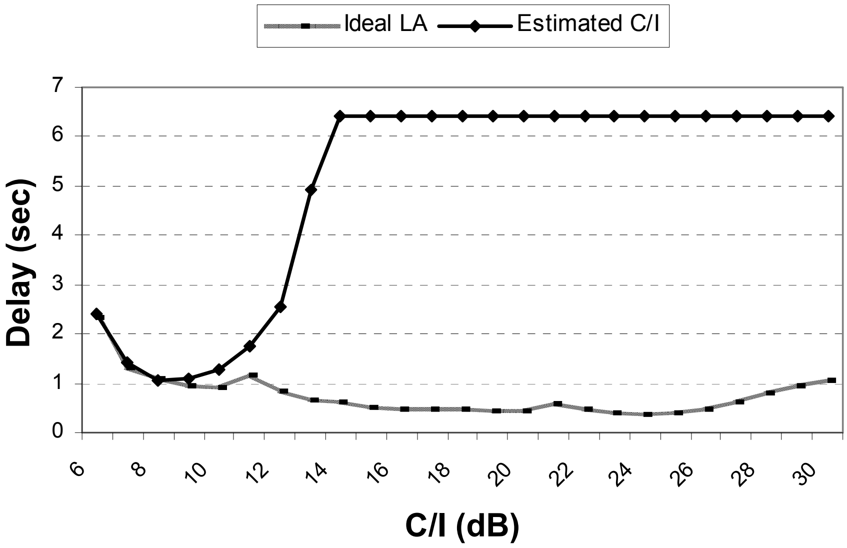


Figure 5.20 Delay performance with link adaptation for fixed polling interval

It can be seen that the Estimated C/I algorithm is unable to select the best coding scheme in terms of delay under poor channel conditions. As channel conditions improve, the average packet delay is reduced, and minimum delay values are achieved under very good channel conditions.

The corresponding graph for adaptive polling is shown in Figure 5.21. It is capable of achieving minimum average packet delay under all channel conditions.

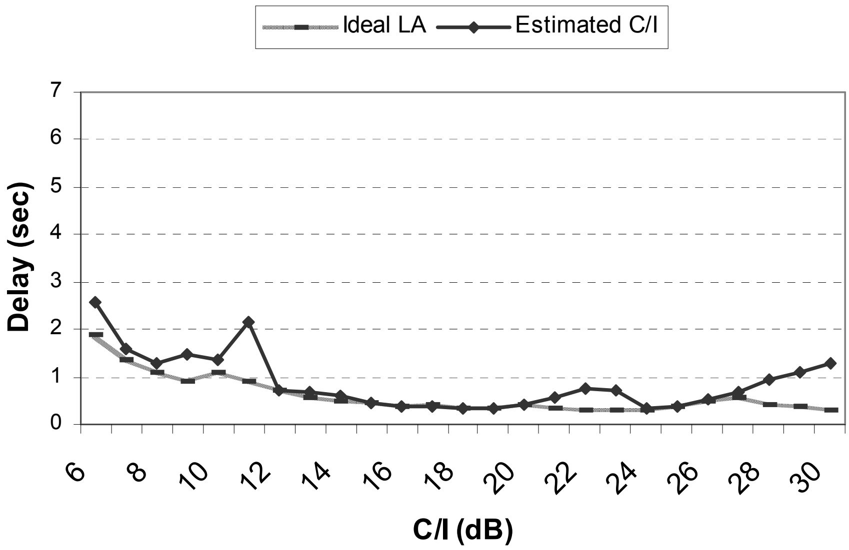


Figure 5.21 Delay performance with link adaptation for adaptive polling interval

5.4.3. Delay variation performance

Using fixed polling interval, the packet delay variation is plotted in Figure 5.22. In addition, the figure shows the minimum simulated delay variation at each C/I (Figure 5.7) over all coding schemes.

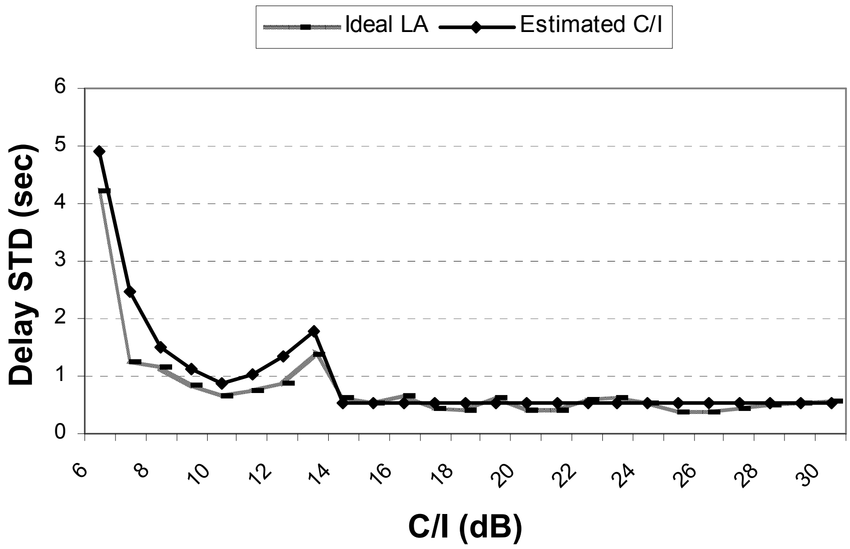


Figure 5.22 Delay variation performance with link adaptation for fixed polling interval

It can be seen that the Estimated C/I algorithm is unable to select the best coding scheme in terms of delay variation under poor channel conditions. As channel conditions improve, the delay variation is reduced, and minimum delay variation values are achieved under very good channel conditions.

The corresponding graph for adaptive polling is shown in Figure 5.23. It is capable of achieving minimum packet delay variation under all channel conditions.

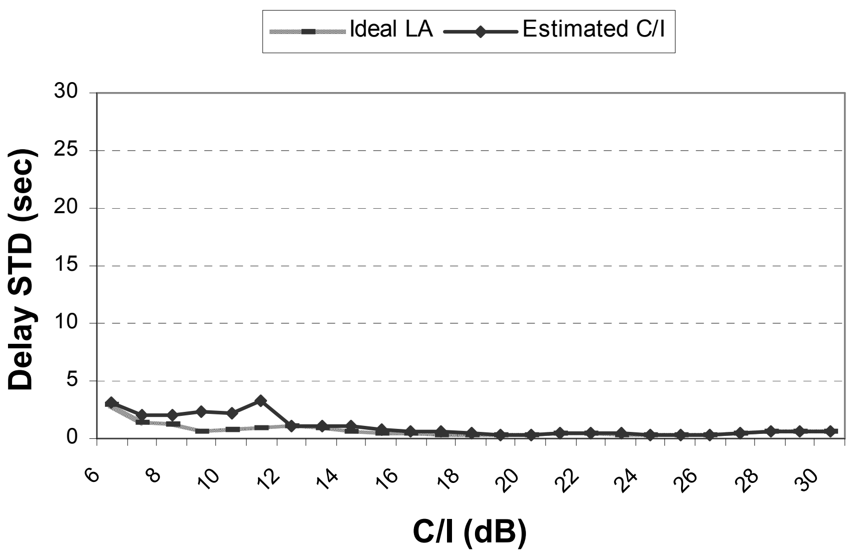


Figure 5.23 Delay variation performance with link adaptation for adaptive polling interval

5.5. TCP Performance with Link Adaptation (BLER Approach)

In the BLER approach, the average error rate is computed based on the number of erroneous blocks amongst a window of transmission corresponding to the polling interval. Since the lower threshold for coding scheme update is 10%, a minimum polling interval of 10 blocks is required. Hence, the adaptive polling scheme is not applicable with this approach and hence only fixed polling will be studied.

5.5.1. Throughput performance

Using fixed polling, the throughput performance using the BLER algorithm for link adaptation is plotted in Figure 5.24. In addition, the figure shows the maximum simulated throughput in the case of ideal link adaptation. The ideal link adaptation throughput is determined by selecting the maximum throughput at each C/I (Figure 5.4) over all the coding schemes.

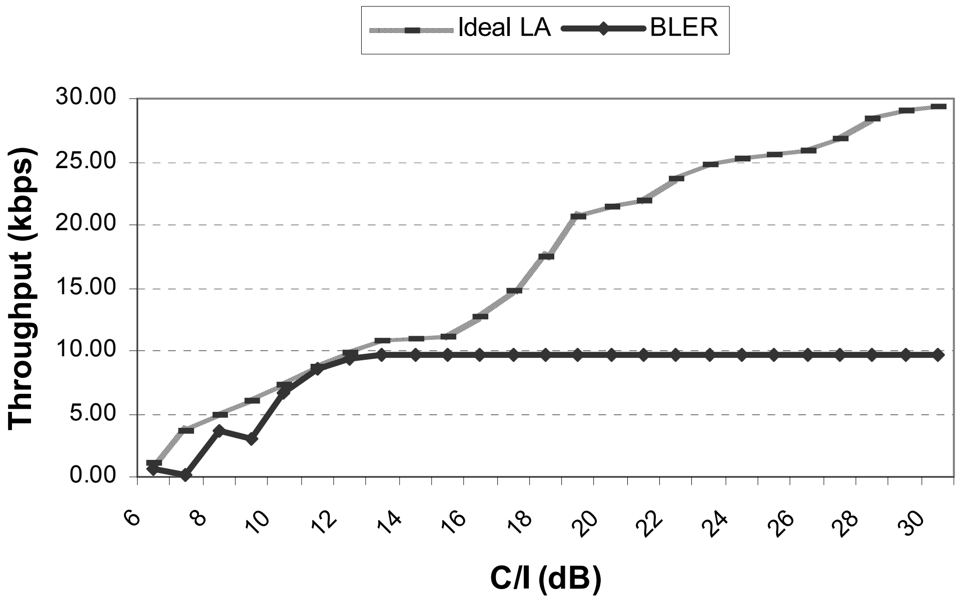


Figure 5.24 BLER link adaptation throughput performance for fixed polling

It is observed that this approach approximates ideal link adaptation for $C/I < 14$ dB. As channel conditions improve, the throughput level saturates at 9.72 kbps, which corresponds to the level achieved using ECS-6 under good channel conditions.

5.5.2. Delay performance

The average end-to-end packet delay obtained is plotted in Figure 5.25. In addition, the figure shows the minimum simulated packet delay at each C/I (Figure 5.6) over all coding schemes. The BLER approach to link adaptation is delay-optimal for *C/I* < 9 dB. Delay performance is degraded as channel conditions improve, stabilizing to a level corresponding to the saturation level of ECS-6.

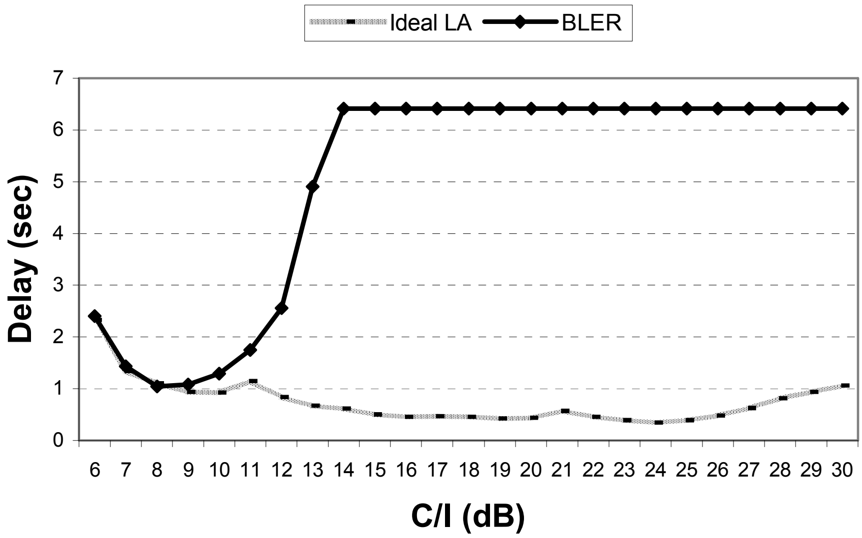


Figure 5.25 Delay performance for BLER link adaptation for fixed polling interval

5.5.3. Delay variation performance

The packet delay STD is plotted in Figure 5.26. In addition, the figure shows the minimum simulated packet delay variation at each C/I (Figure 5.7) over all coding schemes. Optimal performance is achieved under all channel conditions.

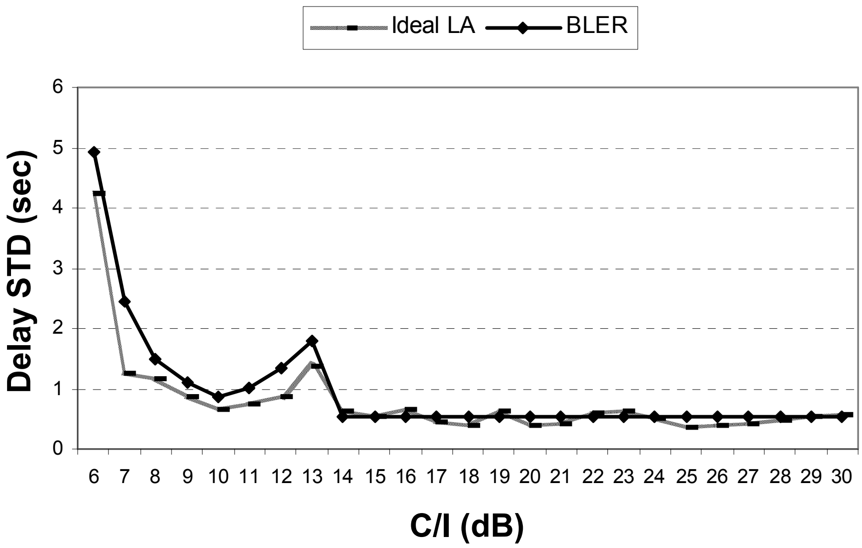


Figure 5.26 Delay variation performance for BLER link adaptation for fixed polling interval

5.6. Comparison of TCP Performance with different Link Adaptation Algorithms

Figure 5.27 to Figure 5.29 shows the throughput, delay and delay variation performance of the Estimated C/I and BLER algorithms compared to ideal link adaptation performance.

In terms of throughput performance, the BLER approach performs better than the Estimated C/I approach for C/I under 18 dB, and vice versa.

In terms of delay performance, the BLER approach performs better than Estimated C/I approach for C/I under 12 dB, and vice versa.

In terms of delay variation performance, the BLER approach achieves better performance for C/I under 18 dB than the Estimated C/I approach. For better channel conditions, both algorithms perform optimally.

5.7. TCP Performance with Link Adaptation (Hybrid Approach)

A hybrid approach to link adaptation is proposed. As it is predicted that EGPRS will focus on non-real time data services, the hybrid approach is selected to optimize throughput rather than delay performance under all channel conditions. Hence, it will use the Estimated C/I algorithm for C/I > 18 dB; otherwise, it will employ the BLER method. The TCP performance obtained with this proposed scheme is shown in Figure 5.30 to Figure 5.32, and is compared against ideal link adaptation.

It is observed that the hybrid link adaptation algorithm is a closer approximate to the performance of ideal link adaptation in terms of throughput, delay and delay variation for all channel conditions than the Estimated C/I or BLER approach.

As expected, the hybrid approach employs the link adaptation algorithm that optimizes throughput rather than delay under all channel conditions.

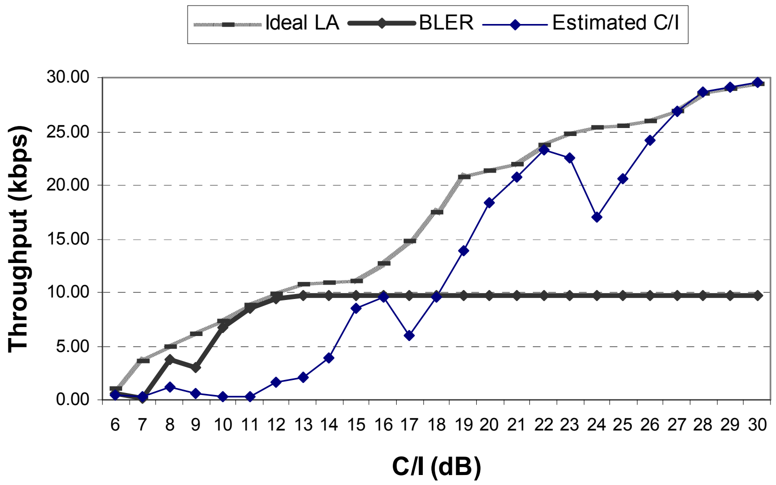


Figure 5.27 Throughput performance for different link adaptation approaches

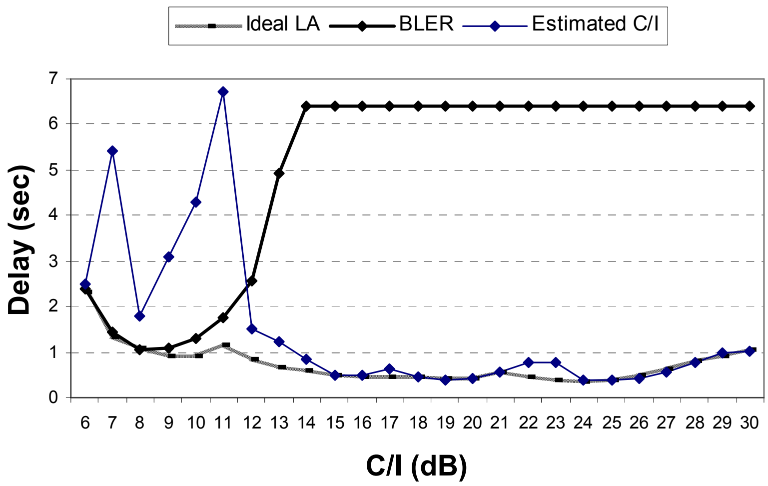


Figure 5.28 Delay performance for different link adaptation approaches

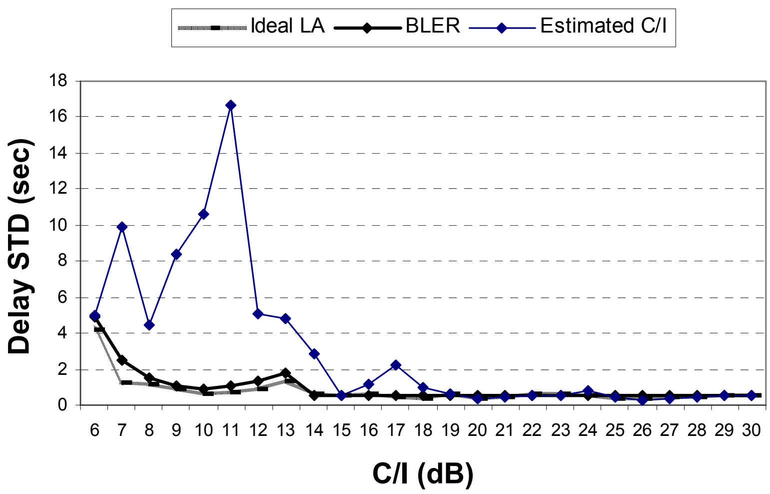


Figure 5.29 Delay variation performance for different link adaptation approaches

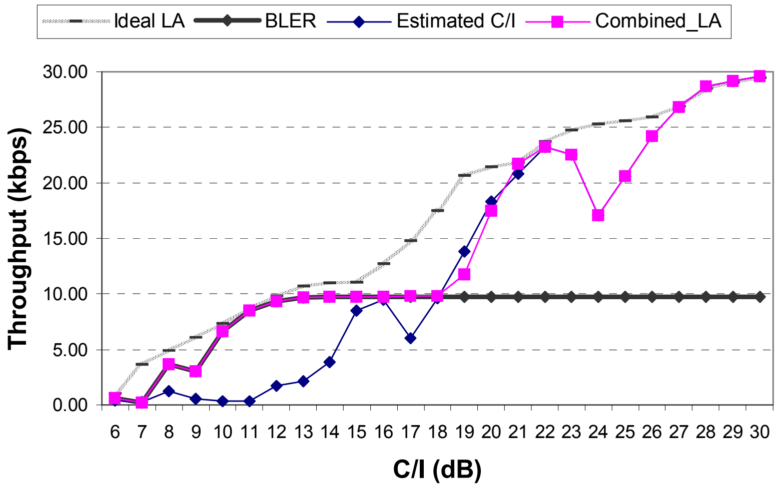


Figure 5.30 Hybrid link adaptation throughput performance

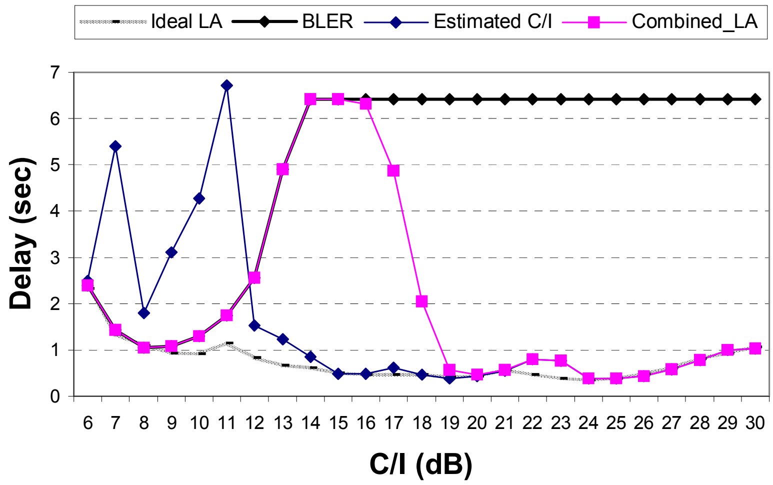


Figure 5.31 Hybrid link adaptation delay performance

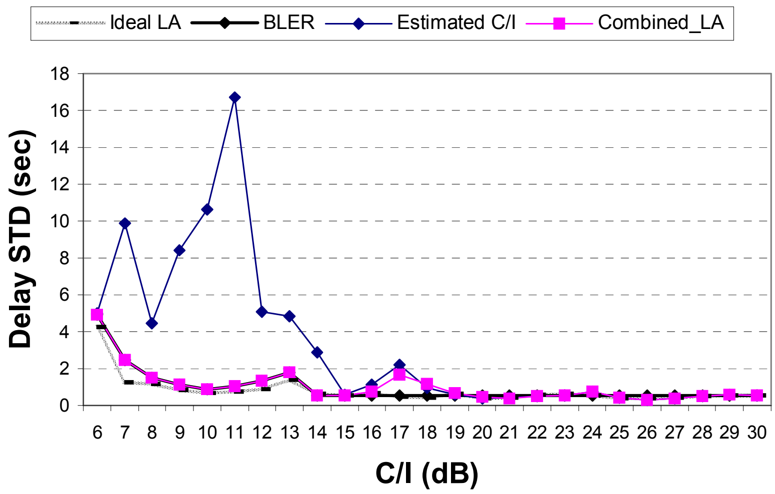


Figure 5.32 Hybrid link adaptation delay variation performance

Chapter 6

Conclusions and Recommendations

This chapter presents the conclusions and recommendations based on the simulation results obtained in Chapter 5.

6.1. Conclusions

EDGE is a radio access technology capable of providing wireless Internet access within the framework of 3G. It offers higher bit rates compared to the current 2G cellular networks such as GSM, GPRS, etc. In this study, the TCP performance over EDGE is evaluated using the *ns* simulation tool. This evaluation is carried out by measuring *throughput, average packet delay* and *delay variation* for end-to-end FTP transfer.

6.1.1. EGPRS data rate capabilities (without link adaptation)

Throughput levels up to 15.5 kbps can be achieved with GPRS-equivalent coding schemes, ECS-5 to ECS-8. With enhanced coding schemes introduced by EGPRS (ECS-1 to ECS-3), throughput levels of up to 30 kbps can be achieved for single-slot operation under good channel conditions. Hence, EDGE is theoretically able to achieve user data rates of 240 kbps with multi-slot operation, which is an improvement over GSM with data rates of 9.6 kbps, and GPRS with data rates of 171 kbps.

The data protection level in ECS-4 is inadequate to produce useful TCP performance under all channel conditions, and hence, it is not considered when implementing link adaptation.

However, the simulated throughput levels are significantly lower than theoretical throughput, S_n, computed using eqn. (5.1), which is shown here in eqn. (6.1).

$$S_n = R_n \times (1\text{-}BLER_n(C/I)) \qquad (6.1)$$

where R_n = Radio Interface Rates in bps,

and n = coding scheme.

A closer look at eqn. (6.1) indicates that it assumes that the latency introduced by re-transmission is negligible. The enhanced coding schemes in EDGE are highly susceptible to noise, and hence, re-transmissions are highly probable under poor channel conditions. The theoretical throughput is re-formulated as shown in eqn. (6.2), which attempts to consider the additional latency due to re-transmitted RLC blocks.

The simulated throughput levels obtained are now within 8 kbps of the theoretical levels. The discrepancy could be due to additional bandwidth that is needed to send signaling messages, e.g., ACK.

$$S_n = P \times M / T\_total = \frac{R_{ECS-5} \times R_n \times (1 - BLER_{ECS-5}(C/I)) \times (1 - BLER_n(C/I))}{(1 - BLER_{ECS-5}(C/I)) \times R_{ECS-5} + R_n \times BLER_n(C/I) \times (1 - BLER_n(C/I))} \qquad (6.2)$$

However, under poor channel conditions, throughput and delay performance of enhanced coding schemes of EGPRS is significantly lower than GPRS levels due to the high sensitivity to noise of the former.

An adaptive polling scheme is proposed to improve TCP performance under poor channel conditions. This yields a slight improvement in performance under poor channel conditions, but the trade-off is a slight degradation in performance when channel conditions improve.

These observations justify the need for link adaptation to achieve optimal performance under all channel conditions.

6.1.2. TCP performance of EGPRS with link adaptation

Two approaches for link adaptation [12] were implemented in this study.

6.1.2.1.Estimated C/I Approach

In terms of throughput, this approach approximates the performance of ideal link adaptation under very good channel conditions. However, it fails to achieve useful throughput performance as channel conditions deteriorate. Similar observations can be made about the delay performance parameters.

6.1.2.2.BLER approach

On the other hand, in terms of throughput, the BLER approach is able to select optimal coding schemes under poor channel conditions (*C/I* < 14 dB). However, its performance plateaus off to GPRS levels under good channel conditions.

For delay performance, this approach is able to achieve ideal link adaptation levels under all channel conditions.

6.1.2.3.Hybrid link adaptation algorithm

Since both approaches are optimal under different channel conditions, a hybrid algorithm is proposed which selects the BLER approach under poor to moderate channel conditions (up to C/I of 18 dB) and the Estimated C/I approach under good channel conditions.

Simulation results obtained show that the ideal link adaptation performance is more closely approximated with this hybrid approach.

6.2. Recommendations

The hybrid link adaptation algorithm achieves optimal TCP performance under poor to moderate channel conditions (C/I < 10 dB) as well as under very good channel conditions (C/I > 21 dB). However, under moderate to good channel conditions, the algorithm falls short of achieving optimal throughput performance.

Hence, the re-formulated theoretical throughput in eqn. (6.2) may not adequately represent the actual throughput in a practical situation. A more accurate formulation for throughput may be needed to enhance the performance of the Estimated C/I link adaptation algorithm.

In addition, there is a need to look into the incremental redundancy approach [16] to link control to supplement link adaptation.

Multi-slot operation should be carried out to demonstrate the actual capabilities of EDGE, where an end-to-end data flow between two entities can use up to 8 TSs.

At the time of writing, 8-PSK has been selected as the high-level modulation scheme to be standardized for EDGE. Hence, it will be interesting to carry out a similar evaluation based on 8-PSK instead of O16QAM.

References

1. M. Mouly, M. B. Poutet, *The GSM System for Mobile Communications*, Mouly and Pautet, 1992.

2. N. R. Prasad, “GSM Evolution towards Third Generation UMTS / IMT-2000”, pp50-54, *1999 IEEE International Conference on Personal Wireless Communications*, 17-19 Feb 1999.

3. Digital Cellular Telecommunications System (Phase 2+); *High Speed Circuit Switched Data (HSCSD)*; Stage 2 (GSM 03.34 version 5.0.1 Release 1997).

4. Digital Cellular Telecommunications System (Phase 2+); *General Packet Radio Service (GPRS); Overall description of the GPRS Radio Interface*; Stage 2 (GSM 03.64 version 7.0.0 Release 1998).

5. A. Furuskar, S. Mazur, F. Muller, H. Olofsson, “EDGE: Enhanced Data Rates for GSM and TDMA/136 Evolution”, pp56-65*, IEEE Personal Communications Magazine*, June 1999.

6. S. Hamiti, M. Hakaste, M. Moisio, N. Nefedov, E. Nikula, H. Vilpponen, “EDGE Circuit Switched Data – An Enhancement of GSM Data Services”, pp 1437-1441, *1999 IEEE Wireless Communications and Networking Conference*, 21-24 Sept 1999.

7. W. Steven, *TCP/IP Illustrated Vol. 1*, Addison Wesley, Reading, MA, 1994.

8. R. Caceres and L. Iftode, “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments”, *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 5, June 1995.

9. H. Balakrishnan, S. Seshan and R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", *Wireless Networks*, Vol. 1, 1995.

10. S. Hoff et al., "A Performance Evaluation of Internet Access via the General Packet Radio Service of GSM", pp1760-1764, *IEEE Vehicular Technology Conference*, 18-21 May 1998.

11. M. Meyer, "TCP Performance over GPRS", pp1248-1252, *1999 IEEE Wireless Communications and Networking Conference*, 21-24 Sept 1999.

12. O. Queseth, F. Gessler, M. Frodigh, "Algorithms for Link Adaptation in GPRS", pp 943-947, *IEEE Vehicular Technology Conference*, 16-20 May 1999.

13. K. Fall and K. Varadhan, "*ns* Notes and Documentation", Lawrence Berkeley National Laboratory, 28 Jan 2000, http://www-mash.cs.berkeley.edu/ns/.

14. The CMU Monarch Project's Wireless and Mobility Extensions to *ns*, Aug 5 1999, http://monarch.cs.cmu.edu/cmu-ns.html.

15. P. Schramm, H. Andreasson, C. Edholm, N. Edvardsson, M. Hook, S. Javerbring, F. Muller, J. Skold, "Radio Interface Performance of EDGE, a Proposal for Enhanced Data Rates in Existing Digital Cellular Systems", pp1064-1068, *IEEE Vehicular Technology Conference*, 18-21 May 1998.

16. H. Olofsson and A. Furuskar, "Aspects of Introducing EDGE in Existing GSM Networks", pp1284-1289, *1998 IEEE International Conference on Universal Personal Communications*, 5-9 Oct 1998.

17. Digital Cellular Telecommunications System (Phase 2+); *General Packet Radio Service (GPRS); Multiplexing and multiple access on the radio path*; Stage 2 (GSM 05.02 version 6.4.1 Release 1997).

18. A. Furuskar, M. Frodigh, H. Olofsson, J. Skold, “System Performance of EDGE, a Proposal for Enhanced Data Rates in Existing Digital Cellular Systems”, pp1284-1289, *IEEE Vehicular Technology Conference*, 18-21 May 1998.

19. R. V. Nobelen, N. Seshadri, J. Whitehead, S. Timiri, “An Adaptive Radio Link Protocol with Enhanced Data Rates for GSM Evolution”, pp54-66, *IEEE Personal Communications Magazine*, Feb 1999.

20. S. Eriksson, A. Furuskar, M. Hook, S. Javerbring, H. Olofsson and J. Skold, “Comparison of Link Quality Control Strategies for Packet Data Services in EDGE”, pp 938-942, *IEEE Vehicular Technology Conference*, 1999.

21. D. L. Lough, T. K. Blankenship, K. J. Krizman, “A short tutorial on Wireless LANs and IEEE 802.11”, The Bradley Dept of ECE, Virginia Polytechnic Institute and State University, Blacksbury, Virginia.

22. Digital Cellular Telecommunications System (Phase 2+); *General Packet Radio Service (GPRS); Mobile Station (MS) – Base Station System (BSS) Interface*; Stage 2 (GSM 04.60 version 7.0.0 Release 1998).

23. H. P. Tan, A. Lo, W. Seah, "Performance of TCP/IP over EDGE", to appear in Lecture Notes in Computer Science, Heidelberg, Springer-Verlag.