Analysis of the Impact of SR-ARQ on the IP Packet Delay Performance in Mobile Data Networks

By: **Hubert GRAJA**

The thesis is submitted to University College Dublin
for the degree of PhD
in the College of Engineering, Mathematical
& Physical Sciences

August 2006

The School of Computer Science and Informatics,
Performance Engineering Laboratory

Head of the School: Prof. Barry SMYTH

Supervisors: Dr. John MURPHY and Dr. Philip PERRY

CONTENTS

Abstract

To achieve Quality of Service (QoS) in mobile data networks is a challenging task mainly due to the specific behaviour of the radio channel. Radio links are characterized by large fluctuations of quality because they are prone to transmission errors. A set of techniques have been developed to maximize their reliability and one of the most common solutions is called Selective Repeat Automatic Repeat reQuest (SR-ARQ). This retransmits previously corrupted Data Blocks and therefore increases the probability of a successful transmission of an IP packet (which may consist of several Blocks). However, the price for higher reliability is higher delay, as every retransmission attempt introduces an additional delay to the IP packet transmission. Thus, being able to analyse and predict the influence of the SR-ARQ loop on total IP packet delay gives a better understanding and description of the radio link that uses SR-ARQ. This knowledge can be used by other protocol layers as a descriptor of radio link performance at the IP packet level.

This thesis consists of a new methodology for analysing the influence of an SR-ARQ loop on IP packet delay in Mobile Data Networks. The proposed methodology uses input data that is easily gathered and, unlike other techniques, privides insight into the influence of Hybrid Type II/III SR-ARQ techniques on IP packet delay performance. A simulator has been designed and implemented and the methods have been tested for different radio link conditions and different SR-ARQ loop designs. This results in a prediction of when and where an average value of IP packet delay can be used with good accuracy. Additionally, two analytical methods to predict the influence of SR-ARQ on the average IP packet delay are proposed: the first one is based on an algorithm which is characterized by extensive computation; the second algorithm requires lower computation complexity and has been proposed to speed up the decision. To examine the error introduced by these prediction algorithms, results from both methods are compared with simulation results. All the tests show that the proposed methods exhibit an acceptable level of accuracy.

Glossary

ACK Acknowledgment

AIMD Additive-Increase Multiplicative-Decrease

API Application Program Interfaces

ARQ Automatic Repeat reQuest

BEC Backward Error Correction

BER Bit Error Rate

BF Brute Force

BLER Block Error Rate

BS Base Station

BSS Base Station Subsystem

CBFQ Credit-Based Fair Queueing

CBR Constant Bit Rate

C/I Channel to Interference ratio

CIF-Q Channel-Condition-Independent Fair Queuing

CRC Cyclic Redundancy Code

CS Coding Schemes

CSDPS Channel State Dependent Packet Scheduling

Delay-EDD Delay-Earliest-Due-Date

DRR Deficit Round-Robin

DSSS Direct Sequence Spread Spectrum

EDF Earliest Deadline First

EGPRS Enhanced General Packet Radio Service

FAF(f) Floor Attenuation Factor

FEC Forward Error Correction

FH Frequency Hopping

FIFO First In First Out

GGSN Gateway GPRS Support Node

GPS Generalized Processor Sharing

GSM Global System for Mobile Communication

HOL Head of the Line

HRR Hierarchical Round-Robin

i.i.d. Independently and Identically Distributed

IP Internet Protocol

IR Incremental Redundancy

ISI Inter Symbol Interference

ITU-T Telecommunication Standardization Section of International Telecommunication Union

IWFQ Idealized Wireless Fair Queuing

LA Link Adaptation

LCC Low Computation Complexity

LLC Logical Link Control

LOS Line of Sight

LQF Longest Queue First

MAC Medium Access Control

MCS Modulation and Coding Scheme

MS Mobile Station

NACK Negative Acknowledgment

NLOS Non Line of Sight

OFDM Orthogonal Frequency Division Multiplexing

OLOS Obstructed Line of Sight

PGPS packet-by-packet GPS

PHY Physical Layer

QoS Quality of Service

RCSP Rate-Controlled Static Priority

RLC Radio Link Control

RR Round Robin

RTO Retransmission Timeout

Rx Receiver

SBFA Server-Based Fairness Approach

SCFQ Self-Clocked Fair Queuing

SGQ Stop-and-Go Queuing

SGSN Serving GPRS Support Node

SMS Short Message Service

SNDCP Sub Network Dependent Convergence Protocol

SR-ARQ Selective Repeat - Automatic Repeat reQuest

σ_τ root mean square (rms) delay spread

TCP Transmission Control Protocol

Tx Transmitter

UDP User Data Protocol

UMTS Universal Mobile Telecommunications System

WAF(w) Wall Attenuation Factor

WFQ Weighted Fair Queuing

WFS Wireless Fair Service

WLAN Wireless LAN

WRR Weighted Round-Robin

VC Virtual Clock

VoIP Voice over IP

Acknowledgments

There are several people who I would like to thank for their help and support.

First of all I would like to thank my supervisors, John MURPHY and Philip PERRY, who have continually motivated and supported me through my PhD studies. I am very grateful for all their help, both that which I asked for and that which I didn't realize I needed but they gave anyway. I would particularly like to thank them for their patience in correcting errors related to my less than perfect English.

I would also like to extend my thanks to my friends from the Performance Engineering Laboratory and my Polish friends for helping me when I needed it most.

I thank my family in Poland, for their unconditional guidance and support throughout my years in University. In particular, I thank my Mum who instilled in me the confidence to believe that I could and would complete this work.

Finally, I would like to thank my wonderful wife, Juebei XIAO, for giving me a fresh and inspiring outlook on life. She has the unique power of changing for the better some of the unproductive habits which a stubborn man such as me possesses.

CHAPTER 1

Introduction

1.1 Motivation

The deployment of Global System for Mobile Communication (GSM) networks in the early 1990's initiated the era of digital mobile communication, with a reliable and affordable voice service. Soon after the deployment of these networks Short Message Service (SMS) has become popular. This popularity was an indicator of the demand for data transmission services in mobile networks. Therefore, more advanced systems, like Enhanced General Packet Radio Service (EGPRS) or Universal Mobile Telecommunications System (UMTS) - called mobile data networks, have been designed and deployed to facilitate the growing demand for data transmission with mobile terminals.

Mobile data networks are increasing both the range of offered data services and are becoming a significant provider of data services based on the Internet Protocol (IP). More recently the Multimedia (e.g. Video Streaming) and Voice over IP (VoIP) traffic types are increasing the occupied portion of the total transmitted data. These facts lead to the conclusion that mobile data networks have to be ready to accommodate this growing demand for Multimedia and VoIP transmission. Nonetheless, both of these data types are delay sensitive and the mobile data networks have not been designed to facilitate these strict delay requirements.

There is a great amount of work addressing the issue of these delay requirements at the fixed side of the mobile data network. In new releases of (E)GPRS and UMTS networks the core transmission delay is substantially reduced and the concept of Quality of Service (QoS) has been introduced. However, the wireless part of these networks still significantly degrades the Multimedia and VoIP traffic due to the delay impairment.

The delay caused by the wireless part of mobile data networks has its roots in the nature of the radio channel, which often experiences high fluctuations of its signal strength during the transmission. Nonetheless, these systems have highly sophisticated methods of both adjusting the amount of radio resources assigned to the particular connection, in the form of Medium Access Control (MAC) scheduling policies, and improving the reliability of this connection, in the form of Radio Link Control (RLC) techniques, which aim to sustain good properties of the wireless link.

However, there is a price to be paid for these improvements. The MAC scheduling policy has a limited number of channels, hence, the improvement on a single link delay performance can be achieved only by limiting other connection's share of the resources. The RLC is facing a similar dilemma, an improvement in the reliability of the radio link causes degradation of the throughput and delay performance. The use of Forward Error Correction (FEC) brings throughput performance degradation, as the number of radio blocks necessary to carry a portion of data increases with the increase of FEC protection level. Whereas, the use Backward Error Correction (BEC), named also Automatic Repeat reQuest (ARQ), has a negative effect on the transmission delay characteristic, since the retransmissions of errored radio blocks may introduce a substantial delay.

The issue of MAC influence on the delay characteristic of the wireless link has already been addressed in great detail in the literature. On the contrary, the influence of the RLC layer, and sophisticated ARQ mechanisms in particular (like Hybrid Type I/II/III ARQ Schemes), on the IP packet delay characteristic has not yet been fully investigated and is the main topic of this thesis.

The introduction of the ARQ loop into the RLC layer significantly improves its reliability. However, this improvement has a profound side effect, which is a degradation of the delay performance. Multimedia and VoIP traffic benefit from the lowered loss rate, however, they suffer from the increased delay. Firstly, because end users have low delay tolerance for the traffic carrying Multimedia or VoIP content . Secondly, due to the fact that, in the case of Multimedia or VoIP transmission, an excessive level of delay can turn into a loss. Basically, if packets arrive too late, the content carried by those packets may be excluded from being played out.

1.2 Scope

In order to investigate the influence of ARQ techniques on the IP packet delay this influence has to be separated from other dynamic mechanisms introduced at the link layer. Hence, the Logical Link Control (LLC) and MAC layers have to be simplified. This simplification should allow for investigating both the character and the level of ARQ influence. Consequently, the MAC is set to work in a simple Round Robin regime. Whereas, LLC layer is set to operate with the retransmission inactive mode.

Another important aspect of the RLC influence analysis is to separate the dependency on the characteristic of the data source load. This is achieved by analysing the IP packet delay, as measured from the moment of the transmission of the first radio block belonging to the analysed IP packet until all radio blocks being a part of the analysed packet are successfully received at the receiver buffer.

Due to the significant development of ARQ techniques, resulting in Hybrid Type I/II/III ARQ techniques, it is important to examine the influence of those methods on the IP packets delay. Therefore, the analysis of ARQ loop on IP packet delay presented in this thesis takes into the account those advanced ARQ methods.

Because the analysis of the ARQ influence on the IP packet delay may lead to some prediction methods, it is important to focus on the simplicity of possessing the input data for both the analysis and the prediction of the ARQ influence. Hence, the model proposed in this thesis is based on input parameters that are gathered in a straightforward way. Thus, the complexity of the proposed solutions have been developed at a level which allows the implementation of the potential prediction methods in real time systems.

1.3 Thesis contribution

The main contributions presented in this thesis are the following:

- A new model of the ARQ loop has been proposed. This model is designed to be able to capture and analyse all aspects of the ARQ transmission in terms of IP packet delay. An important property of this new model is its ability to gather the input data in an easy manner from an existing system.

  - A new method of representation of the influence of PHY layer on radio block transmission has been proposed, denoted $\overline{P}$ vector. This new approach allows the analysis of the influence of advanced ARQ techniques, like Hybrid Type I/II/III ARQ, on IP packet delay.
  - A new representation of the delay associated with the process of transmission attempts has been introduced, called $\overline{\Delta}$. This new description accommodates a different ARQ loop delay associated with each transmission attempt of a transmitted radio block.

- The ARQ component of the IP packet delay distribution has been analysed by means of simulation.

  - It has been shown that the average value of this delay can be a good descriptor of the ARQ influence on the IP delay for IP packets of medium and large sizes.
  - In has been found that for small IP packets the average delay needs to be used with an awareness of the non-uniform and non-Gaussian shape of the ARQ component of the analysed IP packet delay distribution. This knowledge about the expected non-uniform delay distribution shape can be passed to higher layers to prevent the adaptation mechanisms at higher layers from being mislead.

- Two prediction algorithms estimating the ARQ influence on average packet delay have been proposed.

  - The first algorithm is called Brute Force (BF). It offers good accuracy but has complexity that grows exponentially with the size of the analysed packet. This excludes the BF method from being used in a real time system for a wide range of packet sizes.
  - The second algorithm, named Low Computation Complexity (LCC) is based on the BF method for small IP packet sizes. This method gives accuracy of the same order of magnitude as BF, whereas the level of its complexity is significantly reduced in comparison to the BF one. This complexity reduction together with the easily obtained algorithm input data makes this algorithm a good candidate for being implemented in a real time system.

1.4 List of Publications

The publications arising from the work presented in this thesis are the following:

- **Hubert GRAJA**, Philip PERRY and John MURPHY,"A Low Complexity Algorithm for Estimation of Average IP Packet Delay in Cellular Data Networks", Proc. International Teletraffic Congress, ITC-19, pp. 2471-2480, Vol 6c, ISBN 7563511415, Beijing, China, August 2005.

- **Hubert GRAJA**, Philip PERRY and John MURPHY, "A Statistical Estimation of Average IP Packet Delay in Cellular Data Networks, Proc. of IEEE Wireless Communications and Networking Conference (WCNC'05), pp. 1273 - 1279, Vol. 3 , ISSN: 1525-3511 , New Orleans, USA, March 2005.

- **Hubert GRAJA**, Philip PERRY and John MURPHY, "An Analysis of Delay of Small IP Packets in Cellular Data Networks", Proc. of IEE 3G 2004: Fifth

International Conference on 3G Mobile Communications Technologies, pp. 347-351, IEE Press, ISBN 0 86341 388 9, London, UK, October 2004.

- Doru TODINCA, **Hubert GRAJA**, Philip PERRY and John MURPHY, ″Novel Admission Control Algorithm for GPRS/EGPRS Based on Fuzzy Logic”, Proc. of IEE 3G 2004: Fifth International Conference on 3G Mobile Communications Technologies, pp. 342-346, IEE Press, ISBN 0 86341 388 9, London, UK, October 2004.

- **Hubert GRAJA**, Philip PERRY and John MURPHY, ″A Statistical Analysis of IP Packet Delay and Jitter in Cellular Networks” , Proc. of IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'04), pp. 1881-1885, Vol.3, ISBN: 0-7803-8524-1,Barcelona, Spain, September 2004.

- **Hubert GRAJA**, Philip PERRY, Doru TODINCA and John MURPHY, ″Novel GPRS Simulator for Testing MAC Protocols, Proc. of IEE 3G 2003: Fourth International Conference on 3G Mobile Communications Technologies, ISNN 0537-9989, pp. 409-412, IEE Press, ISBN 0-85296-756-X, London, UK, June 2003.

- **Hubert GRAJA**, Philip PERRY and John MURPHY, ″Development of a Data Source Model for a GPRS Network Simulator, IEI/IEE Postgraduate Symposium Telecommunications Systems Research, 6 pages, Dublin, Ireland, November 2001.

Other publications

- **Hubert GRAJA** and Jennifer McMANIS,″Modelling User Interaction with E-commerce Servers”, in Lecture Notes in Computer Science, ISSN: 0302-9743 , Volume 2093 / 2001, Part 1, pp. 803-810, Title: Networking - ICN 2001: First International Conference Colmar, France, July 2001.

- **Hubert GRAJA** and Jennifer McMANIS, ″Quantifying Customer Satisfaction with E-Commerce Websites”, in Proc. of Seventeenth IEE UK Teletraffic Symposium,pp. 25/1-25/6., Dublin, Ireland, May 2001.

1.5 Organisation of the Thesis

The remainder of this thesis is organised as follows:

Chapter 2 addresses the issue of transmission delay influence on users perceived quality. All communication layers are briefly examined with respect to the introduced delay. The significant influence of data link delay performance on the total transmission delay in mobile data networks motivates a deeper investigation of the mechanisms altering this delay.

Chapter 3 offers a literature review of the main mechanisms aiming to improve the delay performance at the TCP layer. Subsequently, different wireless MAC scheduling policies are examined in respect to their throughput and delay performance. Following that, the main wireless PHY models that are used to analyse the link layer delay are shown. The work regarding ARQ loop influence on the link layer delay characteristic is analysed. The chapter concludes that there is a need for deeper analysis of the RLC delay component of the IP packet delay characteristic, in particular the influence of ARQ techniques.

Chapter 4 presents a new methodology for analysing the ARQ loop. Following that, this methodology is used to investigate the trend of average ARQ component of IP packet delay across different radio conditions, ARQ loop delays and IP packet sizes. Concluding that the average ARQ component of IP packet delay can be used to describe RLC delay performance for medium and large IP packets.

Chapter 5 introduces two methods of predicting the ARQ component of average IP packet delay, named Brute Force (BF) and Low Computation Complexity (LCC). These methods are then examined with respect to their prediction error level. These comparisons show that both methods offer an acceptable prediction error over a range of IP packet sizes.

Chapter 6 explains how the LCC technique could be implemented in an EGPRS network and what adjustments are necessary to implement the LCC method. Additionally, several values of $\overline{P}$ are computed on the basis of the radio channel definitions given in 3GPP recommendations. These $\overline{P}$ vectors, together with the most common $\overline{\Delta}$ and IP packet sizes, are used to present some examples of the values of ARQ loop component of average IP packet delay that are likely to be experienced in the real world.

Chapter 7 delivers a summary of the content of this thesis and presents its major conclusions

CHAPTER 2

Background

This chapter aims to expose the different network mechanisms affecting end-to-end loss and delay performance and consequently the user's perception of offered services. The main emphasis is put on the influence of mobile data networks on the IP packet delay, as these networks introduce a substantial delay component to the total end-to-end delay. This will set the content of later discussions about the impact of ARQ techniques on the IP delay, which is the subject of this thesis.

Firstly, the impact of delay and loss characteristics on the user's service perception is going to be addressed. Following that, all protocol layers of the TCP/IP protocol suite will be briefly examined with respect to the delay they may introduce. Next, mobile data networks are surveyed with respect to their data transmission techniques and the impact of those techniques on IP packet delay.

2.1 Services' demands and QoS of networks

Services differ according to their resource demands and traffic characteristics. It is important to match these demands with the level of data transport capabilities offered by the network.

Telecommunication Standardization Section of International Telecommunication Union (ITU-T) has determined that the two primary factors in user perception of QoS are delay[1] and loss. The acceptable ranges for delay and loss are dependent on the type of data. The ITU-T identified eight classes of applications

[1] The delay variation is also considered by ITU-T as the key parameter impacting the user. ”However, services that are highly intolerant of delay variation will usually take steps to remove (or at least significantly reduce) the delay variation by means of buffering, effectively eliminating delay variation as perceived at the user level (although at the expense of adding additional fixed delay)” [64]. Therefore, it is not considered here.

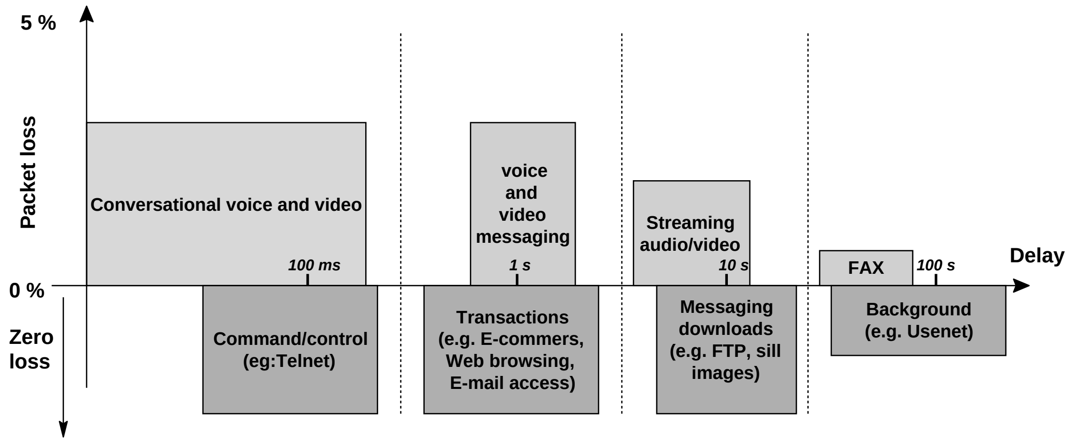


Figure 2.1: *Delay/loss requirements for various Internet applications [64]*

based on tolerance to loss and delay resulting from the type of user interaction [64], figure 2.1.

For example delays of seconds will be acceptable to Voice or Video messaging but not sufficient for Conversational services, where the total end to end delay must be much less than one second in order to satisfy the client's quality expectations. However, the same value of delay (of about one second) will cause no harm to a unidirectional transmission of Streaming audio and/or video.

Therefore, knowledge about the expected delay and loss characteristic may enable a process of matching the demand of the service to the network QoS performance. This matching process is difficult enough to fulfil in wired networks, where the physical link condition does not fluctuate in time. To achieve it when the connection consist of a wireless part is much more challenging.

Assuming that the hardware and software at both end points introduce negligible delay and zero loss, the connection characteristics will be affected by the performance of the networks that connect both end points. Thus, this characteristic is worth examination.

Since the first telecommunication network emerged, a countless number of networks have been created. However, in the domain of data transmission one type of network holds the dominant position; those networks based on the Internet Protocol (IP). Its popularity and strength is so large that even in the strict telecommunication sector it is becoming a very important protocol. Voice over IP (VoIP) [103] and Universal Mobile Telecommunication System (UMTS) working with the IP protocol are a good example of this process.

IP is usually deployed with the Transmission Control Protocol (TCP) to form the well known TCP/IP suite of protocols. Hence, the structure of the TCP/IP protocol stack with its influence on service quality is going to be examined in the following sub-sections.

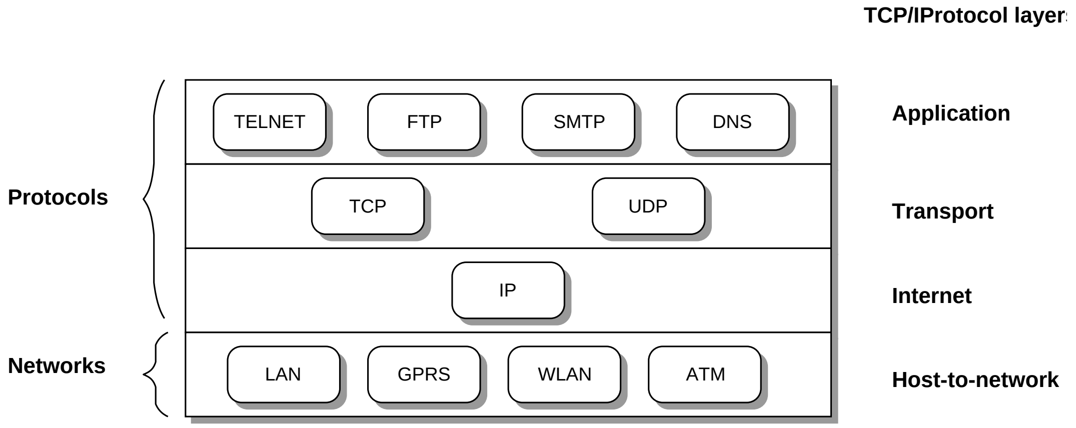


Figure 2.2: *Protocols and networks in the TCP/IP model [130]*

2.2 Overview of TCP/IP suite

The TCP/IP model is the evolving result of work that started with the ARPANET project. Its main aim was to create a protocol stack that would be able to connect multiple networks in a seamless way. Additionally, if any of the connection points were suddenly put out of operation, the connection could be maintained as long as there was a path connecting the two destination points.

To fulfil this requirements the following protocol stack was developed, as shown in figure 2.2 [2], [130]. It consists of the following layers: Application, Transport, Internet and Host-to-network layers. All of these layers will be described briefly and examined with respect to their impact on connection delay and loss characteristics.

2.2.1 Application layer

The application layer contains a set of protocols that are considered as high-layer protocols. Usually, the Application Program Interfaces (API) of these protocols are used by software developers to create a certain type of connection. For example a service to provide remote access to a computer can be realized by using the TELNET protocol, while an application for browsing the Internet would use the HTTP protocol. Initially only a few application protocols existed, but with time more and more have been added, to extend the flexibility of offered services. Among the most popular protocols are:

- **TELNET**, a protocol that emulates the terminal of a remote computer. It allows for remote control of a computer that is connected to the network.
- **FTP** File Transfer Protocol, a protocol that addresses the issue of transporting a file over the network between two end points.

[2] Some of the protocols presented in the picture 2.2 can work with other than TCP/IP networks. However, in this work they will be discussed in TCP/IP context.

- **DNS** Domain Name Service, a protocol that translates domain names into IP addresses.
- **HTTP** - Hyper Text Transfer Protocol, a protocol that defines how messages are formatted and transmitted during a WWW session.
- **H.323** and **SIP** protocols that are used when streaming audio or video is transmitted over a network. This protocols deliver necessary signalling mechanisms to establish and control these kind of transmissions.
- **RTP/RTCP** - Real-time transport protocol is a protocol for the transport of real-time data across networks. RTP is used in the majority of Voice-over-IP (VoIP) architectures, for videoconferencing, and other kinds of multimedia applications. RTCP is a companion protocol of RTP that is used to monitor and maintain QoS in an ongoing RTP connection.

Each of these protocols will create different data stream characteristics and will have different interaction with the network beneath. For example, TELNET will generate an occasional stream of bytes, each representing a single character; HTTP will be a highly bursty source of data, with an ON/OFF data transmission characteristic; and RTP generates a stream of small files on a frequent basis when a VoIP connection is handled.

The delay performance of a particular service or application can be degraded when the chosen lower layer protocol behaviour does not match the requirements of the traffic generated by this service. Thus, the main source of problems with delay and loss performance at the Application layer is the appropriate protocol implementation that will assure the best way of interaction between the different layers in the complete protocol stack. Nonetheless, this is beyond the scope of this thesis and will not be addressed here.

2.2.2 Transport layer

This layer is designed to transport data between peer entities on source and destination hosts. Two end-to-end transport protocols, with very different characteristics, have been defined to let this happening. They are Transmission Control Protocol (TCP) and User Data Protocol (UDP).

2.2.2.1 TCP

The role of the TCP protocol is to enable two hosts to establish a connection and exchange streams of data. TCP offers a reliable connection, meaning that it guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent. TCP supports full duplex transmission. Additionally, TCP performs flow and congestion control operations.

To fulfil such a rich functionality TCP has implemented several mechanisms, which are briefly described below:

- **Segmentation:** TCP treats data coming from application layer as a stream of bytes. This stream is fragmented into a number of segments of a size specified by the TCP. The most common segment size is 1460 bytes[3] due to the constraint of minimising the likelihood that segments will need to be fragmented for transmission over intermediate IP networks.
- **Reliability:** Each segment can be recognized during the transmission by the sequence number, which allows for de-segmentation at the receiver even if these segments have been delivered out of sending order. If there are holes in the segment stream at the receiver side, then the Acknowledgment (ACK) for correctly received segments include segment numbers of these missing segments. In this way the receiver informs the sender that some segments are missing. If more than two such ACKs are received by the transmitter in regards to a particular segment then this segment is retransmitted. The other situation when a segment is retransmitted is if an ACK for this segment does not reach the sender before its RTO timer expires.
- **Flow control:** Before any transmission occurs an initiation of this connection must be performed. It allows exchange of information about the sender and receiver limitations. Maximum buffer sizes at both sides are exchanged, which allows flow control by not sending any more data than the receiver buffer can store.
- **Congestion control:** makes sure that the stream of segments sent from a transmitter is not too large for the network. It uses the slow start mechanism to test how many segments can be sent at once without acknowledgment. That is, to estimate the connection throughput. Initially one segment is sent and waits for an ACK. When the appropriate ACK reaches the sender, two segments are sent and the sender waits for the confirmation of their correct reception. The number of segments sent without being acknowledged is doubled each time the previous attempt finishes successfully. Eventually, some packets are lost, or delayed long enough to be considered lost in the network. In this event, the number of packets sent without retransmission is reduced by half and this number is increased linearly with each received

[3] Cisco's default Maximum Transmission Unit(MTU) for connection speeds $>$ 128 Kbps is 1500 bytes: Cisco,Troubleshooting MTU Size in PPPoE Dialin Connectivity, Document ID: 12918, Updated: Jun 01, 2005, `http://www.cisco.com/warp/public/794/router\_mtu.html`. Since Maximum Segment Size (MSS) = MTU - 40 bytes (for IP and TCP headers), then MSS=1460.

ACK. On the contrary, if the segment is considered delayed, meaning its Retransmission Timeout (RTO) has passed and no ACK was delivered to the sender, then the number of segments sent without ACK goes back to slow start: one, two, four etc. However, when this number reaches half of the value achieved before the delayed packet was sent, then the progress becomes linear.

The mechanisms described here assure high reliability and adjusting the data transmission rate to the current network path condition. But, there is a price to be paid. The average delay of segments, and the content it carries, will increase due to retransmissions. Furthermore, the TCP scheme was design for wired networks, where expected loss of segments is very small, less than 1 %. Additionally, in wired networks the delay is caused mainly by congestion at a node rather than by a failure of the physicial link. That makes TCP vulnerable to fluctuations of path delay experienced in a connection that contains a wireless link at the end.

A substantial amount of work has been done addressing the issue of improving TCP throughput and delay performance when a wireless connection is involved in the communication path. However, there seems to be no solution that solves all problems.

2.2.2.2 UDP

User Datagram Protocol provides a way to send data without having a previously established connection. It is very usefully to quickly send simple and short messages without the need of pre-establishing a connection between two hosts. However, due to its simplicity, it offers no reliability to the transported data. If the data sent within a particular UDP packet is lost at the Internet layer, UDP will not retransmit it. Thus, the application itself has to introduce some retransmission technique. On the other hand, UDP introduces a very small delay to the transmission, as neither retransmission nor congestion control techniques are deployed. Hence, during the UDP based transmission, if a connection path consists of a radio based connection, then the loss performance of the connection will be severely affected.

2.2.3 Internet layer

The purpose of this layer is to enable interconnection of different packet networks for sending packets of data between destination points. This layer focuses mainly on routing data across the same or different type of networks. What it offers is unreliable and connectionless transport of packets through networks. Higher layers have to assure a reliability of the connection path, eg. TCP.

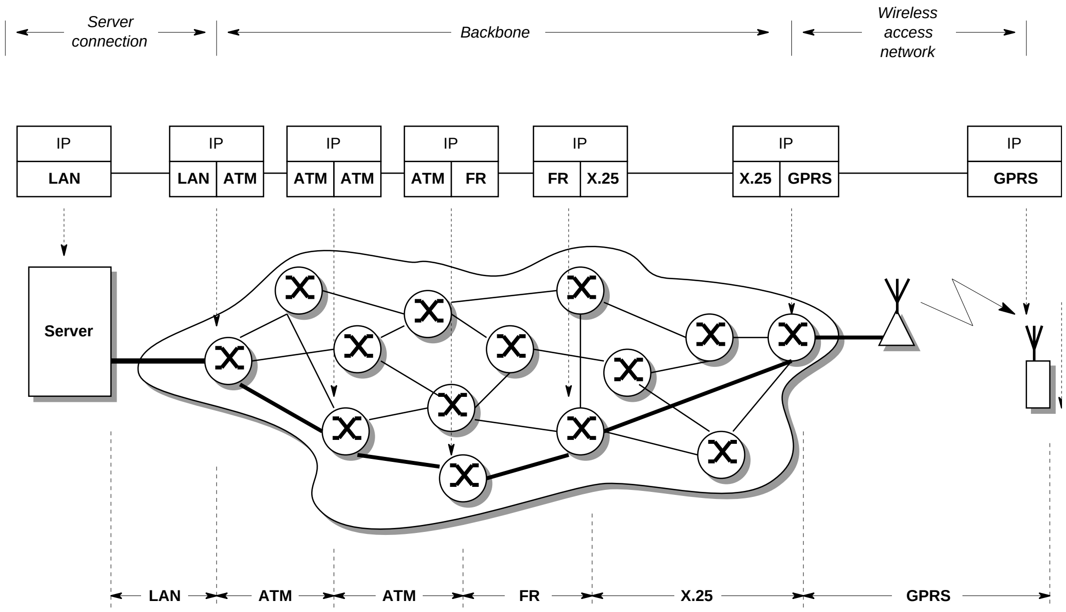


Figure 2.3: *Example of IP based transmission with connection across different network technologies*

The routing algorithm implemented at the Internet layer plays a crucial role in minimizing loss and latency of data packets, by selecting the path that introduces the smallest possible delay. Nevertheless, the routing in backbone of Internet is in most cases static and hierarchical. A low level of IP packet loss and delay is assured mainly by over-provisioning capacity of connection links and routers' transmission speed.

The limited functionality of IP means that the IP layer can only negatively affect the data loss and delay characteristics if the chosen path contains nodes that are overloaded. The node that is overloaded can cause degradation of packet delay. If this node experiences saturation for long time, then buffers may overflow resulting in loss of some IP packets. Therefore, assuming a stable and saturation free node scenario in the chosen path, the IP layer introduces quite static and predictable delay.

Figure 2.3 shows an example of a connection across a few different networks.

2.2.4 Host-to-network layer

This part of the TCP/IP protocol suite is the least specified one. This is because, from the connectionless Internet layer point of view, it is not important what kind of medium and protocol is carrying an IP packet. The only necessary thing is a common interface with IP protocol, so, the packet can be encapsulated into and retrieved from the appropriate network data format.

Therefore, there is a great number of networks that can be used to carry TCP/IP packets. Some of them are listed below:

- **Ethernet** is a network standard of communication using fixed data links in the form of either coaxial or twisted pair cable. It typically operates at 10 or 100 Mbps or even 1 Gbps of data transfer.

- **ATM** - Asynchronous Transfer Mode, is a network technology designed to transfer data in small packets (cells) of a fixed size. The small and constant packet size allows ATM technology to transmit simultaneously data streams with different delay, throughput and loss requirements.

- **X.25** one of the oldest packet data protocols in use. It allows computers on different public networks to exchange data through intermediary nodes at the network layer level.

- **Satellite links** are characterised by high asymmetry of the offered throughput, slow feedback response and high cost of the equipment. This asymmetry makes this kind of link ideal for Web browsing, whereas the high feedback delay causes problems when multimedia content is transmitted.

- **WLAN** - Wireless LAN network. Characterised by short range and high throughput capacity between 1 to 54 Mbps at the air interface. Due to low cost of the WLAN equipment it is used mainly as the Internet access technology in offices and home environments.

- **EGPRS** - Enhanced General Packet Radio Service. A 2.5G kind of network built upon the GSM standard. Offers high coverage range with moderate throughput between 10 and 52 kbps at the air interface.

- **UMTS** - Universal Mobile Telecommunications System. Represents 3G type of system, that is characterised by throughput between 64 kbps up to 2 Mbps at the air interface. It is designed to facilitate both voice and data traffic.

Each of these networks have their own delay and loss characteristic. Thus the connection commenced through a particular network will have this characteristic influencing the total end to end delay and loss performance. It is quite obvious that expensive and fast networks will be placed in the core of the Internet connection, e.g. Gigabit Ethernet or ATM, figure 2.3. Similarly, the connection between server and the backbone network will have a fast and expensive interface. The connection to a single user will be characterized by a low cost and lower performing solution, e.g. 10Mbps Ethernet. Thus, the performance of the last network interface will often play a crucial role in the end-to-end loss and delay characteristic.

As wireless connections become more and more popular as Internet access networks, eg. Wireless LAN (WLAN), (E)GPRS and UMTS, the last network interface becomes even more susceptible to delay and loss fluctuations. Therefore, it is important to be able to improve this part of the connection or at least predict the expected delay and loss level. The ability to predict the delay component introduced by potential retransmissions of data at the wireless link in mobile data networks is a core goal of this thesis. Such a prediction could help to improve the performance the layers above, e.g. TCP or Application. TCP could adjust the RTO calculation in a more suitable way, thus it would not suffer too much from radio link fluctuations. The application itself could adapt the rate and the characteristic of the transmission, according to the expected network performance.

The types of wireless network will be examined in more detail in the next section. This will be done with particular reference to their significant influence on the total end-to-end delay and loss performance and because wireless networks become more and more popular as the method for accessing the Internet.

2.3 Types of wireless data networks

Wireless access technologies can significantly impact the performance of the connection. First of all the quality of the link depends greatly on the position of the receiver in relation to the transmitter. Additionally, the throughput, delay and loss characteristic of a wireless link changes with time, due to mobility of a transceiver or due to changes in the environment that the network is operating in. Finally, the up-link does not have to have the same characteristic as down-link, which may mislead TCP RTO timers.

Therefore, a brief review of the most popular and widespread wireless access technologies is presented below. Following that, more details describing mobile data networks are presented. The main emphasis is put on those mechanisms that most influence the loss and delay characteristic of IP packets transported over mobile data networks.

Wireless data networks can be categorized into three major families. They are: the 802 wireless networks family, mobile data networks and satellite networks. The first two are widely used as bidirectional Internet access technologies. The last one, satellite network technology, is mainly used in intercontinental or intercountry backbone connections and is not going to be examined further here.

2.3.1 Wireless 802 network family

The wireless 802 network family emerged from computer networks. Initially the main idea was to replace the last few meters of LAN cable connections by a wireless link (the 802.11 network group). In such a scenario a deployment of computer networks could be much more easy and flexible. Thus, mobility is not a key design issue in these networks and the cost of a single base station and computer receiver should be kept as low as possible. For the same reasons the design of these units need to be as simple as possible. Other 802 wireless technologies follow the same design principles of simplicity and low cost. Therefore, such networks don't use much of the sophisticated and expensive solutions available in current radio transmission technology. Hence, these networks will not be used as a model of further performance evaluations. In the thesis, the more sophisticated networks, represented by mobile data networks, will be examined.

The wireless data networks belonging to 802 family are the following:

- **WMAN** - Wireless Metropolitan Area Network . The standard that aims to address this type of networks is the 802.16 family [23, 102], called commercially WiMAX. The range is from a few hundreds meters up to several miles. Transmission speed is up to 70Mbps at the air interface.

- **WLAN** - Wireless Local Area Network [69]. This is the family of 802.11 networks with a range from 20 meters indoor up to 150 meters outdoor and with transmission throughput of up to 54 Mbps at the air interface.

- **WPAN** - Wireless Personal Area Network . The family of 802.15 networks belong to this group. The most popular one is Bluetooth [94], with throughput of up to 1 Mbps at the air interface and a range from 10 cm up to 100 meters.

2.3.2 Mobile data networks

In contrast to wireless 802 networks, mobile data networks have their origin in circuit switched mobile networks. This legacy brings much more sophisticated transceiver construction. This is because the coverage area is very large in comparison to WLAN or WPAN service area. Additionally, mobile data networks have to service a great number of users that are moving, which is not the case in current WMAN networks. All these expectations inevitably result in the use of much more advanced technology than in the wireless 802 network family. Hence, mobile data networks will be used for evaluation of the general model of a transport protocol stack used to transport IP packets through a wireless environment.

The following is a quick introduction to the most successful mobile data networks.

- **GPRS** General Packet Radio Services [20] is an evolutionary step for introducing a data-oriented air interface and backbone network into the GSM circuit switched infrastructure. The theoretical throughput is up to 114 kbps at the air interface.

- **EGPRS** Enhanced General Packet Radio Service [10] is an improvement on the GPRS system. The air link is capable of delivering throughput of around 300 kbps. Additional improvements have been made to the core network making the entire system performing much better that the GPRS. Both GPRS and EGPRS are considered as 2.5G systems.
- **UMTS** Universal Mobile Telecommunications System [76, 127] is a system designed to carry a mix of voice and data traffic. It uses CDMA technology to share access to the radio resources. Early versions offer throughput of up to 2 Mbps at the air interface, whereas the new modifications allows downlink transmission speed of up to 14 Mbps.
- **CDMA2000** [31, 104] is a system developed and popularised in USA as a competitor to UMTS. Both UMTS and CDMA2000 belong to the 3G system family.

The GPRS transmission plane protocol stack is examined below, as the UMTS and EGPRS stacks are based on the GPRS stack design.

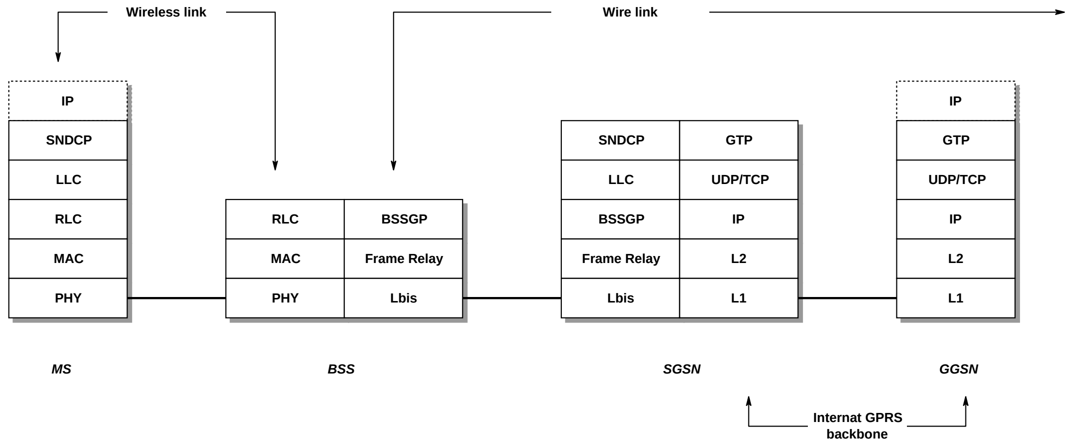


Figure 2.4: *Transmission plane protocol stack of (E)GPRS networks*

An IP packet entering the GPRS network through the Gateway GPRS Support Node (GGSN) is redirected to the Serving GPRS Support Node (SGSN) that services the destination mobile node as shown in figure 2.4. This SGSN stores all IP packets belonging to this connection and creates a Logical Link Control (LLC) frame that contains the first IP packet in the queue associated with this particular connection [4]. This frame is then sent trough a Frame Relay link to the Base Station Subsystem (BSS), where it is fragmented into a number of radio blocks at the

[4] If this packet is too large, it is fragmented into a number of LLC frames. However, in most cases one LLC frame is sufficient to send one IP packet.

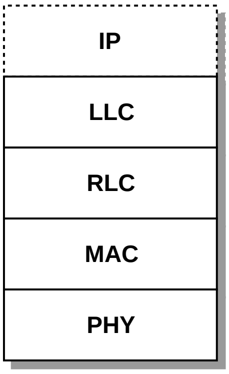


Figure 2.5: *Generic transmission plane protocol stack in mobile data networks*

Radio Link Control (RLC) layer. The number of radio blocks being used for the transport of a particular LLC frame depends on the radio channel condition. If the signal reception is good , then each radio block requires less code protection, therefore, fewer of radio blocks are needed to send this LLC frame. The RLC also performs retransmissions of unsuccessfully transmitted radio blocks to improve reliability of the radio block transmission. The role of the Medium Access Control (MAC) layer is to assign an appropriate radio channel or radio channels for a particular transmission. In the case of GPRS, this is a set of time slots on a certain carrier frequency. Following that, these radio blocks are sent to the mobile node through a wireless channel at the Physical (PHY) layer. Each bit of a radio block is mapped into a symbol of the modulation constellation and transmitted to the receiver.

From the wireless transmission point of view the influence of the Sub Network Dependent Convergence Protocol (SNDCP) layer can be omitted, as it is mainly responsible for sending an encapsulated IP packet into the relevant Base Station Subsystem (BSS). Thus, the most important layers are LLC, RLC, MAC and PHY. The generic version of the transmission plane protocol stack in mobile data networks built upon these layers will be presented in the next sub-section. The main focus is placed on the detail analysis of their functionality and the effects they can have on the delay characteristic of IP packets.

2.4 Generic transmission protocol stack in mobile data networks

The generic transport protocol stack in mobile data networks is shown in figure 2.5. It is based on the GPRS protocol stack, which has been inherited by EGPRS and UMTS. It consists of the following layers: LLC, RLC, MAC and PHY.

2.4.1 Logical Link Control layer

This layer controls the release of packets to and from BSS. Furthermore, it maintains a connection as a user moves from cell to cell. Additionally, the LLC layer is responsible for maintaining a ciphered data link between a Mobile Station (MS) and a SGSN. This layer introduces a small overhead and static delay to the data transmission[5].

2.4.2 Radio Link Control layer

The RLC's role is mainly to deliver the best possible radio channel performance to the layer above. This is achieved by choosing the most appropriate Forward Error Correction (FEC) level for radio block payloads. Backwards Error Correction (BEC) is also used to improve reliability. A few modes of operation, with different transmission delay and loss characteristics are available to be used according to requirements of higher layers.

- **Transparent mode** - in this mode the LLC frame is split into a number of radio blocks. A Cyclic Redundancy Code (CRC) is added to each radio block as a method of checking for errors in transmitted data. This mode offers the most effective use of radio block payload space since the lack of FEC gives all of the radio block payload to LLC frame data.

- **FEC mode** - in this mode, the RLC layer monitors the level of Block Error Rate (BLER), Bit Error Rate (BER), and Channel to Interference ratio (C/I). In addition, the CRC introduces an error correction code. Thus, the payload of a single RLC is occupied not only by bits from LLC frame but also by FEC bits. Therefore, the number of radio blocks needed to transmit the same LLC frame is larger than in the Transparent mode. As this method greatly improves the loss performance of radio blocks, it is used in the case of radio channels with poor signal reception.

- **BEC mode** - in this mode higher reliability is assured by retransmitting wrongly received radio blocks. This process is called Backward Error Correction or Automatic Repeat-reQuest (ARQ) . Each radio block has a CRC that allows the receiver to distinguish errors in the received radio blocks. In the case of an unsuccessful reception, the receiver sends a report back to the transmitter and the transmission is repeated. There are three major categories for how this retransmission happens.

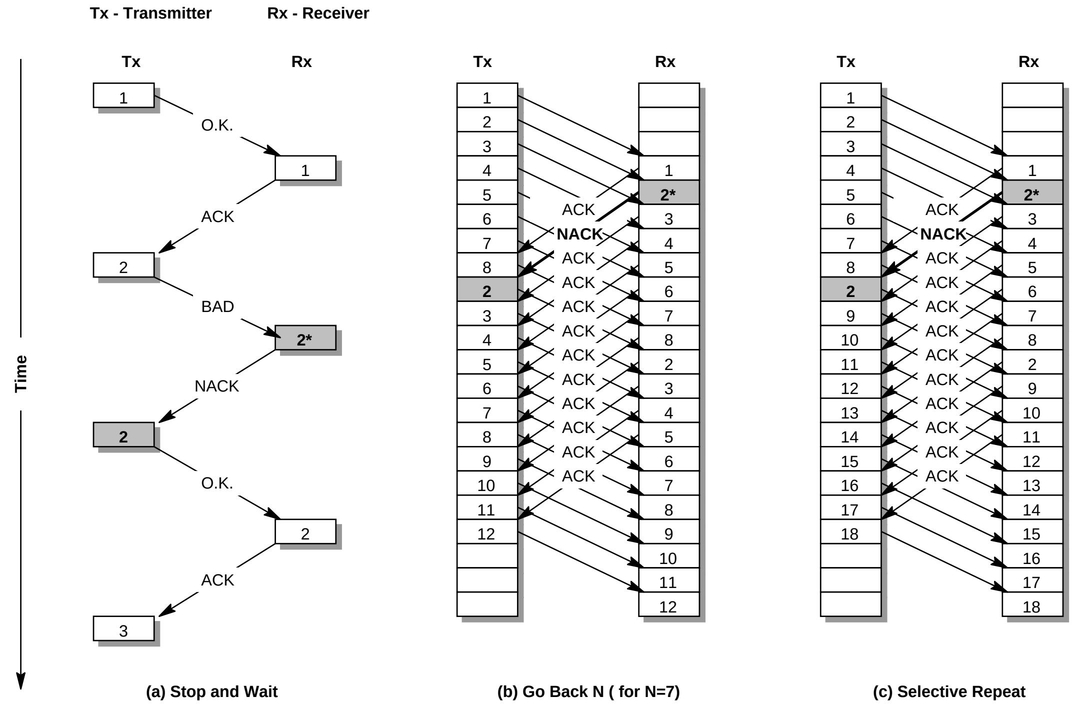
[5] This delay remains static if the retransmission of the LLC frames is turned off, which is assumed in the rest of this thesis.

1. *SW* - Stop and Wait ARQ protocol is the simplest. The transmitter sends a radio block and waits for the message with an Acknowledgment (ACK), or Negative Acknowledgment (NACK), from the receiver. Then the next transmission, or retransmission in the case of previously errored transmission is performed, figure 2.6(a) . Due to long idle periods it is the least efficient BEC method in terms of transmission throughput and delay performance. However, it is the easiest to be implemented from the technical point of view, as no buffering at the transmitter nor at the receiver side is required.

2. *GBN* - Go Back N protocol is more advanced than SW. It sends a stream of radio blocks. Each of them is acknowledged by the receiver if successfully received, or in the case of errored transmission a NACK is sent back to the transmitter. If a NACK is received at the transmitter, the transmission is repeated again for the radio block that is marked with a NACK and all radio blocks following it, figure 2.6(b). This streaming fashion of the transmission introduces higher throughput, as radio resources are used all the time. However, due to a lack of buffering at the receiver some radio blocks have to be retransmitted even though they were correctly received the first time.

3. *SR* - Selective Repeat protocol is the most sophisticated among all pure BEC approaches. It sends a stream of radio blocks to the receiver. Each radio block is then checked for its correct reception. If it was correctly received the receiver sends an ACK message to the transmitter. Otherwise, a NACK is sent and this radio block is retransmitted. This approach requires buffering at both sides of transmission. It is the most throughput effective ARQ scheme, as only errored radio blocks are retransmitted, figure 2.6(c). In modern mobile data networks the ACK/-NACK messages are not transmit instantaneously, as in pure SR-ARQ. The up-link is usually much more limited in terms of transmission power than the down-link, since the up-link's power comes from the mobile node's battery. Hence, it would not be appropriate to send each ACK/NACK. On the contrary, a report is generated once every N_{Poll} number of radio blocks and this message contains ACK/NACKs for the previously received N_{Poll} radio blocks.

- **BEC + FEC mode** - A combination of BEC and FEC is the most commonly used scenario in modern mobile data networks. It is possible to minimise the number of radio blocks that will be errored by choosing the correct level of FEC. For those radio blocks that did not have sufficient FEC support, the BEC scheme brings another chance for these to be transmitted over the



(a) Stop and Wait

(b) Go Back N (for N=7)

(c) Selective Repeat

Figure 2.6: *S&W, GBN and SR packet traffic diagrams*

radio channel. Three main combinations of FEC and BEC are currently in use.

1. *Hybrid Type I ARQ* - represents the most straightforward combination of FEC and BEC. For a given radio channel reception quality an FEC level is chosen. The aim is to minimize the number of unsuccessfully received radio blocks using the least possible amount of space for this code protection. If an error occurs during transmission then the ARQ loop introduces the retransmission of these radio blocks that did not successfully reach the receiver.
2. *Hybrid Type II ARQ* - this technique is more advanced than Hybrid Type I ARQ. It stores all errored radio blocks at the receiver and processes them together with the data being retransmitted to increase the probability of correct decoding.
3. *Hybrid Type III ARQ* - Hybrid Type III differs from Type II by using codes that allow the independent decoding of each retransmission attempt and if this does not give a positive result it will then use all versions of previous attempts to decode the retransmitted radio block.

Due to the falling cost of hardware and increasing CPU power accessible at mobile nodes, the Hybrid Type I/II/III ARQ techniques are more and more popular in modern mobile data networks. GPRS uses Hybrid Type I ARQ, but EGPRS uses all modes. Thus, these hybrid techniques merit close examination with respect to their influence on the transmission performance. This thesis focusses on the delay component introduced by these hybrid techniques to the total IP packet transmission delay. The analysis of the nature of this delay component will lead to two methods of predicting the average value of this component in mobile data networks.

2.4.3 Medium Access Control layer

The main role of the Medium Access Control protocol is to manage access to the communication resources. In the case of wireless networks this is the radio channel. There are a number of MAC protocols, each suited to a certain architecture and type of data. A short classification of these algorithms is presented.

First of all, MAC protocols can be classified into two major categories, centralized and distributed, in regards to the network architecture they operate in. Further classification can be made based on the mode of operation they are working in. These modes are: random, guaranteed and hybrid access protocols, as shown in figure 2.7.

- **Random access protocol** - nodes contend for access to the resources. When only one node makes a transmission attempt, then due to lack of interference from other nodes, the transmission is successful. If multiple stations attempt transmission simultaneously the collision results in unsuccessful transmission. The most known algorithm is ALOHA. On average it offers 18% [95] of the maximum throughput offered at the PHY layer. A modified version introducing syncronisation in form of slots where data can be send is called Slotted ALOHA, S-ALOHA, and it offers twice as much throughput as ALOHA.

- **Guaranteed access protocols** - nodes access the medium in an orderly manner, usually using a round robin algorithm. There are two ways of implementing such an approach. One is by a master-slave configuration, where the Master controls all transmission by polling all nodes, collecting transmission requests, and assigning a portion of the radio resources to each requesting node according to the transmission rules of the collective transmission. The other way of operating is to offer access to the radio resources by a token that is passed from station to station. The only node that can transmit is that one which is possessing the token. The second method is used less often, due to the nature of radio communication. It is quite easy to lose the token , due to high error rate experienced in a radio transmission, and the token recovery process is quite difficult to perform, especially if a great number of users share it, which is the case in mobile data systems.

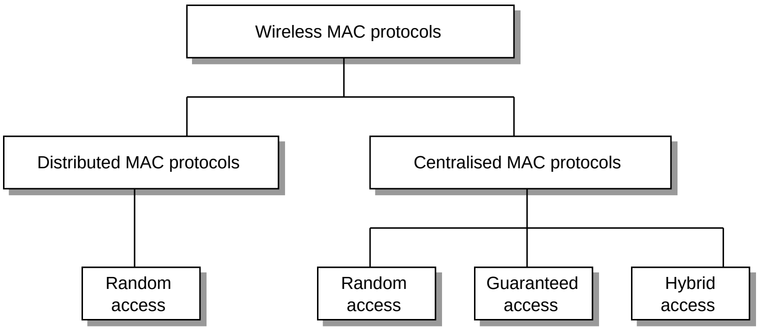


Figure 2.7: *Classification of wireless MAC protocols*

- **Hybrid access protocols** - use the best aspects of the two previously described policies. Most of them are based on a request - grant policy. Each node sends a request to the base station using a random access protocol, specifying the amount of bandwidth or access time required to transport its data. The base station reserves the requested resources, if it is possible, and informs each particular node about the format to be used for transmission and the time when access is granted.

Random access protocols are used mainly in wireless networks with a distributed architecture, like the Ad-Hoc mode available with 802.11 type of networks. On the contrary, in mobile data networks the guaranteed and hybrid protocols are mainly in use. This is because mobile data networks have a highly centralized structure and the synchronization of these networks is much superior to the distributed approach.

When analysing centralized networks, it is very important to distinguish the expected performance of the up-link and the down-link. The base station (BS), which is an essential point in centralized networks, is usually connected to the power grid, has advanced antenna systems and access to powerful computation resources. Thus, the signal sent from such a base station can be strong and can exploit the use of directional antennas. This is not the case at the mobile station, that usually is powered by a battery. This limits the accessible energy, computation power and antenna size which can cause significant reduction in the up-link signal power, antenna gain and prolonged use of sophisticated transmission techniques. This asymmetry and the character of the data traffic, which usually is asymmetric and favours the BS to MS direction, are the main reasons why the down-link transmission dominates in mobile data systems. Nonetheless, once the voice transmission is being sent, the up-link performance becomes an important issue. However, due to a very low throughput expectations associated with voice transmission it is not very hard to assure sufficient up-link performance.

2.4.4 Physical layer

The main role of the Physical layer is to transmit the bits that constitute the data segments from higher layers. In wireless networks that transmission is performed ”through the air”, which brings challenges to the whole process of data transfer.

First of all, a radio block is a set of bits that are mapped into a relevant modulation symbol. One symbol may carry one or more bits, depending on the modulation techniques used by a particular system or system mode. The symbols are then transmitted using radio waves.

The radio waves experience the following phenomenons: path loss, reflection, scattering and diffusion. Due to the non-uniform distribution of obstacles, the path loss does not have the same value in each particular direction. This is called the shadowing phenomenon. Reflections, scattering and diffractions cause the multipath reception of the transmitted signal. Multipath propagation causes large fluctuations in the magnitude of received signal, which in extreme situations may result in 30 dB, rapid attenuation of the signal's amplitude.

There are two phenomena that describe the multipath reception degradation: Doppler spread and Delay spread.

- **Doppler spread** is an effect caused by the mobility of one of the following elements: the transmitter, the receiver, or the surrounding environment. Mobility creates a situation where a mobile node crosses local maxima and minima of radio wave energy distribution. Thus, the magnitude of the received radio wave is significantly altered by this rapid change of energy distribution. The effect of rapid changes in magnitude of received signal is called fading.

  In scenarios where Non Line of Sight (NLOS) transmission occurs and either the transmitter or the receiver or both are moving, then the fading may reach up to 30 dB [60]. If both, the transmitter and the receiver, have a fixed position and Line of Sight or Obstructed Line of Sight (LOS and OLOS respectively) is assumed, then the attenuation has smaller jumps of up to 20 dB. Additionally, these fades are much less frequent due to the fact that the receiver and transmitter do not move and only the environment changes, figure 2.8.

  The frequency of amplitude jumps is proportional to the speed of the receiver (or transmitter) and inversely proportional to the central wavelength of the transmitted signal. Simply speaking, if the wavelength is short and velocity is high, then in a unit of time the receiver will frequently pass through the local distribution of radio wave energy maxima and minima. The mathematical definition of Doppler spread is as follows [139]:

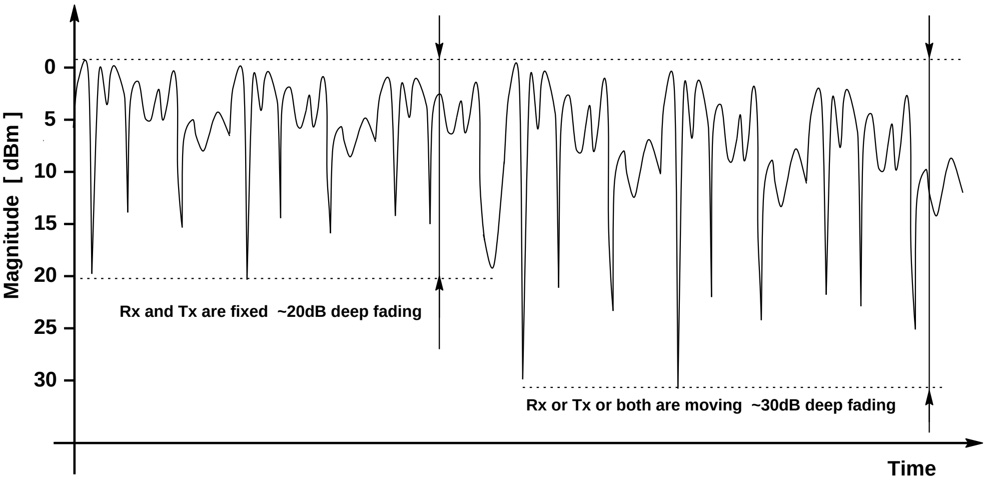


Figure 2.8: *Fading for fixed and mobile Rx and Tx*

$$f_D = \frac{\upsilon}{\lambda} \cdot cos(\theta) \tag{2.1}$$

Where:

- υ represents the relative velocity between the Transmitter (Tx) and Receiver (Rx).
- λ represents the wavelength of the central frequency of the channel used during the transmission.
- θ represents the angle between the vector of velocity and the line of sight between the receiver and transmitter. For simplicity we assume that θ is equal to zero, so that $cos(\theta)$ equals 1.

Constant or semi-constant channel path loss during the demodulation process of a single symbol improves detection accuracy. Thus, it is desirable to have a situation where the symbol transmission time T_s is short enough to experience nearly constant attenuation. Therefore, the fading caused by Doppler shift has minimum impact on this symbol. If the symbol transmission time is long, then this symbol experiences higher attenuation fluctuations that result in problems with correct detection of the transmitted symbol. The maximum symbol duration time, when the attenuation has nearly flat characteristic, is called coherence time T_c, and is defined in equation 2.2 [95].

$$T_c = \frac{0.4}{f_D} \tag{2.2}$$

If $T_s < T_c$, then the Doppler spread effect has a negligible influence on the transmitted symbols.

- **Delay spread** on the other hand creates the problem of Inter Symbol Interference (ISI) and is caused by multipath propagation. The time between the shortest path reception and the last significant path reception describes the environment in terms of multipath [54]. The bigger the coverage of a single transmitter, the bigger the distances to potential obstacles from which the radio wave can be reflected. This results in larger temporal separation between significant signal reflections. There are several ways of describing this effect. Two of the most popular are: maximum excess delay and root mean square (rms) delay spread σ_τ. Maximum excess delay is the time between the first arrival and the last arrival that comes above a predefined threshold, which is usually 10dB weaker that the strongest one [95]. The root mean square delay spread takes into account the strength of particular receptions and give some insight into how much the strongest receptions are placed apart each other. More details about σ_τ calculations can be found in [95].

  In general terms, multipath reception degrades radio channel throughput. This is because transmission of a new symbol should not be conducted before all reflections of the previous one have reached the receiver. A speed of the transmission may be chosen that results in small ISI. This defines the bandwidth of a transmission channel and is called coherence bandwidth, B_C. The formula for computation of the coherence bandwidth is [95]:

  $$B_c = \frac{x}{\sigma_\tau} \tag{2.3}$$

  where the x may vary from 0.02 to 0.2 depending on the desired level of the ISI protection and used modulation requirements.

  If the bandwidth of a single channel is smaller than the coherence bandwidth, then such a channel is called a flat fading channel. For these channels, the ISI has a marginal effect of the transmission performance. On the other hand, if the bandwidth of a single channel is larger than the coherence bandwidth, then such a channel is called frequency selective fading and the transmitted waveform may be significantly affected by the ISI.

  The effect of ISI on transmission in different radio environment is shown in table 2.1.

Doppler and Delay spread may introduce significant reduction of the available bandwidth. Hence, to overcome these limitation more advanced transmission techniques have been proposed and used in modern wireless data systems.

- **Fade margin** - This techniques minimizes the fading effect by assuring that the average power of received signal is larger than would be necessary if

Type of	rms delay spread	Maximum bit rate for BPSK
environment	ns	kbps
Mobile (rural)	25000	4
Mobile (city)	2500	40
Mobile (micro cell)	500	200
Large building	100	1000
Office building	40	2500
Single room office	25	4000

Table 2.1: *ISI interference in different radio environments, [29] and [54]*

there was no significant fading effect.

- **Diversity** - these techniques exploit the small correlation between two reflected signals in the time, space or frequency domain. If the distance in time, space or frequency is large enough the received signal experiences different fading characteristics. Thus, post-reception processing of the signal may greatly improve the reliability of the link. Placing two antennas more than a half wavelength apart is an example of the space diversity technique. Such a distance is usually sufficient to perceive reflections of the radio wave as quasi independent. Thus, it is not likely that both antennas simultaneously experience deep fading, and by selecting the antenna with the strongest signal the overall reception is improved.

- **Coding and Interleaving** - Coding works on the basis of the mathematical theory of finite fields. By attaching extra bits it is possible to improve the reliability of the transmission. The codes used nowadays are sensitive to errors appearing one after the other, which is usually a case in radio transmission. Thus interleaving techniques are used, to spread these errors. Depth of interleaving depends on the burstiness of the channel, the bigger the burst of errored bits, the deeper interleaving is required.

- **Adaptive techniques** - adaptive techniques try to predict the channel condition. Based on this prediction the transmission power level or modulation is chosen in the way that aims to achieve the best performance in terms of throughput, delay and loss. They operate on longer time periods than the Diversity techniques, as changing such parameters as modulation, coding and power level introduce implications for the whole transmission process.

- **Equalization** - this set of techniques is based on the idea of flatten the channel response by signal processing. A training signal is sent periodically to learn about the channel response, and based on its response, the appropriate action is taken to cancel or reduce the ISI influence or to flatten the channel response. Unfortunately, the complexity of such a solution grows significantly with the number of reflections.

- **Multi-carrier** - The transmission bandwidth is divided into many narrowband sub-channels transmitted in parallel. An information signal of bandwidth B is divided by N, where N is a number of sub-channels. Thus, a single sub-channel has a bandwidth equal to $\frac{B}{N}$, which minimizes ISI problem for a sub-channel transmission. One of the most efficient techniques is Orthogonal Frequency Division Multiplexing (OFDM).

- **Spread Spectrum** - spread spectrum transmission is characterized by very short transmission time for a single symbol. However, the self correlation properties of the Direct Sequence Spread Spectrum (DSSS) signal makes all delayed signals marginally correlated to the main path signal. Thus, they are cancelled during the de-spreading process. Some advanced DSSS receivers, like RAKE [51], can combine a few versions of the received sequence to gain a very good throughput performance.

- **Antenna Solutions** - by careful beam shaping it is possible to minimize the multi-path effect on the reception quality. Simply speaking, if proportionally more power is delivered by the LOS path in relation to NLOS and OLOS paths, then the reflected or highly attenuated waves play a less significant role and the reception condition are better.

As can be seen, there is a set of techniques that minimize the Doppler and Delay spread effects, thus improving the reliability of the radio link. Use of a particular solution, or a combination of a few, results in a certain BER and consequently BLER characteristics. Therefore, it is very hard to model the physical layer influence on BER and BLER considering all factors. One of solution that may overcome this problem is to analyse the PHY layer influence by simply measuring BLER characteristics. This solution is pursued in Chapter 3.

2.5 QoS issues in mobile data networks

The level of QoS that can be delivered by wireless data networks is an increasingly important issue as these networks become one of the technologies used by Internet access networks.

The faster and more predictable these networks deliver IP packets to the destination node the better will be overall end users' perception of delivered services. The speed of such networks is highly dependent on the mobile node's current location and direction of movement in respect to the base station. Hence, an improvement in the transmission throughput and delay characteristic may not always be possible. However, information about the expected IP packet delay and loss characteristics may help to improve the performance of higher layers or may be passed up to the application itself, which may take proactive steps to shape its traffic in such a way that will sustain an acceptable quality with lower network resource demands. For example it may cease the transmission or change voice and/or video codecs to produce a lower quality but lower bandwidth demanding stream.

To investigate the use of QoS in mobile data networks, consider the general transmission plane protocol stack introduced earlier. The LLC layer represents a static influence if its retransmission properties are switched off. The RLC layer uses Hybrid Type I/II/III ARQ techniques for improving throughput efficiency and loss rate at the radio block level. However, these ARQ techniques introduce a significant dynamic delay component to the whole IP packet transmission process. The MAC layer also has significant and dynamic influence on the transmission process, since it can dynamically adjust the amount of radio resources allocated to a particular connection. The PHY layer influence can equally alter the packet transmission performance. However, this usually cannot be controlled by the network, as it depends substantially on the location and velocity of the terminal.

The RLC and MAC play a crucial role in adjusting wireless systems to the local situation of radio wave propagation. Both have dynamic mechanisms that greatly affect the transmission performance. The use of MAC scheduling algorithms to optimize the network utilization, fairness and users' QoS contract delivery problems has been studied in great detail. The influence of RLC and Hybrid Type I/II/III ARQ loop in particular on the total delay of IP packet has not been studied as intensively and is the major concern of this thesis. This work does not focus on improving any existing Hybrid techniques, but aims to analyse and find a method of predicting the delay of an IP packet transmitted in a previously described scenario.

To achieve this a few fundamental assumptions have been made.

- First of all, the MAC access policy has to be very static to capture RLC dynamic behaviour. Thus the guaranteed access protocol is assumed as the scheduling policy. This results in a situation that MAC offers an access to the portion of radio resources on regular basis, every *"x"* time units.
- Since the transmission in mobile data networks is usually down-link-oriented, this direction in studied.
- The influence of LLC layer is omitted. It is assumed that IP packet fits into one LLC frame and that the retransmission feature at LLC layer is not active.
- In regards to RLC there are two assumptions:

  - First, the FEC in the feedback is strong enough and that there are no errors in delivering reports containing ACK/NACKs.
  - The second is that after three transmission attempts each radio block will be successfully decoded at the receiver. This limit comes from the fact that in existing EGPRS systems, that use Hybrid Type II/III ARQ, there are maximum three transmission attempts of a particular radio block. After that the process of transmission of the LLC Frame, consisting the triple unsuccessfully transmitted radio block, starts again from the beginning. As it is shown in Chapter 6 the loss ratio for most common radio channel scenarios is of the order of magnitude lower than 10^{-4}, when Hybrid type II/III ARQ is in use. This low error rate and the fact that the main focus of this work is put on the delay aspect of IP packet transmission determined that it is assumed that triple transmission attempt offers negligible error rate.

- The PHY layer influence can be analysed from two perspectives, based on bits or radio blocks. The analysis based on bit level requires a detailed knowledge about radio wave propagation characteristics, chosen modulation, FEC etc. Additionally, simulation at such a level consumes great a deal of time. The solution based on radio blocks bypasses all previously described issues and goes straight into the error process of radio blocks. Furthermore, it reduces simulation complexity and simulation time. For these reasons the PHY level is analysed at the block level in this thesis.

2.6 Summary

The perceived quality of every service that communicates between end points through a network connection depends greatly on the performance of the networks. The network that currently dominates is the Internet, which in reality is an interconnection of different networks. Hence, its protocol stack, the TCP/IP suite, is used in the majority of data oriented connections.

One of the most important developments in recent years is the use of wireless networks for Internet access. Examples include WLAN in offices or UMTS in outdoor scenarios. This introduces challenges in the form of link quality fluctuations that cause problems to protocols above, like TCP or to the Application itself. Therefore, if it is possible to predict the delay performance of IP packets in the Internet access part, it would help to tune TCP or adjust the data rate and characteristic generated by the application -thereby improving the QoS.

In mobile data networks there are two layers that greatly influence the overall IP packet delay and loss characteristic, RLC and MAC. The RLC influence on the IP packet delay characteristic has not been analysed as deeply as the MAC. Therefore, this is the topic for further investigation in this thesis. In particular, the effect of Hybrid ARQ protocols on the delay of an IP packet is examined.

CHAPTER 3

Related work

This chapter presents an overview of methods for improving the delay performance of the most influential protocol layers: named TCP, MAC, PHY and RLC.

First TCP improvements are investigated with an emphasis put on the interaction between TCP congestion and flow control mechanisms with link layer protocols and potential benefits these mechanisms can gain from cross layer communication. Second an overview of the well wireless MAC scheduling policies and their effect on IP packet delay performance is given. Following that, the issue of PHY layer influence on the RLC radio block error process is investigated. Different models of emulating the behaviour of a real PHY layer are discussed regarding the possible use of one of these models in further research. Finally, the RLC layer is investigated with main emphasis put on the models discussed in the literature regarding the ARQ loop delay and throughput performance analysis.

3.1 Introduction

As was shown in the previous chapter, the total end-to-end service quality perception depends greatly on the delay characteristic of the connection. This characteristic is altered mainly by TCP's flow and congestion control mechanisms,since TCP occupies most of the nowadays Internet backbone traffic[1].

The delay and loss performance of TCP is very sensitive to the delay and loss characteristic of the connection path being used. Since the most vulnerable part of this path is the Internet access network, it becomes the dominant source of limitations that affect the TCP performance. The magnitude of these limitations is much higher when a wireless technology is used for Internet access, which is

[1] The TCP occupied around 80% of all Internet backbone traffic in late ninetees [70] and year by year it is loosing few percent (see `http://www.cs.columbia.edu/~hgs/internet/traffic.html`). Nonetheless, it will keep the dominant position for a while.

becoming a more and more popular option. Therefore, the loss and delay performance of wireless networks are potentially substantial factors influencing the TCP performance.

The wireless networks can be classified roughly into three groups: wireless 802 family of networks, mobile data networks and satellite networks. Of particular interest are mobile data networks as they use the most sophisticated mechanisms to sustain a good connection quality. Hence, they have been chosen as a model for the investigation of the mechanisms that greatly alter the delay and loss performance of TCP connections.

The previous chapter exposes that the layers affecting mostly the delay characteristic of IP packet, carrying TCP segment, being transported over mobile data systems are: MAC, RLC and PHY layers. Therefore, they are closely analysed in the following sections.

The TCP part focuses mainly on the problems of achieving a good quality of TCP connections in paths consisting of wireless links. Following that, the main approaches to minimize the negative effect of wireless links on the TCP transmission performance are presented. Additionally, an example of a situation when the knowledge about an expected IP packet delay performance can improve TCP delay performance is shown. At the next stage, RLC layer is analysed, where a general overview of its capability and the research history of BEC is shown. The main focus is put on the analysis of Hybrid Type I/II/III ARQ techniques in relation to the characteristic of IP packet delay. After that, the MAC layer is investigated, with an emphasis on its resource access scheduling algorithms. The main focus is placed on their impact on the fairness and transmission delay properties. Finally, the PHY layer is analysed, since its influence on the IP packet delay characteristic is significant. A good radio reception can give a perception of a quasi ”error free” environment, while a poor one can reduce the effective transmission rate to nearly zero.

3.2 TCP acpects

3.2.1 TCP in wired enviroment

The wired environment is characterised by a very low link error rate[2], so that TCP segment errors are usually caused by overloading of intermediate routers. Thus, the congestion control mechanism, implemented in TCP, assures that in the event of segment errors, all stations will decrease the rate of their transmissions [66]. This decrease results in reducing the load on saturated routes, so that, the TCP segment loss ratio falls.

[2] E.g. in Gigabit Ethernet technology the BER is kept below 10^{-10} [137].

The details of implementation of such the policy can be found in the specification of TCP-Tahoe and its most widely used improvement TCP-Reno [124].

3.2.2 TCP in the wireless environment

TCP-Reno performs very well in a wired environment. However, in a wireless environment there are phenomena that were not taken into account by researchers during the design of the TCP.

Due to the transmission properties of a radio channel, a wireless link is characterised by much lower accessible bandwidth than a wired one. Additionally, the performance in terms of throughput, delay and loss fluctuate greatly during the connection period. Thus, the assumption about the low link error rate is not valid any more. Hence, the performance of standard TCP-Reno is significantly degraded by the presence of these phenomena [81, 83, 133].

This high and unstable error rate introduces errors to TCP segments which are misinterpreted as being caused by too heavy traffic on intermediate routers. As a consequence, the procedure of drastic load reduction, intending to neutralise the cause of the losses, is triggered. Nevertheless, the action of lowering the transmission rate will not decrease the error rate, because the reason behind those losses is not congestion-based but rather of link level origin.

Additionally, fluctuations in delay characteristics, as the result of a short-term poor physical link status, create a situation where the timers responsible for deciding when a particular link is lost become confused. Sudden degradation of TCP segment performance can be misinterpreted as the link outage. In a wired environment such a sudden delay degradation would be of relatively long duration, while in wireless environment it is quite dynamic. Additionally, the long delay itself slows down the feedback about channel performance, and the message that the link is operating again on its full capability is transferred slowly to the transmitter. Thus, TCP operating within such a wireless scenario has to take into account the effects of possible long transmission delay [81].

The above phenomena significantly degrades TCP delay performance [5, 22, 56, 135]. Therefore, the research community was and is working extensively to address these problems. A number of proposals to modify TCP or to replace it by new, more wireless friendly protocols have been proposed.

There are two ways to address the problem of spurious errors negative influence on TCP throughput and delay performance. The first way aims to solve this problem locally by using local Link Layer techniques to improve the link reliability. However, following this approach quite often results in degradation of average link throughput performance. In other words, it can be said that the errors are traded for transmission throughput efficiency. The second way is to solve the problem of sudden link layer errors at the transport layer. Making the TCP aware of the origin of particular errors lets TCP congestion and flow control policies react in a more appropriate manner. TCP can be informed directly by the wireless link side of the connection that some errors had happened due to a poor physical link condition. The same message can be announced by intermediate routers in an indirect manner. If there is an error and those routers do not signal congestion-based drops of packets at their nodes, it means that the origin of the errors has to be at the link layer.

3.2.3 TCP improvements for wireless environment

A substantial number of the approaches addressing the problem of TCP performance improvement for transmission in a wireless environment have been proposed. A comprehensive introduction to these methods with some experimental comparison is presented in [58, 84]. A short overview presenting the major ways of addressing these issues will be shown. These improvements can be classified roughly into three major categories: End to End, Split Connection and Link Layer schemes.

3.2.3.1 End To End schemes

This class of TCP improvement keeps the standard protocol layer intact. This means that the flow and congestion control is decided at the Transport Layer. Additionally, there is a direct communication between the transmitter and the receiver.

End to End schemes can be classified into two subgroups. The first one relies only on the information gained from its own observations and on messages from the other end of the TCP connection. The second approach uses information from other layers or from external infrastructure, like intermediate routers.

The most well-known solutions belonging to the first subgroup are as follows.

- **TCP-New Reno** [41] is a modification of TCP-Reno . It improves the fast recovery process and keeps this state even if a partial ACK is received. Partial ACKs are defined as those confirming part but not all of the segments that were outstanding at the start of the fast recovery period. The retransmission of the partial ACKs allows the TCP sender to recover from the losses, and thus avoid retransmission time-out [24].

- **TCP-SACK** [35, 85] is a TCP modification that deploys the Selective Acknowledgement mechanism to the standard TCP protocol stack. This improves the recovery from multiple packet losses within a single transmission window.

- **TCP-SMART** [33] operates similarly to TCP-SACK. This scheme trades off resilience to out of order delivery for a reduction in the overhead of the process of generating acknowledgements.

- **TCP-Vegas** tries to estimate the expected connection capacity by using minimum RTT [40]. This gives throughput improvement of between 37 and 71 % over the standard TCP-Reno [19, 68, 134]. However, TCP-Vegas has a substantial problem with fairness and with operating within a wireless, error prone enviroment [71]. One of the roots of this problem comes from giving up the AIMD mechanism described below for TCP-Veno and relying fully on the RTT based bandwidth estimation.

- **TCP-Veno** [42] works on similar basis as TCP-Vegas, except that it uses the Additive-Increase Multiplicative-Decrease (AIMD) mechanism. Basically, if the packet loss is detected when the connection is in the congestion state it is assumed to be caused by congestion and the AIMD mechanism degrades the transmission rate. In a case of the congestion free state, the detection of a lost packet is assumed to be of link error origin and the AIMD mechanism is not triggered. This scheme performs better than TCP-Reno . Nevertheless, it still has problems coping with high error rates likely to be present in wireless environment [101].

- **TCP-Westwood** [116] continuously estimates the rate of the connection by monitoring the intervals between incoming ACKs at the transmitter. When duplicate ACKs start to arrive at the transmitter this scheme uses the estimated bandwidth as a start point for the fast recovery process. Problems with correct bandwidth estimation, especially when high link layer packet loss is present, cause it to perform poorly in a number of scenarios [46].

- **TCP-Peach** [63] is designed primarily to improve the throughput using the specific case of satellite links which have a huge capacity and a quite significant bandwidth delay product. It slightly modifies TCP-Reno by replacing slow start and fast recovery with sudden start and rapid recovery mechanisms. In sudden start and rapid recovery the sender estimates the available bandwidth with only one RTT measurement. Based on this measurement it adjusts its flow control mechanisms.

- **JTCP** Jitter based TCP - uses measurements of jitter to adjust congestion and flow control policies. Results in [143] show that in tested scenarios it outperforms TCP-Reno, TCP-New Reno and TCP-Westwood.

The two most popular solutions belonging to the second subgroup are as follows.

- **ECN** Explicit Congestion Notification solutions [39] operate on the assumption that the ECN mechanism, implemented at intermediate routers, will inform the transmitter that there is an incipient loss due to congestion at one or more of those routers. An example of such a policy is implemented in **TCP-Jersey** [72], where ECN messages help to distinguish between losses caused by congestion and those caused by link error.

- **ELN** Explicit Loss Notification solutions work mainly on the basis of informing the sender that a particular segment is lost due to the packet loss being a result of link error [15, 114].

3.2.3.2 Split Connection schemes

Split Connection schemes aim to shield the wired part of the TCP connection from the wireless one. It can be achieved by introducing at the end of the wired path a proxy that talks independently to the wired and wireless side. Each of these mediums have their own characteristic and by introducing two independently working recovery mechanisms a gain in throughput and delay performance is achieved. A good example of such a solution is **I-TCP** [14], which achieves a significant improvement over standard TCP-Reno. Another example of a Split Connection schemes is **MTCP** [21], which differs from I-TCP only in the way the communication between the mobile node and the end of fixed line is achieved. Unlike I-TCP, which uses TCP on both sides of the proxy connections, MTCP uses a unique protocol for the communication between the proxy and relevant mobile node. This protocol is tuned to the specific characteristics of the wireless channels. Both solutions outperform standard TCP-Reno. Nevertheless, they share the same drawbacks of the lack of end to end TCP semantics, double processing of the same segment and the need for additional equipment with significant buffers in case of long path connections.

3.2.3.3 Link Layer schemes

The Link Layer schemes follow a different philosophy. They aim to hide losses caused by the radio link. There are two major ways of achieving this: Link Layer unaware and aware schemes.

- **Link Layer unaware** approach tries to improve the quality of the radio link regardless of the kind of protocol at the Transport Layer. Thus, it cannot exploit the unique features of a particular protocol, eg:TCP. However, even without taking into consideration the specifics of the transport layer protocol characteristic this method is capable of achieving a throughput improvement for TCP transmission [61]. It is based on deploying techniques like FEC and BEC to improve reliability of the radio link. Some additional techniques like modulation and power adaptation can also improve the reliability at the expense of the throughput performance. A good example representing such an approach is AIRMAIL [34].

- **Link Layer aware** solutions improve TCP performance by working together with the local agent of TCP. All retransmissions of the segments happen locally, instead of being resent by the transmitter. The main difference is that the Link Layer unaware solution would retransmit only a corrupted part of segment, and would not be able to suspend requests from the receiver to retransmit missing segments, as it is not aware of any logical structure above Link Layer. This artificial blockage of the retransmission request messages can minimise the problem of retransmitting the segment which is almost entirely transmitted at the Link Layer. The most prominent example of this category is the **Snoop-TCP** protocol [56, 57].

Link Layer unaware schemes are used very often in cellular data systems due to the large range of radio channel conditions experienced by their terminals. Since mobile radio networks were not designed to carry only IP traffic, they usually do not use Link Layer aware approach, which is very specific to TCP congestion and flow control mechanisms.

A practical example of using this link layer unaware scheme is EGPRS [55], where advanced Hybrid Type I and II ARQ are used to improve the radio link quality. Additionally, the EGPRS system is able to adjust its Modulation and Coding Scheme (MCS) according to the current radio propagation scenario [75, 97]. This adaptation aims to deliver the maximum achievable throughput by choosing a MCS constellation that will transport the maximum amount of data with the minimum retransmission rate.

Trading bandwidth for reliability not always is possible. Sometimes the radio condition is so poor that no FEC and BEC can help to sustain a reliable connection quality and some TCP segments are going to be considered lost. Thus, similarly to Split Connection schemes the Link Layer schemes do not fully assure operational loss shielding to the TCP connection. Additionally, the timing between the Link Layer adaptation schemes and TCP congestion and flow control mechanism is not synchronised. This often creates a degradation in performance of TCP [5].

3.2.4 Proposed improvements for TCP

ECN and ELN are TCP wireless-oriented improvements that use cross-layer communication and infrastructure based help. The information about the origin of TCP segment loss is used in these techniques to tune the TCP throughput performance. Similarly, if an estimation of average IP packet delay carrying TCP segment is known, then it can be used to tune the performance of TCP.

The IP packet delay characteristic relies greatly on the performance of the MAC, RLC and PHY layers. Hence, if the influence of the RLC (which is inevitably connected with the status of the PHY layer) is predictable, then it can be used as a lower-bound of the average IP packet delay, which can also be useful to TCP tuning mechanisms. The analysis and prediction of the combined RLC and PHY layers average delay component of the IP packet delay when ARQ is used is the aim of this thesis.

Due to the influence of these three layers on the delay characteristic of IP packet, they are going to be investigated further with respect to this influence.

3.3 MAC aspects

As was mentioned in the previous chapter the MAC layer plays an important role in the process of sending an IP packet through a wireless medium. It assigns a portion of radio resources and schedules the access to these resources among all users attempting to transport their data.

A MAC protocol in Mobile Data Networks specifies two important things. Firstly, it creates a framework based on which all communication between transmitter and receiver is conducted. Secondly, it also describes the data format used for the transmission of information. In the case of centralised networks the MAC uses scheduling policies to find a compromise between user expectations and utilisation of limited radio resources.

The first role of the MAC is to provide logical channels, which offer synchronisation of mobile nodes to the core network and exchange information about signal to interference ratio, number of unsuccessful transmission attempts etc. The details of logical channels implementation have a decisive influence on speed of feedback, up to date radio channel status information, and feasible granuality of carried data.

The second role of the MAC layer is to provide transport channels. These channels take control over the format of the data units within the wireless transmission part. This section of MAC is responsible for scheduling and multiplexing streams of data belonging to multiple users and connections. The scheduling policy that is implemented in a particular solution plays the dominant role in the impact of MAC layer on the transported IP packet delay characteristic.

Countless approaches have been proposed to find an algorithm that balances fairness, network utilization, low complexity, delay requirements, resilience to spurious errors and other relevant factors. Good overviews of the main scheduling algorithms designed for error free scenarios, have been presented in [53, 147, 149]. Some major scheduling algorithms for the wired, error-free environment will be presented here, with short descriptions of their major features.

Wired networks do not deal with the problem of location dependent throughput and presence of high level of transmission errors. These two phenomenon create transmission scenarios in which the previously mentioned algorithms do not work appropriately. Hence, some modifications to error free scheduling algorithms are needed. The new - modified solutions, designed to work in a wireless environment are presented in [38, 78]. Some of the most important classes of these solutions will be briefly reviewed in the following sections.

The implementation of any algorithm in a real system is always a challenge. This is because the assumptions taken during the modelling and/or simulation process have to be relaxed. Thus, some principal problems in implementing scheduling algorithms are discussed at the end of this section. Finally, some comments about the possible use of IP packet delay characteristic prediction are made.

3.3.1 Scheduling algorithms for wired (error free) enviroment

One of the main tasks to be fulfilled by scheduling algorithms is to meet certain transport characteristic criteria. These criteria can be network-centric or user-centric. In the case of network-oriented scheduling algorithms maximisation of resource utilisation is usually the primary aim. Some other required features are low complexity , robustness and real-time applicability and fairness. In the case of user-oriented approach, factors like throughput, maximum experienced delay and reliability are the main parameters of concern. Obviously, the perfect scheduling algorithm should meet all mentioned requirements, but such an algorithm has not yet been discovered.

Two the most sought after features in wired scheduling algorithms are delay boundaries and fairness. One of the simplest scheduling algorithm is First In First Out (FIFO). However, neither fairness nor delay boundaries features are achieved with this policy. The Round Robin (RR) policy improves the level of fairness and ensures some delay boundary. Nonetheless, due to the very simple nature of these algorithms their performance in terms of meeting the required delay limits is not very good. More advanced schemes have been proposed and the most distinct of them are going to be presented in this subsection.

The scheduling algorithms designed for operating in an error free environment can be classified into two groups: *work-conserving* and *non-work-conserving*. The primary difference is in the fact that the first group always utilise the available resource, while the second one may not.

The most common solutions working in *work-conserving* mode are as follows.

- **GPS** - Generalized Processor Sharing [99] is a scheduler that is characterised by a good level of fairness and flexibility. It uses a continuous model of load distribution, which means that load is not quantised into entities like frames or packets. Additionally, the GPS scheme assumes that each connection has its own buffer and is able to transmit data from this buffer simultaneously with other connections. These idealisations makes GPS impossible to deploy in a real world scenario. Nevertheless, it is used quite often as a reference point for other scheduling disciplines.

- **PGPS** packet-by-packet GPS [100] is also known as **WFQ** Weighted Fair Queuing [4]. This approach emulates the behaviour of a GPS scheduler in a discrete domain where the load is quantised. The WFQ algorithm selects the packet to be sent at a start of the transmission τ which would finish its transmission at the same time as in a corresponding GPS system. However, this solution does not always mimic the performance of GPS. Hence, a new scheme was proposed WF^2Q, that takes into consideration both start and finish times of packets in GPS to achieve better emulation of the GPS behaviour [16].

- **VC** - Virtual Clock [150]. This discipline emulates the behaviour of TDM-like systems. Each portion of data is allocated a virtual transmission time, which is assigned by the scheduler on the basis of declared throughput and current real time.

- **WRR** Weighted Round-Robin [6] aims to approximate the GPS behaviour. Thus, for every cycle it assigns a fraction of time to a particular connection. It differs from Round-Robin by allowing the assignment of different fraction of the access time to different connections by giving them different weighting factors. In this way it is able to differentiate the access to the communication resources. It does not introduce any practical delay boundaries, since the practical resource access time depends on the number of active connections.

- **SCFQ** Self-Clocked Fair Queuing [47, 48] is a simple alternative to PGPS which does not need to refer to the GPS model. Instead it computes the virtual time in reference to the virtual time of the packet that is actually being serviced.

- **DRR** Deficit Round-Robin [117] is a version of Round-Robin that is able to handle variable packet size.

- **CBFQ** Credit-Based Fair Queueing [18]- operates on the idea that each connection has an associated price as a result of some QoS contract. Based on this price value and the size of the Head of the Line (HOL) of packets from other connections this algorithm decides when to schedule access to a particular connection.

The most common solutions working in *non-work-conserving* mode are as follows.

- **HRR** Hierarchical Round-Robin [73] is a modification of Round Robin algorithm that allows implementation of different priority classes, where each class gets its portion of time slots that create one cycle.
- **SGQ** Stop-and-Go Queuing [49] is an scheduling policy that aims to smooth the shape of the traffic. It introduces the idea of fixed time frames when data can be transported. The carried packet can be forwarded toward the next hop only at the beginning of the next time slot.
- **Delay-EDD** Delay-Earliest-Due-Date is an extension of Earliest-Due-Date (EDD) [17]. In pure EDD each packet is associated with a deadline and packets are scheduled to be transmitted in order of increasing deadlines. The Delay-EDD extends the EDD service discipline by introducing a contract with each serviced source of data. If a particular source is maintaining the relation between its peak to average load rate, then the scheduling policy can keep promised delay boundary.
- **RCSP** Rate-Controlled Static Priority [148] aims to introduce a fair level of flexible allocation of both delay and bandwidth to a particular connection. Firstly, each connection has a regulator which shapes the traffic of this particular connection. Secondly, the traffic from one of these regulators is directed to one of the predefined traffic classes, implemented as simple FIFO queues. These classes have different average access to the radio resources.

The work-conserving approach offers higher bandwidth utilization than the non-work-conserving one. It results in lower average delay experienced by connections. Nevertheless, this solution has also some drawbacks. It is harder to assure that the delay boundary of a particular connection is kept regardless of the characteristic of incoming traffic, as the access to the resources is assigned in relation to the whole population of connections wanting to send data. Higher jitter than in comparison to non-work-conserving is also a side effect of work-conserving algorithms. Thus, the choice of algorithm from one of the presented classes is a choice which has to be tailored to the particular characteristics and demands of the expected traffic.

Other important factors that need to be considered are fairness and complexity. Table 3.1 shows clearly that there is no winning scheduling algorithm. Those which offer small delay boundaries, like WFQ, VTC and WF^2Q are usually quite complex. While, DRR having good complexity characteristic has lower performance in terms of fairness and delay.

	Delay Bound	Fairness	Complexity
PGPS (WFQ)	Small	Good	O(N)
SCFQ	Large	Moderate	O(log(N))
VC	Small	None	O(log(N))
DRR	Large	Poor	O(1)
WF^{2}Q	Small	Very Good	O(N)

Table 3.1: *Comparison of scheduler policies [38]*

3.3.2 Scheduling algorithms for wireless environment

The transmission of data in a wireless environment faces problems that are not present in the wired one. The two most distinctive features of wireless channels are time and location dependent throughput. Time dependency results in fluctuations of the accessible bandwidth, as the physical properties of the radio link change. Location dependency is determined by the attenuation of the radio channel. Thus, it is very difficult to assure a good level of fairness and at the same time to sustain a good delay performance per connection. A good overview of most the important wireless oriented scheduling policies can be found in [38, 78].

Due to link quality fluctuations it may happen that the transmission is suppressed for a short time, to avoid a high level of errors. Hence, it is common for scheduling policies working with wireless connections to have a *leading/lagging monitor*. Having that monitor allows the introduction of a *compensation module* that will make sure that the lagged connections have a higher priority than a leading one, if the lagged connection can transmit in a considered unit of time.

Although it is very hard to predict the state of a radio channel condition, it is preferable to have a unit that keeps up to date information about this radio channel status. Based on this unit it is possible to have a *channel state predictor* which would offer some level of forecasting about radio channel quality.

The scheduling policies should try to achieve a good performance in at least a few of the following issues: fairness, delay bound, throughput, implementation complexity, link utilization, graceful service degradation, isolation from the influence of other users and energy consumption. The next part of this section describes some of the wireless scheduling algorithms.

In general, the scheduling algorithms can be classified into two main groups: TDMA and CDMA based scheduling algorithms. The TDMA algorithms can service only one connection per given time unit within the assigned bandwidth. This gives the ability to adjust the transmission power level on the basis of a single connection and minimize the influence of other users on the physical properties of the radio channel. On the contrary, the CDMA approach allows for simultaneous transmission of multiple connection streams, but the transmit power must be limited to ensure the minimum cross-user channel degradation.

3.3.2.1 TDMA based schedulers

Some of the most important TDMA wireless scheduling algorithms are introduced here.

- **CSDPS** Channel State Dependent Packet Scheduling [105] is a framework which allows the use of different scheduling policies, like Round-Robin, Longest Queue First (LQF) or Earliest Deadline First (EDF). The main concept lies in the idea of suppressing the scheduled transmission if the relevant radio link quality is poor.

- **IWFQ** Idealized Wireless Fair Queuing [122] represents a realisation of the PGPS approach with a mechanism that compensates for the delay experienced by error prone sessions. Basically, sessions that have poor radio receptions are lagged by the system which assigns to them the lowest service tags. However, once they regain good radio link conditions they are given highest priority.

- **CIF-Q** Channel-Condition-Independent Fair Queuing [128] uses the Start Time Fair Queuing (**STFQ**) error free scheduling policy as a reference model. Each session has a lagging/leading counter, counting the number of packets by which the session is leading or lagging in respect to the STFQ model. Every leading connection has an assigned random variable $\alpha \in [0, 1]$. This parameter decides by how much a particular leading session is suppressed by the lagging session.

- **SBFA** Server-Based Fairness Approach [109] is a slightly modernized version of CSDPS. In addition to a mechanism for suppressing the connection if the radio channel condition is temporarily bad, it contains a special queue called a Long Term Fairness Server (LTFS). This queue always occupies a portion of the bandwidth and is used to speed up all lagging connections.

- **WFS** Wireless Fair Service [121, 123] is based on a modified version of the SCFQ algorithm. The internal mechanisms reduce the usage of time slots in an exponential manner if the leading counter is increasing.

3.3.2.2 CDMA based schedulers

In CDMA systems, each user's transmission appears as noice on every other user's reception within a given cell. Thus, scheduling algorithms for CDMA based systems distribute this across active connections. Access to radio resources in CDMA systems is controlled by sharing a total power budget amongst users in each cell. In [13] a solution was proposed for this problem. If the inequality 3.1 is true, the interference introduced by N^{th} connection will not degrade other connections unacceptably.

$$\sum_{i=1}^{N} g_i < 1 \tag{3.1}$$

Where:

- i is an index of i^{th} connection
- N is a total number of users, including the user which is about to be accepted or rejected
- $g_i = \frac{1}{1+\frac{G_i}{\gamma_i}}$ is a power index of the i^{th} session
- G_i is the fixed spreading gain of i^{th} session
- $\gamma_i = \frac{E_b(i)}{N_0}$ is the representation of expected BER level for i^{th} connection
- $E_b(i)$ is the energy per bit for i^{th} connection
- N_0 is the spectral noise density

The most important parameter is the relation between spreading gain and expected level of BER, $\frac{G_i}{\gamma_i}$. The maximum expected BER comes from minimum expected $\frac{E_b(i)}{N_0}$. Thus, minimising the BER requires higher $\frac{E_b(i)}{N_0}$. This causes the g_i to increase, meaning that more resources are taken from the radio channel. However, by increasing G_i, if possible, a reduction of g_i is achieved. In conclusion, lower error level requires higher spread gain or/and power index.

As with TDMA systems, CDMA systems have a great number of scheduling policies and it is not possible to present all of them. A brief description of the most common ones is given below.

- **Packet-by-Packet GPS** [12] is a PGPS model for CDMA based systems. Different transmission rates are assigned to a number of serviced sessions so that the sum of their power indexes is less than one. However, due to radio condition variations it may be necessary to offer higher gain to a certain connection to keep the required BER level, and this may not be achievable as this gain increase may affect all other connections being serviced.

- **SCDMA** Scheduled CDMA [11] is a scheduler that represents a hybrid CDMA/TDMA approach. The data is sent between a Base station and all associated mobile stations (MSs) in a slotted manner. However, during one slot, all MSs can transmit simultaneously if equation 3.1 for all active sessions is satisfied.

- **DRS** Dynamic Resource Scheduling [52] is a modification of SCDMA without the TDMA aspect. All traffic requested by the MSs is placed into one of two queues, Guaranteed Queue and Best Effort Queue. Based on this preliminary classification and particular radio channel conditions the DRS scheduler informs selected MSs when they can send their data. Thus, the scheduler makes sure that the equation 3.1 is satisfied.

- **WISPER** Wireless Multimedia Access Control Protocol with BER Scheduling [62] represents a much more advanced solution. It consists of a number of priority classes, with respect to an expected maximum BER level. Additionally, the delay performance is considered as one of the most important parameters.

3.3.3 Practical implementation issues

Implementation of a scheduling algorithm in a real system is always a challenge. The challenge grows in the case of wireless networks. The assumptions taken during the design and simulation test phase need to be relaxed. Phenomenon which have not been taken into account can expose noticeable changes in performance.

One of the most important features of a scheduling algorithm that aims to be implemented in a wireless environment, is its relatively low complexity and design simplicity. Fairness and delay performance are also important, but they play a less significant role since the perceived system performance depends greatly on the location and velocity of a particular user, so that the delay and fairness cannot be assured. On the contrary, simplicity and low complexity can be guaranteed. The obvious benefit coming from the use of low complexity solutions is an ability to sustain a high number of ongoing connections that carry real-time traffic. Another reason supporting the usage of simple scheduling algorithms is the fact that in most cases it is impossible to describe the traffic behaviour. This description is necessary in some scheduling policies to sustain good fairness among active users and to keep the delay boundaries agreed with each connection. Buffer space was previously taken into account, but it is not so important nowadays, since the cost of buffer space has been reduced significantly during recent years.

(E)GPRS type of networks typically use a simple Round Robin scheduling algorithm to manage a fairness between connections belonging to the same QoS class. Hence, the same approach was used when the performance of these networks is analysed [125]. Some extension over a standard Round Robin policy have been proposed. However, some of them lose the short term fairness aspect by suppressing the transmission when the channel is weak [126] or the complexity of the proposed solution is much higher than the Round Robin based scheduler [108]. The other approach presents Service Priority Scheduling (**SPS**), a simple implementation of RCSP, where the traffic is assigned to one of a number of queues, representing different delay classes, while the jobs in a particular queue are scheduled according to a FIFO policy. This approach is much simpler than Round Robin and can offer some delay performance differentiation among traffic classes but the fairness in a particular class is not achieved because of the FIFO policy [67, 108, 115].

The scheduling algorithms discussed above have limitations that affect the delay performance of a particular connection. For example, simple scheduling algorithms will not be able to offer a delay boundary, Table 3.1. More complex algorithms need so many initial conditions to work efficiently that they are not applicable in real networks. Some limitations are associated with relation between some physical property of the radio environment, like distortion coming from mobility, and the expected network throughput. It is obvious that, in a slowly fluctuating environment, issues like delay spread (in the case of TDMA base systems) and power control accuracy (in the case of CDMA networks) can be addressed in a more appropriate manner. Hence, in that scenario those networks can offer much higher throughput and can deploy more sophisticated scheduling algorithms which can yield a delay boundary. Taking into account all aspect of the limitations of the network it is very important to choose a resource scheduling algorithm that will exploit the properties of the expected radio channel condition. Thus, there is no a single algorithm that is superior to others, in each case the choice has to be suited to the expected working conditions.

3.3.4 Advantages of the separation MAC and RLC influence on IP packet delay

The delay of an IP packet transmitted over a mobile data system depends greatly on the MAC behaviour. Therefore, in the context of this thesis, it is very important to choose a MAC scheduling mechanism that will introduce a minimum of dynamic behaviour. In this way, the dynamic influence of the ARQ on the IP packet delay characteristic can be exposed and carefully analysed. Thus, it is decided that for the rest of this thesis an idealised MAC is used, that gives a fixed number of time slots per unit time as may be expected from a simple RR scheme in a static user environment. It is therefore possible to separate the influence of MAC from influence of RLC and PHY. Additionally, this approach is easily implemented, requires low CPU power and its reaction time is very quick. The main disadvantage is that it may not exploit the possibility of using information about the temporal state of the channel and carried traffic, like C/I reports and characteristic of the carried traffic.

Moreover, once the MAC influence is separated and the RLC and PHY influence is analysed a new window of opportunity is opened. If the nature of ARQ influence is known and the state of PHY conditions is predictable [3] then the prediction of an average delay is possible. Knowing the expected IP packet delay performance when the MAC is simple in its nature, it is possible to adjust the number of channels that are accessible for a certain connection. For example, if a transmission is carried on one data channel and it is expected that the average delay will be 20% above the value promised by a network, this information can be used as a trigger mechanism for assigning an extra portion of the radio resources that will speed up the transfer and minimise the delay, so the promised maximum delay introduced by a network can be achieved.

3.4 PHY aspects

The influence of the PHY layer on the IP packet delay characteristic is profound. The MAC scheduling algorithm can assign a higher or lower portion of the radio resources but it is the PHY characteristic that decides how good the overall connection performance is. A user that is experiencing a highly attenuated path will achieve worse delay and loss performance than a user which is experiencing a good quality transmission. The MAC can compensate for the lack of radio link quality by increasing the portion of the radio resources, however, it has limited capacity to improve the overall connection performance and this become impossible as the cell load approaches saturation.

The PHY influence on the IP packet transport process is presented in more detail within this section. Following that the methods of modelling the PHY influence are going to be briefly presented and their pros and cons are going to be discussed. In particular, problems related to models used in analysis and prediction will be of particular interest. Finally, some remarks in regards to modelling the PHY influence on the Incremental Redundancy policy are presented.

3.4.1 Transport of IP packet by PHY

As was mentioned before, the process of sending an IP packet over the wireless channel is a complex issue.Therefore, it is natural that it is separated into a num-

[3] As a statistical model is used here, a simple EWMA approach could be used.

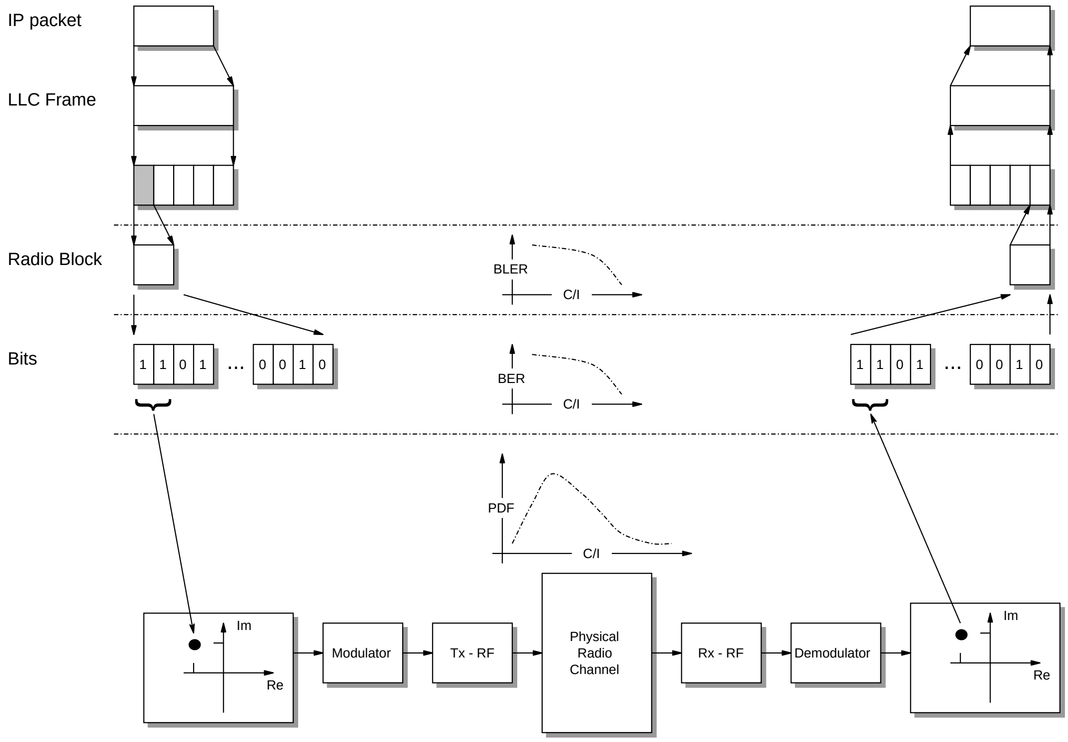


Figure 3.1: *Process of sending an IP packet through a wireless link*

ber of sub-tasks, as shown in figure 3.1.

- modelling physical properties of a radio channel
- modelling the characteristics of BER
- modelling the BLER characteristic

These three classes of problems will be addressed in some more detail in the following sub-sections.

3.4.2 Methods of simulating the radio channel

Radio channel models can be classified into three groups: Path Loss , Small-Scale Fading and Impulse-Responses models. Additionally, all three groups are split into outdoor and indoor sub-categories [92]. A good survey of propagation models for mobile communication can be found in [8, 132]. A brief introduction to the most popular models is given here.

3.4.2.1 Path loss models

Path models aim to address the issue of computing the average value of received power at a certain distance from the transmitting antenna. The most well known and used outdoor models are:

- **Free-Space model** [138] is an idealized model that is used as a reference point for more advanced methods and for point to point line of sight link budgets. It is described by the following formula:

  $$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2}[W] \tag{3.2}$$

  Where:

  - P_t represents the transmitted power in W
  - P_r represents the received power in W
  - G_r represents the gain of transmitter's antenna, where this gain is referred to an idealized isotropic antenna
  - G_r represents the gain of receiver's antenna, where this gain is referred to an idealized isotropic antenna
  - λ represents the wavelength of the central frequency of the transmitted signal

- **Path-Loss model** is an extension of the Free-Space model that allows distance dependent exponent to be a variable (n). It can also accommodate a statistical variation of the power, which aims to reflect the shadowing effect.

  $$\overline{PL(d)}[dB] = PL(d_0) + 10 \cdot n \cdot log_{10}\left(\frac{d}{d_0}\right) + X_\sigma[dB] \tag{3.3}$$

  Where:

  - $\overline{PL(d)}$ is an average path loss at the distance d between the transmitter and the receiver
  - $PL(d_0)$ is the free-space path loss, at a close-in reference distance d_0
  - n is path-loss exponent, which depends on the type of the surrounding environment
  - X_σ represents the variation of the path loss. It represents Gaussian process with variance equal to σ and the value of medium placed at the power value given by the Free-Space model at the given distance d.

- **Okumura et al. model** [144] this model is based on a large collection of measurements for different terrain and building settings. The results were carefully analysed by statistical means and expressed in forms of graphs. Following the estimation of the median field strength, taken from these graphs, a series of adjustments is performed, to consider a specific settings of the analysed terrain. The major strength of this method is its relatively low computational complexity with a relatively good accuracy.

- **Hata model** [59] this model is based on the Okumura model. It mimics the behaviour of the Okumura method for large cell systems by means of analytical equations.
- **COST-231 Walfish-Ikegami model** [30] is used typically in urban and suburban environments, where building placement is relatively uniformly distributed. It utilizes free space model with respect of roof diffraction and multi-screen diffraction loss. This model is recommended by the ITU-R for evaluation of the IMT-2000 standard.

Indoor propagation has to cope with many reflections and crossing walls or partitions before reaching the receiver. The material, shape and size of these walls and other elements play a dominant role in computing the average power level at the receiver. The **Distance/Power model** is the most most popular model for path loss estimation within indoor environment, Equation 3.4. While the most research is done in the domain of tuning this model to specific locations, different values of n, Floor Attenuation Factor ,$FAF(f)$, and Wall Attenuation Factor,$WAF(w)$, like: office buildings, factory space, private room, number of passing walls, F, and floors, W, etc [79, 96, 111]. The Distance/Power model is a modification of Path Loss model, Equation 3.3, that considers the influence of signals going through floors and walls, $FAF(f)$ and $WAF(w)$ respectively.

$$\overline{PL(d)}[dB] = PL(d_0) + 10 \cdot n \cdot log_{10}\left(\frac{d}{d_0}\right) + \sum_{f=1}^{F} FAF(f) + \sum_{w=1}^{W} WAF(w)[dB] \quad (3.4)$$

All the models described above represent a class of radio channel models having their roots in empirical measurements or statistical modelling of the surrounding environment. However, in some cases the accuracy offered by those models is not sufficient and some more complex methods, called site specific, must be used. Methods belonging to this group are fed extensively with data coming from site measurements. Naturally, this approach is characterized by much higher accuracy, since the model represents more accurately the tested environment. Nonetheless, their complexity is huge and they require extensive computation power, which can be justified only if the benefit of improved accuracy is obvious. Some example of this methods can be found in [7, 27].

3.4.2.2 Small-Scale fading models

Multipath transmission is a natural state of mobile communication, thus, the signal experiences dramatic changes of its amplitude. The path loss models are not sufficient, as they offer only a mean value of power experienced at a certain location from the transmitting antenna. Therefore, the Small-Scale fading models have been introduced, which aim to model a distribution of power during the transmission period. There are a number of models addressing this issue [87, 88, 136, 146]. Two of these models are of particular note, Ricean Distribution and Rayleigh Distribution models:

- **Ricean Distribution** is used to model an environment where the scattered and reflected waves have one dominant component. This component is significant when a LOS path is present. The following formula describes the Ricean power distribution:

$$p(r) = \begin{cases} \frac{r}{\sigma} \cdot exp\left[-\frac{r^2+A^2}{2\cdot\sigma^2}\right] \cdot I_0\left(\frac{A\cdot r}{\sigma^2}\right) & \text{if } A \geq 0, r \geq 0; \\ 0 & \text{if } r < 0; \end{cases} \tag{3.5}$$

  Where:

  - r is an amplitude of the envelope of the receive signal
  - $2 \cdot \sigma^2$ is the mean power of the multipath signal
  - A is the peak amplitude of the dominant signal, amplitude of the LoS component
  - $I_0(\cdot)$ is the modified Bessel function of the the zero order

- **Rayleigh Distribution** is used to model an environment that is characterized by a high order of multipath transmission influence. The multiple reflections play a dominant role in shaping the signal's envelope for NLOS transmission. Lack of strong LOS signal causes increasing speed and depth of the signal's power fluctuation. Thus, the Rayleigh power distribution is much wider that the Ricean one. The following formula defines Rayleigh distribution:

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \cdot exp\left(-\frac{r^2}{2\cdot\sigma^2}\right) & \text{if } 0 \leq r \leq \infty; \\ 0 & \text{if } r < 0; \end{cases} \tag{3.6}$$

3.4.2.3 Impulse-Response models

Inter Symbol Interference (ISI) influence can be neglected when the transmission rate is much lower than the coherent bandwidth. However, if this is not the case a model considering the ISI presence has to be deployed.

These models can be roughly classified into three categories, Models Based on Measurements Results, Statistical Models of Time Delay Spread and Deterministic Models of Time Delay Spread.

- The first group aims to describe different environments in terms of channel response to a test signal. A number of tests have been performed and average values have been assigned to different terrain and indoor settings [43, 44, 113].

- The second group wants to exploit the idea of statistical description of the radio wave behaviour. An example is a Two-Ray Rayleigh Fading model [110], where only two rays are assumed to have significant impact on the channel properties. Each of these ray attenuation properties is controlled by Rayleigh distributions, while phase shift is assumed to be of uniform nature. This model can be easily extended into a more than two rays [151] and cover more advanced transmission scenarios.

- The last model set is location dependent and its channel response description is a result of extensive computation based mainly on the Ray Tracing method [82, 120].

3.4.3 BER characteristics

All of the methods described above give more or less accurate description of the radio channel. However, since modern mobile data systems are digital in nature it is necessary to express the performance of a system in digital domain descriptors, such as Bit Error Rate (BER).

The BER is dependent on many factors like chosen modulation and detection technique and properties of the radio channel. Thus, the usual description of BER characteristic is a graph of BER as a function on Carrier-to-Interference ratio (C/I) [4]. If a more advanced techniques of improving radio reception is used it has to be considered as well. Thus, it is very hard to determined these relations analytically. Nonetheless, some simple scenarios have been described in an analytical way and BER as a function of signal to noise ratio has been derived [106, 119, 129]. However, if the considered system imposes advanced techniques like CDMA, deep interleaving, equalisers, Frequency Hopping (FH) etc. it is quite often impossible to derive an appropriate mathematical formula describing BER as a function of C/I. Hence, the only way to plot these curves is by building an appropriate simulator, which will include the impact of all considered mechanisms. A significant

[4] In case of system unaware modulation/detection analysis the C/I is represented as Eb/N

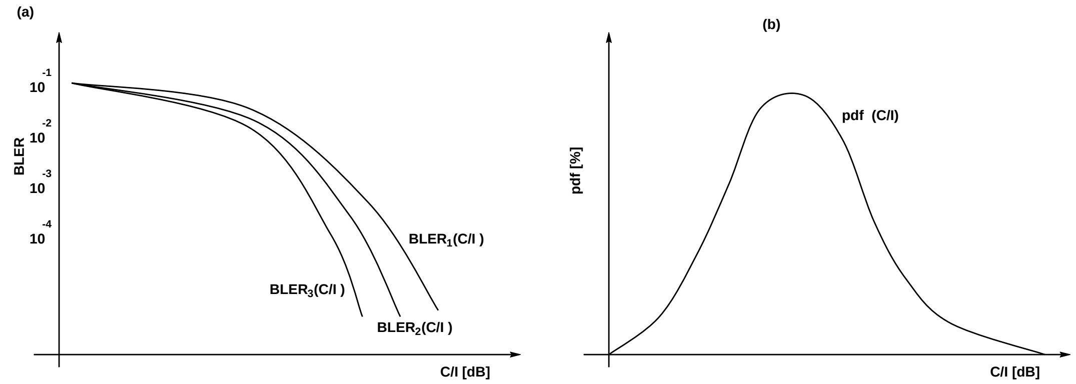


Figure 3.2: *BLER vs C/I for three transmission attempts in IR mode with relevant PDF distribution*

number of such plots have been prepared for GSM/GPRS and WCDMA systems by ETSI and later by 3GPP/3GPP2 [5].

The plots of BER vs C/I give quite a detailed description of the system behaviour. It introduces high computation complexity, especially if the simulator considers a detail implementation of the transmission process. Thus, to simplify the simulation process an average value of BER is often introduced. If the BER plot, BER(C/I), is known for the considered system and its radio propagation scenarios are known, pdf(C/I), then the average BER, $\overline{BER}$, can be calculated as [151]:

$$\overline{BER} = \int_0^\infty \{BER(C/I) \cdot pdf(C/I)\}\, d(C/I) \tag{3.7}$$

3.4.4 BLER charccteristics

The average BER describes the physical properties of the radio system mapped into the digital domain quite well. However, this performance descriptor is not sufficient for a more detailed system performance survey. Since most of the traffic carried by mobile data systems is IP based, it is important to model and include the error process at higher layers. This mean that the error process of the radio block is crucial for a detail analysis of IP traffic performance.

As with the BER case, it is possible to perform a series of simulations for a given combination of modulation/detection and advanced transmission settings and obtain a BLER as a function of C/I as in figure 3.2. Having these graphs and the distribution of the received power during the transmission it is possible to compute the average BLER value, $\overline{BLER}$.

$$\overline{BLER} = \int_0^\infty \{BLER(C/I) \cdot pdf(C/I)\}\, d(C/I) \tag{3.8}$$

[5] An example of such document is: Tdoc SMG2 EDGE 274/99 9rev (2), ETSI SMG2 EDGE Working Session, France, August1999.

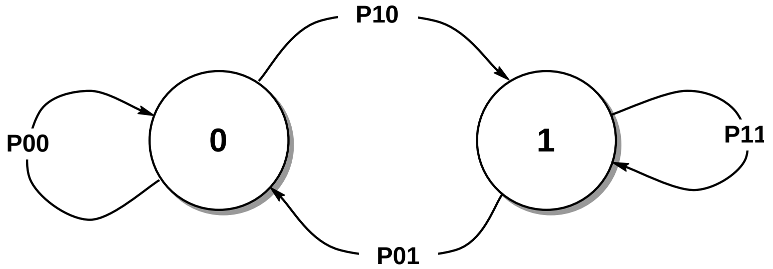


Figure 3.3: *State transition diagram for simplified Gilbert model*

However, it has been reported that the simple description of BLER in form of average BLER is not sufficient is some scenarios [155]. Such a description implies an Independently and Identically Distributed (i.i.d.) characteristic of the error process with an inherent lack of process memory. However, in many transmission scenarios, in particular within a wireless environment, these error processes are not memoryless. The presence of memory in the process requires a more detailed description of this process, including the presence of bursty error characteristic.

One of the most direct candidates for a mathematical descriptor of a radio channel with bursty error property is the Markov chain. Some solutions based on a simple two state Markov chain has been proposed by Gilbert and Elliot [145, 154]. Here, the two states represent good and bad status of the transmission channel, 0 and 1 respectively, figure 3.3. The channel is described fully by its packet error process transition matrix, $\mathbb{P}$, and the probability of packet error is $\overline{BLER}$, equations 3.9 and 3.10.

This obviously gives a rich description of this process [90, 152]. However, it assumes that this process is stochastic, which is not always the case in mobile transmission [9]. Additionally, in most cases it is assumed that the ISI phenomenon is not present or its influence is minimised and that there is perfect power adaptation and a quasi-stationary power envelope during the entire radio block transmission period [153].

$$\mathbb{P} = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} \tag{3.9}$$

$$\overline{BLER} = \frac{p_{01}}{p_{10} + p_{01}} \tag{3.10}$$

3.4.5 Simulation and Prediction

One important issue that also needs to be considered is the impact of the chosen model on the predictability of system performance. Most radio channel models, are well suited to the analysis of performance issues in different transmission conditions and system configuration scenarios. But the application of these models to real systems can be difficult.

When some prediction technique is being investigated it is more likely to be based on real life data [45]. This data can be reports of history about experienced C/I ratio, BER or BLER during the previous transmission attempts [65, 93, 98]. Such radio channel descriptors may lack the mathematical clarity of the models described previously, nonetheless, they usually inherit the most significant aspects of the channel. Hence, they offer a more accurate description of the analysed system performance state. Therefore most of current adaptation techniques that try to predict the behaviour of the system in the near future try to track the previously experienced performance descriptors like, C/I, BER or BLER statistics [50, 98, 131].

3.4.6 Incremental Redundancy (IR) issues

The other problem for many radio channel models is the Incremental Redundancy technique and its influence on the BLER characteristic. The BLER of a Hybrid Type I ARQ, with simple FEC, can be simulated by using i.i.d. or Markov chain based models. It can be assumed that the distribution of errors at the radio block level is known, or if additional PHY level data is known, one or a combination of channel modelling techniques can be used to obtain a power distribution. This power distribution plus graphs of Modulation and Codding Scheme (MCS) performance under certain assumed mobility and terrain conditions can be used to compute an average BLER, equation 3.8.

However, if IR is present, the information that is stored at the receiver can be used to correctly decode the retransmitted bits. Thus at each iteration of retransmission decoding attempt the curve of BLER vs. C/I has a different shape, Figure 3.2. Hence, the average BLER is different after first, second and third transmission attempt, Equation 3.12.

$$\begin{aligned}
\overline{BLER_1} &= \int_0^\infty \{BLER_1(C/I)\cdot pdf(C/I)\}\, d(C/I) \\
\overline{BLER_2} &= \int_0^\infty \{BLER_2(C/I)\cdot pdf(C/I)\}\, d(C/I) \\
\overline{BLER_3} &= \int_0^\infty \{BLER_3(C/I)\cdot pdf(C/I)\}\, d(C/I)
\end{aligned} \tag{3.11}$$

None of the existing models are capable of capturing this phenomenon. The simple i.i.d. models deals with a single average BLER value, as does the error process model based on the Markov chain.

3.5 RLC aspects

The impact of the RLC layer is defined by the trade-off between reliability, throughput and delay. The RLC layer has a variety of techniques to improve the error characteristic of a particular connection, although this improvement usually decreases the throughput and delay performance. Thus, it is important to be able to predict the influence of the RLC layer on the delay performance of IP packets and use mechanisms that will result in minimising the delay associated with the process of sending this IP packet.

Since the dynamic nature of the influence of ARQ based techniques on the delay of IP packets are the prime focus of this thesis, a brief classification of these methods will be delivered with emphasis on the throughput and reliability. Then, the works that address the problem of the influence of BEC techniques on the delay performance of radio blocks and IP packet will be examined.

3.5.1 BEC techniques - principal issues

The principles behind BEC techniques are shown in Figure 3.4. The system has a transmitter buffer from which radio blocks are sent through the radio channel towards the receiver. The radio channel introduces a certain probability of error, $P_t(E)$, and transmission latency, T_{td}. On the other side of the connection there is a receiver buffer, waiting for radio blocks. If radio blocks are incorrectly received and the receiver sends a message through the feedback channel to the transmitter. This channel has its own probability of error, $P_f(E)$, and feedback transmission latency, T_{fd}.

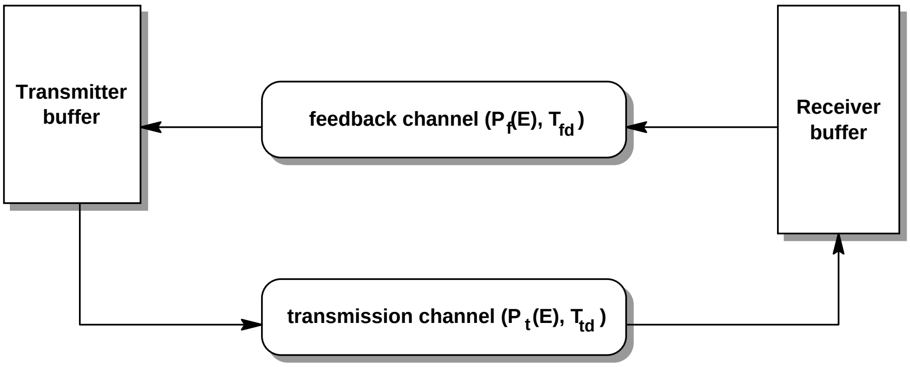

Transmitter buffer
feedback channel (P$_f$(E), T$_{fd}$)
Receiver buffer
transmission channel (P $_t$(E), T$_{td}$)

Figure 3.4: *Simplistic view of BEC techniques*

In most of the research work related to ARQ systems it is assumed that the error in the feedback channel,$P_f(E)$, can be neglected [32, 112, 141]. One of the reasons behind this is that these messages carry a relatively small amount of data, reports about the status of previously conducted transmissions, thus they can be heavily protected.

The feedback transmission delay, T_{fd}, is influenced by two things: the properties of PHY layer and the protocol scheduling policy that decides when and how the report is going to be released. The PHY layer delay is dominated by the design of a particular network, for example a radio block is 20 ms long in (E)GPRS systems, while 10 ms long in UMTS Rel.'99.

In most models considered in the field of ARQ research the message is generated instantaneously with a relevant radio block reception at the receiver and the physical propagation time is assumed to be negligible, $T_{fd} = 0$ [25, 37, 74]. However, in practical systems this assumption is not valid, as there is a time needed for detecting incorrectly received transmission, sending the appropriate report back to the transmitter and scheduling this unsuccessfully transmitted radio block for retransmission. Thus, some work has addressed this non zero feedback delay [26, 77, 89].

The effect of scheduling policy of the retransmitted traffic is very important, but rarely considered. It is not a problem when the feedback is idealised because the retransmission message is usually prioritised over new traffic. However, in real systems feedback delay exists and some radio blocks need to be retransmit several times before correct reception takes place. Thus, it is reasonable to give higher priority to radio blocks that have a higher number of previously performed transmissions. This creates a number of priority classes , where the higher priority is given to the radio blocks that have been retransmitted a higher number of times, as it is quite likely that higher layers are waiting for these radio blocks. This issue of ARQ transmission is not addressed in the literature to date and will be addressed in this thesis.

The delay of the transmission channel, T_{td}, is affected by similar factors, but the error process of the channel plays the most important role.

For throughput and reliability performance analysis, the channel is modelled often, in great detail, using Rician and Rayleigh models. These are mapped into BER characteristics for a certain combination of modulation/detection methods and a particular level of FEC protection [80, 112, 141].

When the delay characteristic of a particular ARQ technique is considered, the error process for radio blocks is used to represent the radio channel influence [74, 89]. Similarly, the ARQ loop influence on the IP packet delay characteristic presented further in this thesis considers the influence of PHY layer at the radio block level. More details of models based on the BLER process will be given in the next subsection.

The transmitter and the receiver buffer capacity introduced problems when the price of silicon based memory was significant. With the progress in miniaturisation of digital circuits [1] it is possible to over-provision the memory capacity. Thus, it is assumed that buffer sizes at the transmitter and the receiver are large enough to hold all unacknowledged packet (mostly less than 10kBytes), so their impact can be neglected.

3.5.2 BEC classification

The BEC technique can be roughly classified into two groups, pure BEC and Hybrid solutions section 2.4.2. There are three principle pure BEC techniques: S&W, GBN and SR. The SR-ARQ is superior to S&W and GBN in terms of reliability and throughput, as it retransmits only radio blocks that have been incorrectly received [118]. The advantage of two other techniques is their simplicity and smaller power consumption due to lower memory requirements[6].

The Hybrid techniques are extensions of pure BEC systems. The code used there is able not only to detect errors but also to correct some level of errors. The first Hybrid technique called Hybrid Type I ARQ is a straight forward implementation of this idea. The payload is protected by redundant bits which decrease the available payload space in a radio block but also decrease the number of unsuccessfully transported radio blocks [118, 140].

Hybrid II based techniques save all incorrectly received radio blocks and by deploying advanced coding methods are able to use information from unsuccessful retransmissions [86, 107]. This obviously gives an advantage over Hybrid Type I ARQ. Hence, they are used in modern mobile data systems, eg: EGPRS and HSDPA.

The Hybrid III ARQ techniques goes further that Hybrid II by using such FEC codes which allow for correct detection of independent transmission attempts, like in Hybrid I systems, and for merging all stored versions of incorrectly received radio blocks, like in Hybrid II ARQ [142].

The Hybrid Type II/III ARQ uses mainly two coding techniques; Chase combining and Incremental Redundancy (IR). The Chase combining approach combines the data belonging to previous radio block transmission attempt(s) with the new radio block transmission attempt. Each transmission attempt increases the probability of gathering enough correctly received bits to decode the analysed radio block correctly. In this technique all versions of the transmitted radio block are the same. On the contrary, the Incremental Redundancy has a different version of the same radio block at each of the transmission attempts. These different versions are the result of coding the payload of the radio block with a redundancy code and cutting it further into a number of radio blocks. The coding techniques used in IR offer the ability of creating versions of the considered radio block in such a way that the first transmission attempt will contain the original data (before redundancy coding). Hence, the first radio block transmission attempt is sent

[6] In Mobile Stations the need for memory usage limitation is a result of the battery drain caused by the use of memory

without any FEC protecting its content. If additional transmissions are required, due to previous failure, the information transmitted at the next transmission attempt(s) are another part of the redundancy code. Thus, at each transmission the FEC protection level grows faster than in the case of Chase combining, which is using the same version of the transmitted radio block.

3.5.3 Performance aspects of BEC techniques

The research related to ARQ systems has two main streams, throughput-oriented and delay-oriented analysis. The dominant aspect for most researchers has been throughput performance. This was driven by the assumption that the throughput characteristic is the most important feature of ARQ techniques. For delay insensitive traffic and for simple transport protocols this is true. However, when the transported traffic is delay sensitive, eg. VoIP or multimedia, or the transport protocol represents a higher level of sophistication, like TCP, the delay characteristic of the BEC technique becomes very important. Interaction between the ARQ mechanism and internal TCP flow and congestion control mechanisms can substantionally degrade the overall end-to-end performance [58].

3.5.3.1 Throughput-oriented BEC research

It has been shown that the SR-ARQ technique offers the best throughput performance among all simple BEC techniques . In the case of Hybrid Type I ARQ, where additional FEC is added, the throughput performance of the transmission in a noisy channel is superior to the pure BEC solution [118]. Hybrid Type II and III ARQ are superior to Hybrid Type I ARQ systems [142].

The main effort in Hybrid systems has been focused on the finding an appropriate FEC code, which will maximize the throughput of the selected Hybrid technique within the assumed transmission environment [112, 141, 142].

3.5.3.2 Delay-oriented BEC research

The Delay-oriented BEC research did not attract as much attention as the problem of maximising the achievable throughput performance. However, there is a number of publications that deals with this issue of analysing the influence of ARQ on the delay of data transmission [36, 74, 77, 89, 91].

The three major components that contribute to delay of the radio block within ARQ loop are: Queueing Delay, Transmission Delay and Re-Sequencing Delay, Figure 3.5.

- **Queueing Delay** represents the time which the radio block has to spend in the transmitter buffer queueing for access to the radio resources. This time

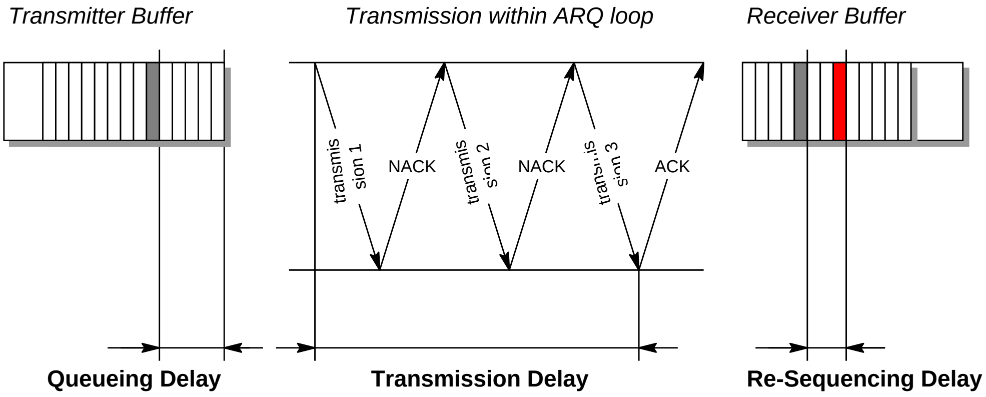


Figure 3.5: *Delay components in tranismission of radio block through ARQ loop*

delay depends on the source characteristic, the channel characteristic, the MAC scheduler and chosen MCS.

- **Transmission Delay** specifies the time between first transmission attempt of a particular radio block and its successful reception. This implies that this delay depends only on the radio channel characteristic.

- **Re-Sequencing Delay** defines the time which is spent by the particular radio block waiting for all previously transmitted radio blocks for correct reception. This delay characteristic is the most difficult to analyse, as it depends on the error process being experienced by the analysed radio block and all previously transmitted radio blocks.

The i.i.d. radio channel model was dominant at the early stage of the delay oriented analysis of ARQ techniques, with respect to different parts of radio block delay components and their combinations [77, 91]. The radio channel with time varying characteristic was first investigated in [36] by use of queueing theory. This approach did not consider the Re-Sequencing Delay aspect of the radio block delay.

All ARQ delay contributions have been addressed in [74], although there were a number of assumptions which made the considered system unrealistic, notably the assumption about *ideal SR*, which assumes an instantaneous feedback message. The work presented in [89] eliminates this assumption. However, during the computation of the Re-Sequencing Delay it neglects the influence of retransmitted radio blocks. Additionally, it assumes that each radio block has a static time slot assigned to its transmission attempts, eg: if radio block number three has been corrupted and needs to be retransmitted, this retransmission will take place at third place of the new *fundamental ARQ window*.

The above approaches give a good insight into the behaviour of ARQ systems in terms of throughput and delay analysis. However, their accuracy is limited by a number of assumptions taken in respect to the source of data, the radio channel characteristic and ARQ feedback loop. Therefore, it would be hard to rely fully on such models when a prediction of the influence of ARQ loop on the IP packet delay characteristic is needed in a real world system. A new model, much closer to the real characteristic of the radio block error process and real delays associated with the feedback has to be proposed. Such a new model, addressing the requirements related to the analysis and prediction of ARQ influence on the IP packet delay, is proposed in the next chapter.

3.6 Summary

This chapter gave an overview about the influence of different communication layers on the connection delay characteristic.

The performance of the TCP layer is highly dependent on the reliability and the delay characteristic of the peer to peer connection. This dependency is clearly visible once a link containing a cellular radio connection is present.

The MAC layer contributes substantially to the delay characteristic of a radio link and a great number of articles address different aspects of MAC influence on the radio link delay characteristic. Fortunately, it is possible to separate this influence and focus entirely on the RLC and PHY contribution to the delay of transported IP packets, which have not been as widely studied so far.

As the PHY plays a significant role in the altering the IP packet delay characteristic it is important to model it in a way that offers sufficient accuracy with minimum complexity. Additionally, the ability to predict the PHY state would be beneficial, as having that property could allow for more advanced prediction of delay introduced by RLC layer. Thus, the very complex models involving Markov chains are too complex for real time execution and hardly predictable. On the other hand, simple i.i.d. radio error models are too simplistic to capture some of the properties of Hybrid Type II and III ARQ systems.

The number of proposals mentioned in the literature do not allow for simple analysis of the impact of Hybrid Type I/II and III ARQ system on the IP packet delay. This is due to the number of assumptions that limit the use of the proposed analysis. These limitations trigger the idea of further modification of existing RLC analysis models towards one that would accommodate all the most important aspect of the analysis of ARQ loop influence on the IP packet delay characteristic.

CHAPTER 4

ARQ loop analysis

In this chapter the method of analysis of the SR-ARQ influence on IP packet delay is presented. In order to do this, firstly the influence of all important factors will be investigated. Therefore, the following issues will be addressed: IP packet size, SR-ARQ design and radio channel influence. Following that, a methodology for analysis of SR-ARQ influence will be proposed and explained in detail.

The new methodology is then used to investigate the characteristic of average IP packet delay and IP packet delay distributions. This leads to a discussion about usability of average IP packet delay vs IP packet delay distribution characteristics.

4.1 Introduction

It was shown in Chapter 2 that the delivery of an IP packet over a mobile data network is subject to very complex interactions between different parts of the protocol stack. This cross interaction makes a detailed analysis of that characteristic extremely difficult and connection setting dependent. The analysis of this delay characteristic is approached here by separating the influence of particular layers and treating them independently. Such a separation is possible in the networks that do not change the RLC or MAC settings rapidly, so there is enough time to gather statistical samples that are used for calculating the statistical description of the PHY effects mapped onto the RLC layer. An example of such network is (E)GPRS type of networks, in case of GPRS it is valid for all RLC modes, in the case of EGPRS it is true when (Incremental Redundancy) IR is used by the RLC.

The main end to end delay contributors considered in Chapter 2 include TCP, MAC, RLC and PHY. TCP does not affect the delay of an IP packet, as it uses IP packets to carry its own PDUs, called TCP segments. Hence, there are only three major contributors to the total IP packet delay characteristic. Since the MAC influence on the IP packet delay performance has been extensively studied in the past, it is not going to be deeply investigated in this work. However, the influence of RLC layers, and advanced Hybrid Type II and III ARQ techniques in particular, on the IP packet delay characteristic, forms the main area of interest of this work. The RLC performance is closely linked with the behaviour of PHY for a particular link, thus, this phenomenon will also be considered.

Central to this chapter is the analysis of the delay behaviour of IP packets altered by the ARQ feedback loop mechanisms. This analysis is IP centric - rather than bit, bytes or data flow centric. This implies that the granularity and the size of IP packets should have a significant reflection in the analysis model structure.

An important factor that needs to be considered is the ability to readily gather input data, as this would allow the model to be tuned easily into any existing network. Additionally, the model used for this delay characteristic analysis should be simple and intuitive yet provide a structure to capture the most important mechanisms at RLC and PHY layers affecting the characteristic being analysed. The model used for this analysis should incorporate the features of the most advanced Hybrid Type II and III ARQ techniques, as these techniques are becoming incorporated into more mobile data networks.

Furthermore, the aspect of re-sequencing radio blocks at the Transmitter, in case of retransmission of a certain radio block, should be incorporated into the model. This re-sequencing phenomenon has not been investigated by others. Considering that there is priority given to retransmitted radio blocks over those being scheduled for first transmission attempt, it seems obvious to take this issue into account.

Finally, the model should be easily expandable to be used for prediction of ARQ loop performance and the impact this has on the IP packet delay performance in real time.

Unfortunately, none of the models used so far can offer this range of features. Thus, there is a need for a new model that will address the current models deficiencies, whilst incorporating the previously described features. This new model can be used to capture and closely analyse the effect of ARQ feedback loop technique on the IP packet delay performance.

Such a model will build on current models, either by imposing some limitations to very advanced models or by introducing extensions to simpler ones. In order to facilitate the above requirements, the following assumptions are made:

- The influence of LLC layer will be neglected. As described in the previous chapter this layer only introduces static delay to IP packet transmission, therefore its presence can be modelled as an extra static delay.

- The influence of MAC is modelled as an idealised MAC, which gives a fixed number of time slots per unit time, as may be expected from a simple RR scheme in a static user environment. Therefore, MAC influence becomes static, which allows the separation and close analysis of the influence of BEC mechanisms on the IP packet, independent of the delay characteristic.
- The PHY will not use an advanced Markovian chain type process to represent the error process of radio blocks. Although, this means that the property of burstiness will not be fully captured, it will simplify the model and will allow future extensions of the model to include the ability to predict the PHY behaviour. This would be hard to obtain in the case of Markovian chain based radio channel models.
- The RLC will not facilitate power or Modulation and Coding Scheme adaptation during the transmission of a particular IP packet. This limitation is not too unrealistic, since the Hybrid Type II or III ARQ itself is a form of time diversity and code adaptation and does not have to be used in conjunction with these techniques to perform well.

4.2 Proposal of the methodology

The goal is to estimate the delay of an IP packet as a function of its size, taking account of the design parameters of the ARQ loop and the radio channel conditions. In particular, this methodology focuses on small IP packets, for example, Voice Over IP (VoIP) packets are typically 240 bytes long and would require less than 11 radio blocks in the EGPRS system.

The methodology begins when an IP packet of some known size is presented to the RLC for transmission. It is assumed that the traffic channel and signalling channel have already been established in both directions. If this is not the case, then substantial delays can be incurred while negotiating for access to radio resources. Since this work is primarily aimed at real-time applications and the assumption is that the traffic source is a constant bit rate one, constant packet size stream so that there is no requirement for a stochastic traffic generator. Even if variable bit rate sources occur in the real system, the results for each packet size found in this work are still applicable in a general sense as the constant bit rate assumption is valid over the short time intervals that the ARQ loop operates over.

4.2.1 Novel ARQ loop model

It was shown in Chapter 3 that there are two major ways of modelling the ARQ loop delays. The first one is called *ideal SR-ARQ*, and assumes that the feedback messages are received instantaneously after a successful or errored transmission attempt, as presented in figure 4.1 (a). The second type of ARQ loop delay models assumes the static delay associated with the transmission of the feedback messages from the receiver to the transmitter, figure 4.1 (b). However, it is possible that the second retransmission attempt may not experience the same feedback delay as the first one. Thus, in the model proposed the assumption about the same feedback delay at each retransmission attempt is relaxed and these retransmissions may have different feedback delays for each transmission attempt, see figure 4.1 (c).

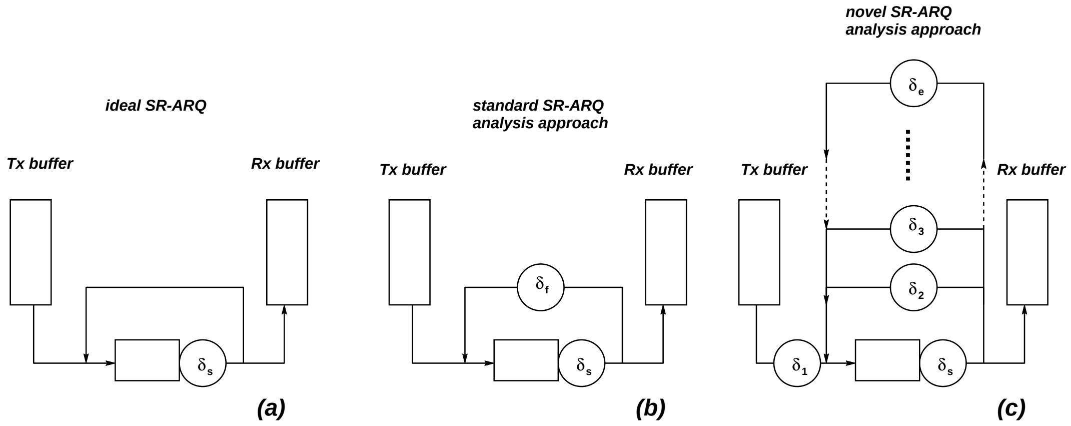


Figure 4.1: *Three different ARQ loop delay models*

To accommodate the possibility of different values of the delay associated with each transmission attempt, the design of the ARQ loop is represented by a vector of delays, as in equation 4.1.

$$\overline{\Delta_{general}} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \dots \\ \delta_e \end{bmatrix} \tag{4.1}$$

Where, δ_1 represents the system delay[1] associated with the first transmission attempt. Whereas, δ_2, δ_3, ... , represents the time between the end of the radio link transmission at the receiver and the moment of scheduling this radio block for the next transmission attempt. Finally, δ_e represents the delay associated with the last transmission attempt in the case of errored reception after a number of transmission attempts. This delay describes the time between between the end of the radio link transmission at the receiver and the moment of discovery that this radio block is wrongly transmitted.

[1] The system delay represents all delay associated with preparation of the radio block for the transmission through the radio link; eg. interleaving, coding or transmission delay caused by significant separation of the actual transmitter and its coding/interleaving module.

It is assumed that in most real systems the maximum number of transmission attempts is three[2], so, the vector has the following form:

$$\overline{\Delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_e \end{bmatrix} \tag{4.2}$$

The retransmission limit is based on the fact that real time traffic should not have too many retransmissions due to its delay constraints. The unit of delay used within the simulator is the time taken to transmit a radio block over a radio channel. For example 20ms is the block period of EGPRS and is used to give some realistic delay results. Here this block period will be denoted δ_S, and all other delays are an integer multiples of it.

4.2.2 Novel radio channel condition model

As it was outlined in Chapter 3 there is a problem of properly expressing the properties of the radio block error process when Incremental Redundancy is deployed. Simply, a single variable is not sufficient to describe the fact that the probability of successful reception is higher after double transmission attempt than after a single one.

To overcome this impairment a new way of describing the PHY properties is proposed. This novel radio channel condition model does not try to simulate the process of physical layer variations coupled with mapping this into the error process of radio blocks. Instead, the error process of radio blocks is directly modelled at the RLC layer by using a statistical descriptor. This is done by assuming that the distribution of successful transmission attempts can be predicted from the measured history of the radio block transmission success rate.

The previously used methods for representing PHY influence at the radio block level have significant problems with their ability of coping with Hybrid Type II/III ARQ influence on the radio block process. None of them (i.i.d. or Markovian based model) can easily model a system when Hybrid Type II/III ARQ is deployed. Additionally, models based on Markov chain are characterized by problems with accurate prediction of higher order statistics of the radio block error process. The Markovian approach is very good when the ARQ loop is analyzed for a given set of Markov chain parameters. However, the accuracy of the prediction of these parameters presents a significant challenge.

On the contrary, the model proposed here, which is based on statistical description of PHY at the radio block level, is able to cope with all three Hybrid

[2] This assumption comes from the fact that in most Hybrid Type II/III ARQ algorithms the number of decoding attempts is three [142].

ARQ types. Moreover, its statistical character makes it easier to have its parameters predicted in real systems (unlike the Markovian approach)[3] .

The description of the radio channel condition is a vector of probabilities, denoted $\overline{P}$, that a radio block is successfully received at first, second or third transmission attempt, p_1, p_2, p_3 respectively. Thus,

$$\overline{P} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_e \end{bmatrix} \tag{4.3}$$

Where p_e represents the probability the block is still errored after the third attempt. An example of the vector $\overline{P}$ is presented in figure 4.2.

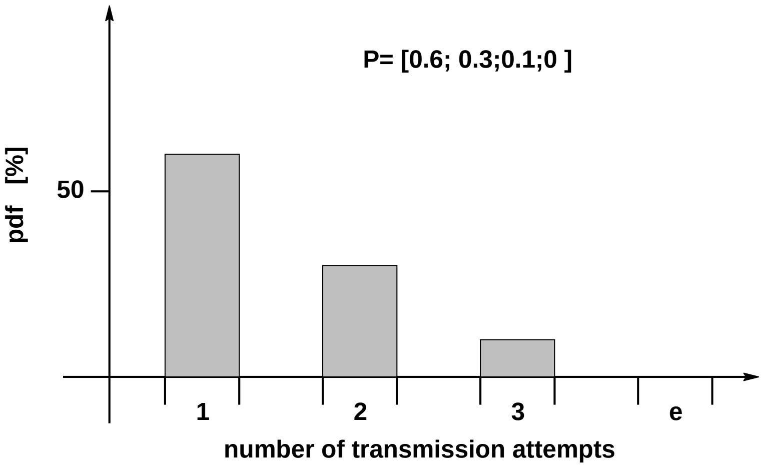


Figure 4.2: *A graphical representation of an example of $\overline{P}$ vector.*

Having the vector describing the behaviour of PHY layer defined in the proposed manner it is possible to find the probability of a radio block experiencing the system delay associated by each transmission attempt. The first system delay, δ_1, is being experienced by all radio blocks being sent, thus, $p_{\delta_1} = p_1 + p_2 + p_3 + p_e = 1$. The second system delay is experienced by all radio blocks that are retransmitted at least once, hence, $p_{\delta_2} = p_2 + p_3 + p_e$. Following that, $p_{\delta_3} = p_3 + p_e$ and $p_{\delta_e} = p_e$. These calculations mark the occupancy of particular paths on the proposed ARQ loop model, as shown in figure 4.3.

The interesting feature of the $\overline{P}$ parameters is the ability to choose the size of the time window for predicting its parameters. The predictor may be asked by different protocol layers to offer a prediction for large (e.g. 10 min), medium (e.g. 1 min) or short (e.g. 1 sec) terms, as different protocols are operating on different time bases (e.g MAC operates on the basis of ms, whereas TCP operates on the basis of seconds or several seconds).

[3] A general description of the prediction method is given chapter 6

4.2.3 Model architecture

The new architecture describing the SR-ARQ loop in mobile data systems for transporting IP packets is created by merging together both previously described models for ARQ loop delays and PHY influence on the radio block error process. The description of this architecture is illustrated in figure 4.3, and can be summarised as follows:

- IP packet size - represents a Constant Bit Rate (CBR) source of traffic, as is typical for the real-time applications under investigation here.
- $\overline{P}$ - vector describes the condition of the radio channel from a radio block viewpoint from equation 4.3.
- $\overline{\Delta}$ -vector describes the ARQ loop in terms of feedback timing from equation 4.2.

The novel aspect of this methodology is that the model of the error process of radio blocks and the ARQ feedback loop mechanism use statistical descriptors. These descriptors are in the form of vectors, $\overline{P}$ and $\overline{\Delta}$, linking together a description of the distribution of radio block retransmission and the delay of feedback for each relevant retransmission.

Based on this description, a wide range of scenarios can be analysed. It is possible to vary the IP packet size, with any statistical radio conditions for every type of timing design of ARQ feedback channel. This is all based on the assumption that the error in this feedback channel can be neglected.

4.3 Implementation

The scheme described above has been implemented within a simulation environment which generates graphs of delay as a function of IP packet size. This allows the behaviour of the ARQ loop to be investigated under different settings, representing radio conditions and ARQ design, $\overline{P}$ and $\overline{\Delta}$, respectively.

The model is created using a general purpose discrete event simulator called SES/workbench[4]. The SES/workbench has been selected because it offers intuitive, simple and accurate ways of modeling processes based on queues. The problem addressed in this thesis represents a generic approach, rather than an analysis of a detailed implementation of a particular system in which case ns2, OPNET or other network simulator would be a more appropriate choice. Its major components are presented in figure 4.3, and described now.

[4] For more information about SES/workbench please look at `www.hyperformix.com`

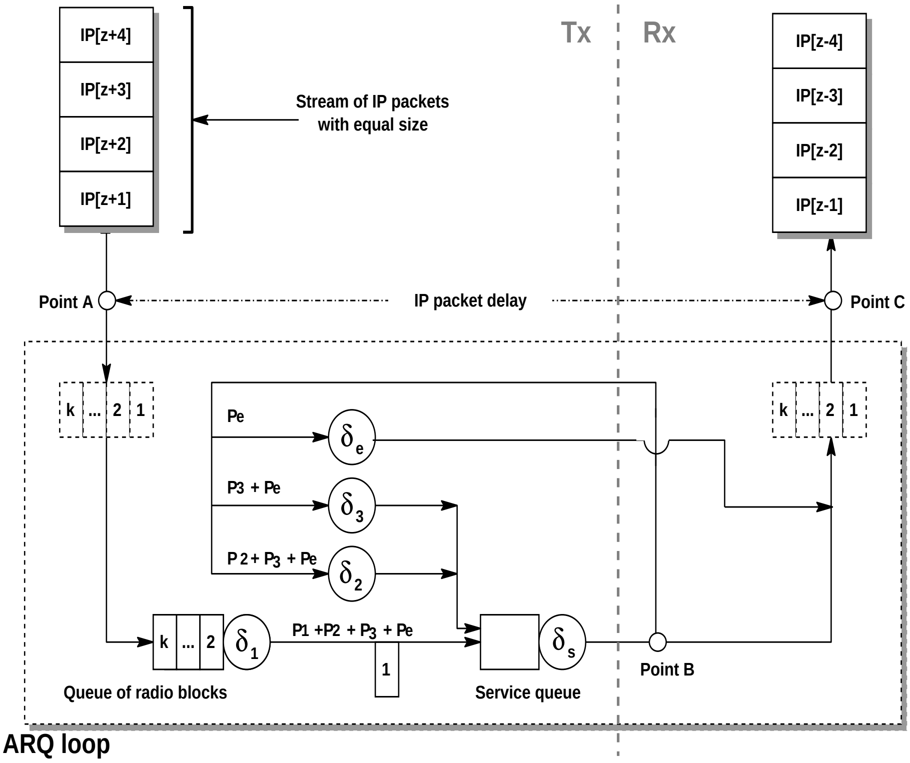


Figure 4.3: *Methodology description model*

A stream of IP packets of a certain size is generated and buffered before the ARQ loop (point A). The size of the IP packets is described by the number of radio blocks necessary to carry it, denoted k. The first IP packet in the queue is segmented into k radio blocks and each radio block is associated with the number of retransmissions it will experience. This has a value between zero and two. A value of zero represents the situation when the radio channel condition is sufficiently good that the radio block is transmitted over the wireless link successfully at the first attempt. Values of one and two represent the situations when one or two retransmissions, respectively, are necessary for successful delivery. The number of retransmissions is allocated on a statistical basis to achieve the required retransmission vector $\overline{P}$. The strong points of $\overline{P}$ vector are its ability to be predictable on the basis of a history of radio block transmission process, which is shown in more details in chapter 6, and its capability to represent the influence of combined PHY and Hybrid Type II/III ARQ techniques on the radio block error process. The i.i.d based descriptors of a radio block error process did not offer any mechanism of the Hybrid technique influence on the analysed error process, as they were described by a single number, BLER. The Markovian based methods of describing the PHY at the radio block level offer a more advanced method of illustrating the radio block error process. However, they experience a serious problem when the parameters used for specifying those models have to be predicted in a real life transmission scenarios. The description proposed here, in the form of $\overline{P}$ vector, offers a compromise between predictability and accuracy and is the only method that offers the ability of intuitive analysis of Hybrid Type II/III influence on the radio block error process. Additionally, the issues of capturing the effect of radio wave fading on the radio block error process is solved, as the statistical approach simply averages the fading influence across the time of $\overline{P}$ vector calculation.

Next, the radio blocks are sent through the radio channel denoted as $\overline{\delta_S}$, where each radio block waits the amount of time necessary to transmit a single radio block over the channel including the MAC layer delays. At point B, the decision about retransmission is taken, so that, if the number of previously defined retransmissions matches the number of retransmissions experienced then that block goes to the receiver buffer and waits for the rest of the radio blocks from that particular IP packet. If the numbers do not match, then the radio block is directed to a retransmission delay node and its retransmission counter is incremented. To model the retransmission process, the radio block is delayed at the retransmission delay nodes as specified in $\overline{\Delta}$ vector before it is sent to the service queue. The queuing policy states that priority is given to radio blocks with the highest number of retransmissions. This priority is given to minimize the time an IP packet waits in the receiver buffer. In the case of small or medium size IP packets, the radio block with the highest number of retransmissions is usually the last radio block from the considered packet. Thus, it makes sense to give the highest priority to those kinds of radio blocks, as it is quite likely that they significantly slow down the transmission of IP packets. This priority feature can be implemented in (E)GPRS or UMTS'99 types of networks. When all radio blocks reach the receiver buffer, the IP packet is reassembled and leaves the ARQ loop.

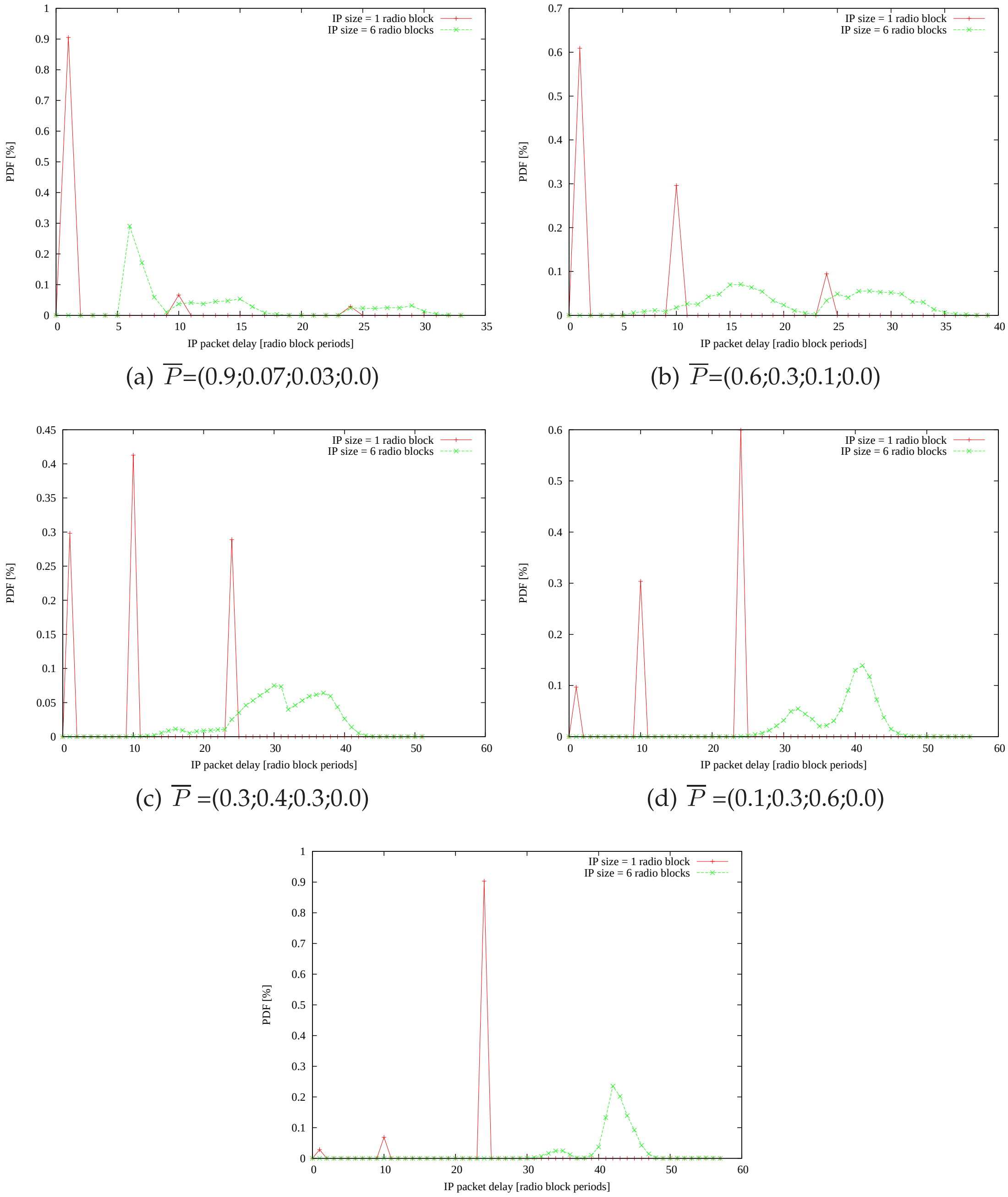
The delay of each IP packet is measured as the time between the first radio block from a particular IP packet leaving the queue of radio blocks and the time when all the radio blocks belonging to this IP packet successfully reach the receiver buffer. Since the processing time at the transmitter and receiver will be dependent on the hardware platform and the efficiency of the software implementation of these processes, these processing delays have been neglected here. The delay measured here, therefore, is the minimum expected delay of an IP packet between point A and point C, as shown in Figure 4.3.

4.4 Verification of the simulator implementation

The verification of the correct implementation of the ARQ loop simulator has been performed into two steps.

The first one involved a series of tests regarding the correct functionality of sections of the implemented ARQ loop model.

In the second step the main focus was put on the analysis of the entire system by comparing the distribution of the delay for IP packet of a size equal to one radio block with the distribution of radio block error, $\overline{P}$, for different constellations of $\overline{P}$ and $\overline{\Delta}$. It was shown that both distributions were exact in all considered scenarios. Thus, the implementation of the proposed ARQ loop model used in the rest of this thesis is correct. An example of this comparison is presented in figure 4.4[5].



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 4.4: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$.

[5] All test scenarios are presented in Appendix D

4.5 Simulation settings

The simulation model was used to determine the IP packet delay performance under different PHY and ARQ loop conditions. These simulations were performed for different values of IP packet size (k) ARQ loop delays ($\overline{\Delta}$) and radio channel conditions ($\overline{P}$).

The simulations aim to analyse the trend of the average delay of IP packet and the relation between this average value and min and max values and upper and lower 90% boundary as an indication of deviation of the analysed delay.

The other interesting aspect is the distribution of the delay experienced by IP packets of a certain size. To determine this several simulations were conducted for different IP packet sizes, under a series of different ARQ and radio channel condition scenarios. This analysis can give an insight into the shape of the delay trends, in particular, the difference between its mean and median values.

Before this can be done, it is important to consider the range of input values of IP packet size, ARQ loop delay vectors and the radio channel conditions.

4.5.1 IP Packet size

One of the main issues affecting the delay of the IP packet in a radio based system is its size. The size of an IP packet is measured in bytes (B) or bits (b). However, from a transmission point of view it is possible to measure the packet size as the number of radio blocks required to send it over an error free channel. This metric is more suitable for an analysis of SR-ARQ influence on IP packet delay. This is because at the ARQ level the internal structure of the packet is irrelevant. Instead, the relation between the Modulation and Coding Scheme (MCS) used and the actual size of the IP packet is important. Since, this relation results in the number of radio blocks required to send a particular IP packet.

Defining k as the size of an IP packet measured in radio blocks gives:

$$k = \left\lceil \frac{IP_{PacketSize}}{RB_{PayloadSize}} \right\rceil [radioblock] \tag{4.4}$$

Having defined the packet size as above, it is interesting to specify what range of IP packet sizes may be expected during standard transmission. First of all, the maximum size of an IP packet is 65,535 bytes [130]. However, due to the fact that the transmission will use TCP or UDP protocols it is very unlikely that the packet will be this large.

Nowadays, since most network traffic is transmitted through Ethernet networks at some point, the IP packet size in TCP mode will not be bigger that 1.5 kB in the majority of cases [6].

[6] There is a proposal to extend the size of MTU to up to 9kB (http://sd.wareonearth.

When UDP traffic is send over IP networks, there is no explicit limit imposed by the UDP protocol. VoIP applications usually use UDP, so that the application limits the packet size in order to be able to sustain a connection at the lowest possible bit rate.

Table 4.1 presents a comparison between the throughput offered by different modes in EGPRS and UMTS'99. As can be seen the IP packet size varies between one and eleven radio blocks for VoIP traffic, which represents a UDP connection[7]. While for a TCP oriented connection, the IP packet ranges between 4 and nearly 70 radio blocks. However, from the practical point of view it is quite unlikely that a Multimedia or VoIP transmission will be delivered through the TCP data stream in which a single segment is fragmented into a large number of radio blocks (e.g. 35 or 69). The time associated with the transmission of such a large number of radio blocks per IP packet would in most cases result in the transmission termination by the end user. Therefore, it is assumed that the transmission of the IP packet in TCP mode will be facilitated by the quality of the radio channel that will have a signal reception strong enough to use MCS5 or higher, resulting in minimizing the size of IP packet to about 30 radio blocks. Hence, the IP packet size being considered in this work is between 1 and 30 radio blocks. The ARQ loop analysis method proposed here will work fine as well in the case of IP packets larger than 30 radio blocks. The IP packet is limited only for the sake of minimizing the simulation time but in a fashion that will ensure that the analysed IP packet size will cover the majority of the currently expected VoIP and Multimedia IP packet sizes.

4.5.2 SR-ARQ Delay Vector

The influence of the ARQ loop is described in the form of a vector that stores the average delays associated with each particular transmission attempts, denoted $\overline{\Delta}$. These delays are caused by the interaction between the ARQ system and lower protocol layers (mainly altered by the PHY condition). Thus, the $\overline{\Delta}$ vector describes the design of the SR-ARQ loop, as it stores the average delays introduced at transmission attempts.

An example of such a vector is $\overline{\Delta} = [0; 8; 13; 0]^T$, where it is assumed that the Transmission Window N_{poll}, is set to 10 radio blocks[9].

- The first parameter of the vector, zero, corresponds to the first transmission of a radio block. A value of zero means the first radio block transmission

com/~phil/jumbo.html). However, this is not widely used in the Internet nowadays. Additionally, it is shown later in this chapter that the size of analysed IP packet does not limit the proposed ARQ loop analysis methodology

[7] The VoIP packet payload given in Table 4.1 includes the entire overhead necessary for the analysed type of connection

[9] $N_{poll} = 10$ radio block periods is the most common value used in (E)GPRS systems.

	Throughput	*Radio Block*	*VoIP packet*					*TCP segment*
	at RLC	*Payload* [8]	[bytes]					[bytes]
	[kbps]	[bits]	80	120	160	200	240	1500
UMTS'99								
High throug.	348	3480	1	1	1	1	1	4
Low throug.	64	640	1	2	2	3	3	19
EGPRS								
MCS 9	59.2	1184	1	1	2	2	2	11
MCS 8	54.4	1088	1	1	2	2	2	12
MCS 7	44.8	896	1	2	2	2	3	14
MCS 6	29.6	592	2	2	3	3	4	21
MCS 5	22.4	448	2	3	3	4	5	27
MCS 4	17.6	352	2	3	4	5	6	35
MCS 3	14.8	296	3	4	5	6	7	41
MCS 2	11.2	224	3	5	6	8	9	54
MCS 1	8.8	176	4	6	8	10	11	69

Table 4.1: *Number of radio blocks required for a VoIP packet and a TCP segment at UMTS'99 and EGPRS radio interfaces*

experiences only the delay caused by queuing in the RLC layer and the transmission delay is one block period.

- The second parameter, eight, represents the average time necessary for the ARQ feedback for the completion of the first radio block retransmission. It is composed of five radio block periods representing the average occupancy of the polling buffer at the transmitter[10] (N_{poll} /2), one block period for creating the retransmission request message regarding these radio blocks that were corrupted during the transmission in the previously finished Transmission Window[11]. Another block period is required for the transmission of the retransmission request over the uplink[12] and finally one block period is added to include the processing time at the transmitter[13], in total eight radio block periods, as shown in figure 4.5.

- The third parameter, thirteen, represents the time delay due to the feedback delay for the second retransmission. It is similar to the first retransmission, except that this block was errored during the last polling period and has been inserted at the head of the transmission queue. It will, therefore, be the first errored block in the polling buffer at the receiver, so that its polling buffer delay will be equal to the full length of that queue, rather than its mean occupancy.

[10] Look at the **ARQ Transmission window time** in the figure 4.5.

[11] Look at the **Computation Delay at the receiver** in the figure 4.5.

[12] Look at the **Transmission Delay** at the receiver side - in the figure 4.5.

[13] Look at the **Computation Delay at the transmitter** in the figure 4.5.

- The last parameter, zero, represents the delay due to an error after the second retransmission. A value of zero in the last position indicates that there is no delay due to the errorred radio blocks after the second retransmission.

Another set of values for $\overline{\Delta}$ vector, used for further simulation tests, can be computed in the same manner as described above. Different values of N_{Poll} and delay at the Transmitter and Receiver are combined resulting in new values for $\overline{\Delta}$ vector. The complete set of these ARQ delay vectors used in this thesis is presented in Table 4.2.

N_{Poll}	Computation Delay at the Transceiver or Receiver				
[radio block periods]	[radio blockperiods]				
-	1	2	3	4	5
10	$\begin{bmatrix}0\\8\\13\\0\end{bmatrix}$	$\begin{bmatrix}0\\10\\15\\0\end{bmatrix}$	$\begin{bmatrix}0\\12\\17\\0\end{bmatrix}$	$\begin{bmatrix}0\\14\\19\\0\end{bmatrix}$	$\begin{bmatrix}0\\16\\21\\0\end{bmatrix}$
20	$\begin{bmatrix}0\\13\\23\\0\end{bmatrix}$	$\begin{bmatrix}0\\15\\25\\0\end{bmatrix}$	$\begin{bmatrix}0\\17\\27\\0\end{bmatrix}$	$\begin{bmatrix}0\\19\\29\\0\end{bmatrix}$	$\begin{bmatrix}0\\21\\31\\0\end{bmatrix}$
30	$\begin{bmatrix}0\\18\\33\\0\end{bmatrix}$	$\begin{bmatrix}0\\20\\35\\0\end{bmatrix}$	$\begin{bmatrix}0\\22\\37\\0\end{bmatrix}$	$\begin{bmatrix}0\\24\\39\\0\end{bmatrix}$	$\begin{bmatrix}0\\26\\41\\0\end{bmatrix}$

Table 4.2: *Complete set of different ARQ loop delay vector settings, $\overline{\Delta}$, considered in the simulations presented in the following section*

The total number of different delay vectors considered for the simulations in section 4.6 is 15. This number covers a variety of different scenarios where both N_{Poll} and delay at the Transmitter and Receiver are varied. This provides a good overview of ARQ loop influence on the IP packet delay characteristic.

The minimum N_{Poll} size is 10 radio block time periods, the minimum retransmission request transmission time is 1 radio block time period and the minimum processing time at both the Transmitter and the Receiver is 1 radio block time period. Consequently, the minimum retransmission delay associated with the first retransmission attempt is $10/2+1+1 = 8$[14] radio block periods and with the second retransmission attempt is $10+1+1 = 13$ radio block periods. This is reflected in the first proposed $\overline{\Delta}$ equal to $[0;8;13;0]^T$. All other $\overline{\Delta}$s have greater delays in order to check the behaviour of the proposed analysis methodology for a variety of ARQ feedback loop delay settings. All of these 15 scenarios do not represent all possibilities of $\overline{\Delta}$ vector settings. However, they represent a large number

[14] The N_{Poll} is divided by 2, as this gives the average queue ocupancy of the N_{Poll} size.

of scenarios that in the case of wireless systems should cover the minimum and maximum delay vectors expected to be found in real mobile data networks[15].

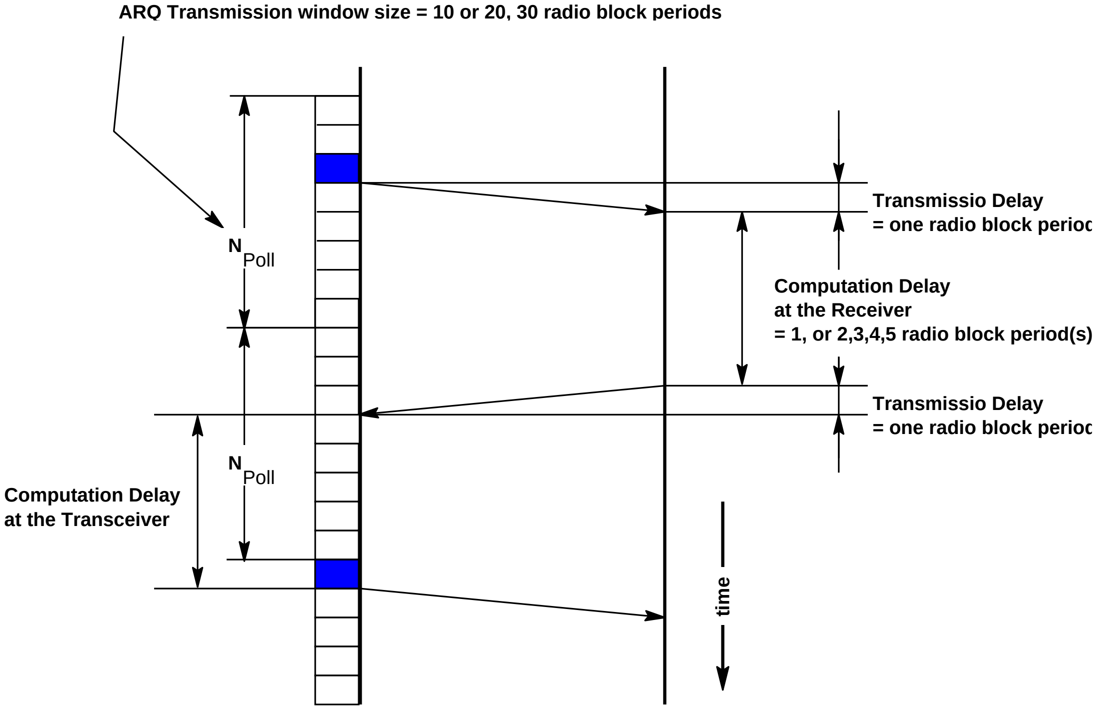


Figure 4.5: *Different contributions to the Δ vector*

4.5.3 Radio Channel Vector

The new model proposed in this work models the interference of the PHY channel as a vector of transmission attempts. Thus, a good channel will be represented by the majority of radio blocks being required for only single transmission in order to get through RLC layer successfully. Whereas, bad radio conditions will be represented by a vector where a significant number of radio blocks will need two or three transmission attempts to successfully reach the receiver.

It is very hard to obtain real system data for the $\overline{P}$ vector, as it requires access to working equipment and appropriate software. Hence, in this work this kind of data is not present.

Tests based on the data coming from a particular system can represent only this system and the radio and hardware environment it works in. Thus, such a test would be single system oriented. Hence, the value of $\overline{P}$ vector will be prepared in a more abstract way, which will include different possible radio conditions. Additionally, some extreme (and possibly impossible from the point of view of current modulation/detection and coding field state of the art) tests will be added, to show the behaviour of the ARQ loop in very different scenarios.

[15] The Mobile Data Network are design in the way that aim to minimize the size of $\overline{\Delta}$ vectors

These extreme cases may not be realistic at the moment, although, with progress in modulation/detection and coding they may be realistic in the future.

Following the principles described above, five different radio channel conditions are proposed as input values for the tests conducted in the next section. The first case, called P_0, represents a situation where only 10 percent of radio blocks have to be retransmitted, while cases P_1, P_2, P_3 and P_4 represent worsening retransmission scenarios.

P_0: $\overline{P} = [0.9; 0.07; 0.03; 0.0]^T$, which means that we assume that 90 percent of radio blocks will reach the receiver at the first transmission attempt, 7 percent at the second attempt, and 3 percent at the third attempt. The last position is zero and it indicates that in this scenario there are no errors experienced by any of the radio blocks after the second retransmission. The other cases, P_1 to P_4, can be interpreted in the same way.

P_1: - $\overline{P} = [0.6; 0.3; 0.1; 0.0]^T$

P_2: - $\overline{P} = [0.3; 0.4; 0.3; 0.0]^T$

P_3: - $\overline{P} = [0.1; 0.3; 0.6; 0.0]^T$

P_4: - $\overline{P} = [0.03; 0.07; 0.9; 0.0]^T$

The cases P_3 and P_4 represent less likely scenarios, as they need mostly three transmission attempts to successfully send a radio block through the RLC layer. Nonetheless, their presence illustrates the behaviour of the ARQ loop in extreme situations.

4.6 Results

In this section the results of simulations for all the combinations of IP packet size (k) radio channel condition ($\overline{P}$) and ARQ feedback loop delays ($\overline{\Delta}$) are analysed. Only a selected set will be presented in this section to illustrate the aforementioned phenomena, while the full results are available in the Appendices B, C, D and E.

This analysis aims to study the pattern of an average IP packet delay as a function of its size for different settings of $\overline{P}$ and $\overline{\Delta}$. It also focuses on the accuracy of this average value, when used as a descriptor of the delay characteristic of the RLC layer. Finally, the aspects of difference/similarity between the mean and median values for selected IP packet sizes will be discussed.

4.6.1 Average IP packet delay

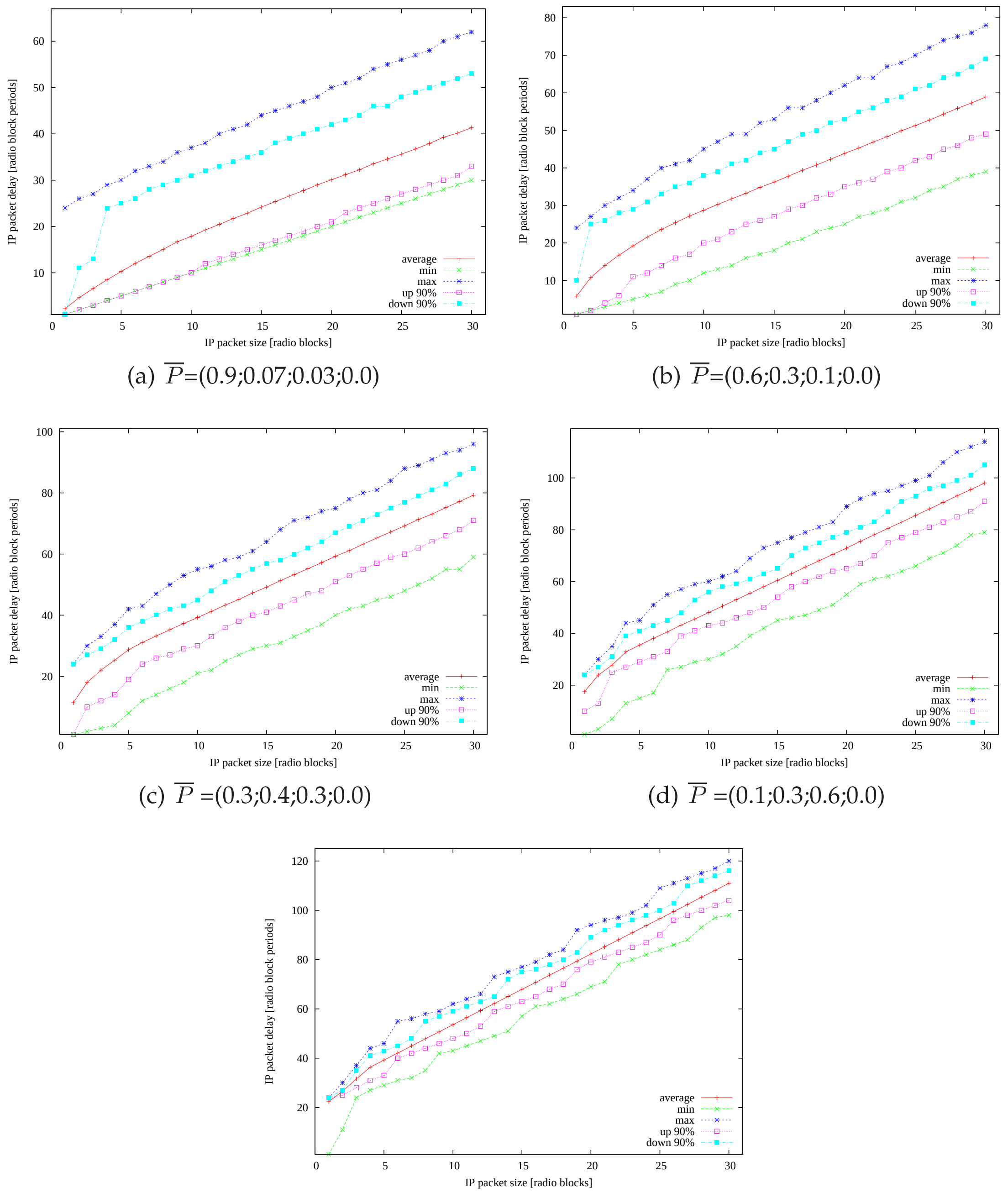
Here the average IP packet delay will be compared to patterns of minimum and maximum values of delay experienced by a packet of a certain size, k. Additionally, two trends exposing upper and lower 90% result bounds are plotted. These additional lines illustrate how far most of the results are spread from the mean value, for a given packet size within an analysed $\overline{P}$ and $\overline{\Delta}$.

An example set of results is shown in figure 4.6, which reflects the general trend of the full results. This means that the average value has approximately the same slope as the minimum, maximum, upper 90% and lower 90% results. Additionally, it is placed centrally between the maximum and minimum curves, which is a good indication of symmetry between average and minimum, maximum, upper and lower 90% bounds. This symmetry is more visible in the case of medium and large packet sizes. Thus, it exposes the fact that in the case of small IP packet size the validity of using an average value as a descriptor of the RLC delay component has to be investigated further. Although, the general conclusion from all the different simulation settings is that the average value can be considered a reasonable delay descriptor at the RLC layer.

To further analyse the results the uncertainty is calculated as the percentage difference between the average value and maximum, minimum, upper and lower 90% bounds as shown in equation 4.5.

$$\begin{array}{r}Uncertainty_{maximum}(k) = \frac{|average(k)-maximum(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{minimum}(k) = \frac{|average(k)-minimum(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{upper90\%}(k) = \frac{|average(k)-upper90\%(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{lower90\%}(k) = \frac{|average(k)-lower90\%(k)|}{average(k)} \cdot 100\%\end{array} \tag{4.5}$$

These calculations have been performed for all scenarios and the relevant graphs are presented in the Appendix C. A typical example of five different radio channel scenarios for a given ARQ loop delay,$\overline{\Delta} = [0; 8; 13; 0]$, is given in figure 4.7. These graphs indicate that the uncertainty goes below 50% when the IP packet size is medium or large. Whereas, in the case of small packet size this can reach quite a significant value of 1000 % error between average and maximum. Therefore, it is adviseable to treat the average value of the IP packet delay for small packet sizes with caution due to the high uncertainty it can introduce. Nonetheless, it is important to notice that the upper 90% group performs very well and in most cases keeps its uncertainity level below 25%. This means that the majority of IP packets transmitted in a considered scenario of k, $\overline{\Delta} = [0; 8; 13; 0]$ and $\overline{P}$ have experienced a delay lower than average value + 25% error margin. This is especially true in case of medium and large IP packet sizes.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

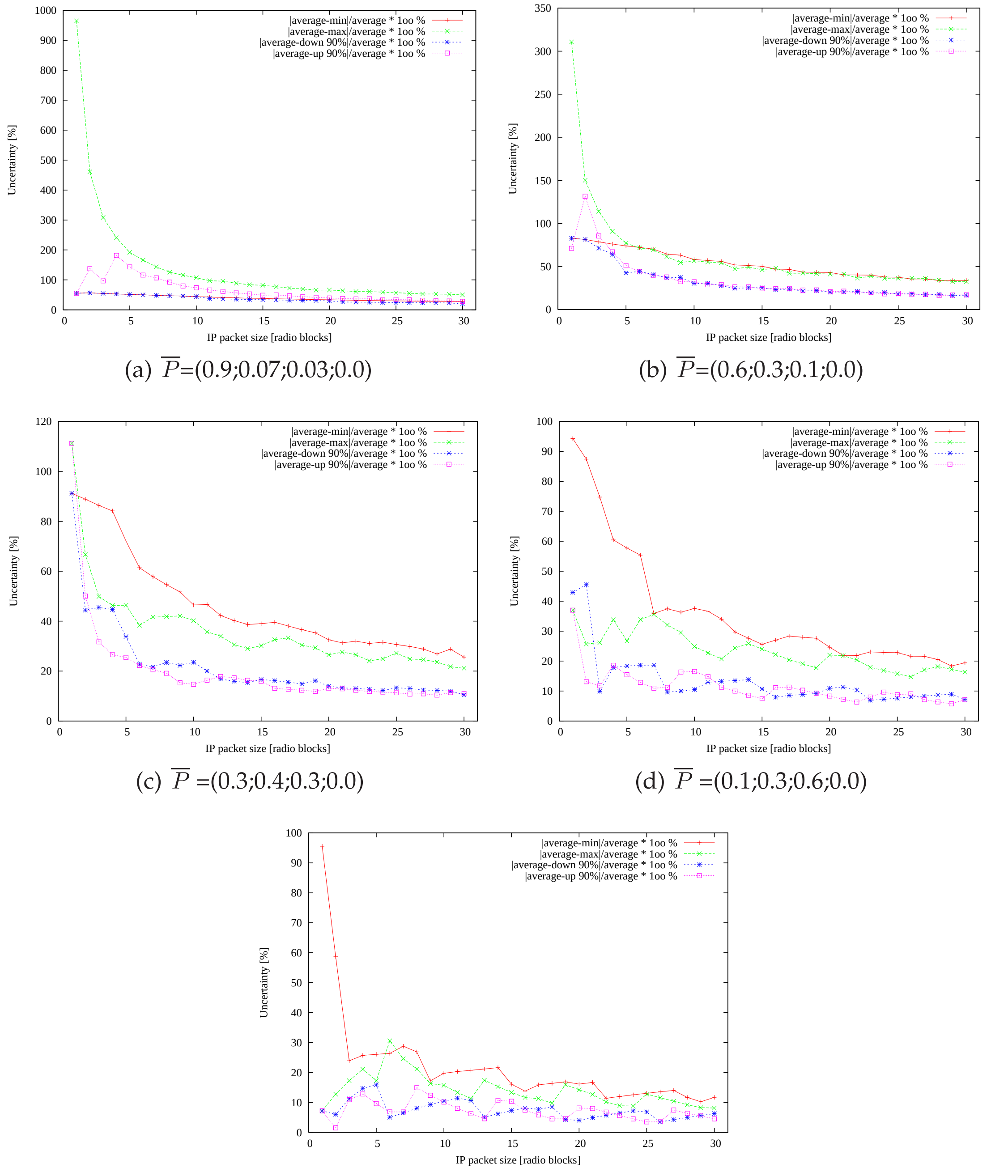
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 4.6: *Average IP packet delay and min, max, upper 90% anl lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 4.7: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0;8;13;0]$ *and* $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$.

4.6.2 IP packet delay distribution

In the previous section the main emphasis was on the level of deviation between the value of the average IP packet delay and majority of the packet delay experienced by transmitted packets. This information is useful when the variation of the analysed process is tested. However, quite often it is assumed that the shape of the process represented for a given IP packet size is of the Gaussian type or at least that it is symmetrical. This assumption is not always true and it is worth investigating when it is a realistic assumption and what kind of consequences it imposes.

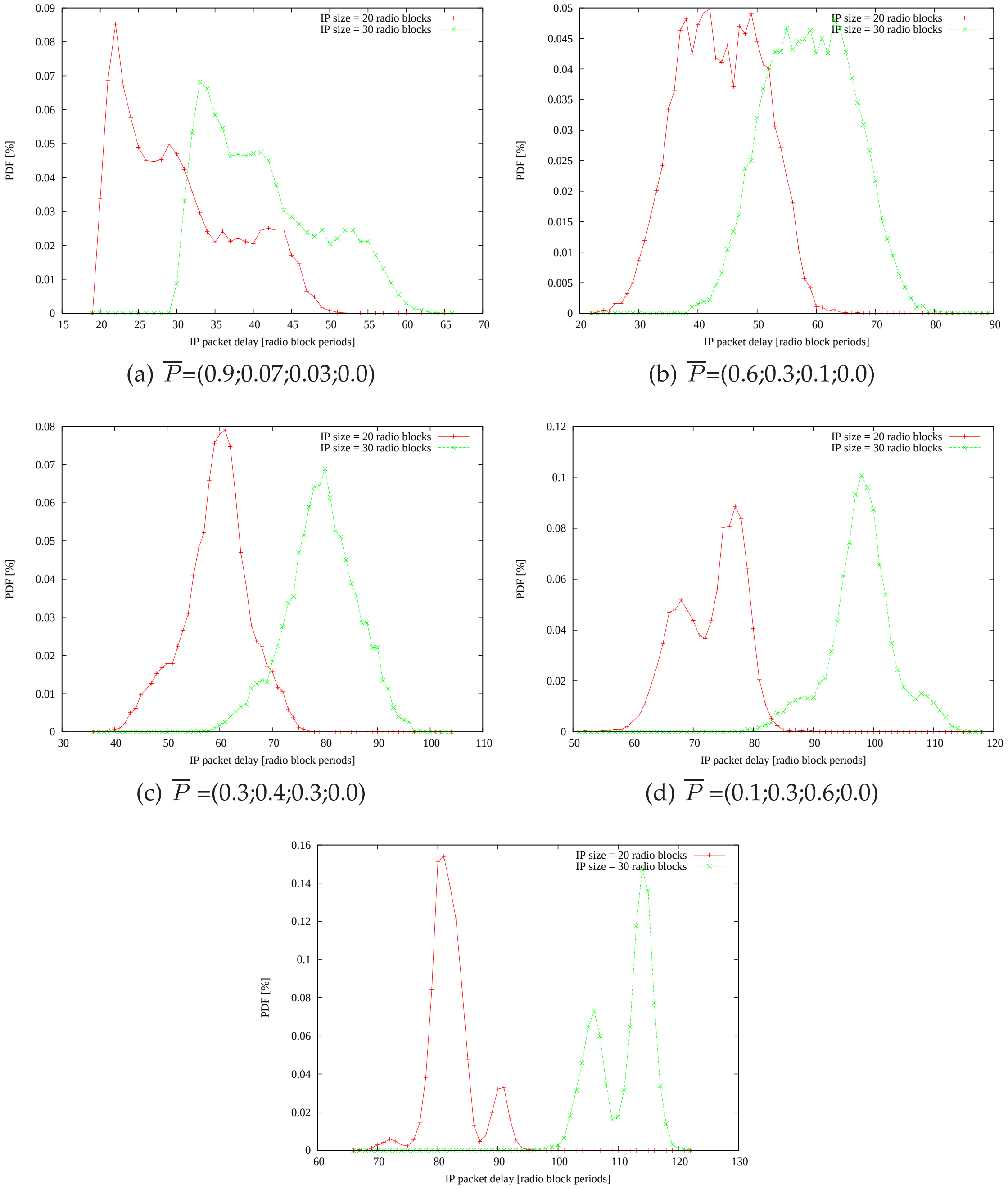
A set of simulations was run, generating graphs that show the distribution of the delay experienced for a transmission of IP packet size of one, six, twenty and thirty radio blocks. These have classified into two groups, small packets of one and six radio blocks and large packets of twenty and thirty radio blocks. These graphs are included in the Appendix D and Appendix E, for the purpose of general discussion two sets of graphs are presented in figure 4.4 and figure 4.8.

As can be seen in figure 4.4 the delay of packets that are one radio block long directly matches the shape of the radio block error characteristic,$\overline{P}$. Increasing the packet size smooths these peaks, but the curves are highly irregular. The mean value is often placed in regions where the probability of experiencing the delay by the transmitted IP packet is extremely low and sometimes approximately equal to zero. Thus, relying on the average value in such cases should always be treated with a high awareness that delay will have a large variance.

In contrast to the previously analysed delay distributions, the distributions for large IP packets exhibit a significant number of symmetrical shape, figure 4.8. Additionally, quite a good number of these distributions represent a Gaussian like shape. This makes the use of average delay value much more reasonable as the generic description of the RLC delay characteristic - for medium and large IP packet sizes.

For small IP packets, however, the average must be treated with caution as the highly asymmetric and non Gaussian delay distribution shapes shown previously may be misinterpretted. Thus, the influence of these properties of the IP packet delay distributions requires further investigation.

An investigation of the ARQ component of IP packet delay distribution can be achieved by looking at the dynamic behaviour introduced by the ARQ loop. It is quite common that in the case of Multimedia and VoIP applications there is an adaptation mechanism that controls the size of the play out buffer. By doing so, it controls the maximum acceptable delay of IP packets. Once the packet is delivered beyond this delay threshold it is discarded and considered as lost. The analysis of ARQ influence on this mechanism could give a new perspective on the importance of ARQ interaction with higher protocol layers.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 4.8: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$.

To achieve this it is important to assume that all other delay components of the IP packet delay are static, then any dynamism introduced by ARQ loop will be transferred directly into the mechanism of buffer adaptation. Having that assumption in place, it is possible to draw figures describing the behaviour of the delay distribution for different radio channel conditions. Two figures presenting the behaviour of small and large IP packet show this phenomenon, see figures 4.9 (a) and (b) respectively.

It is visible that the buffer adaptation mechanism would experience a linear progress of losses when decreasing the size of the play out buffer. This is exactly the behaviour which the adaptation mechanism is expected to experience.

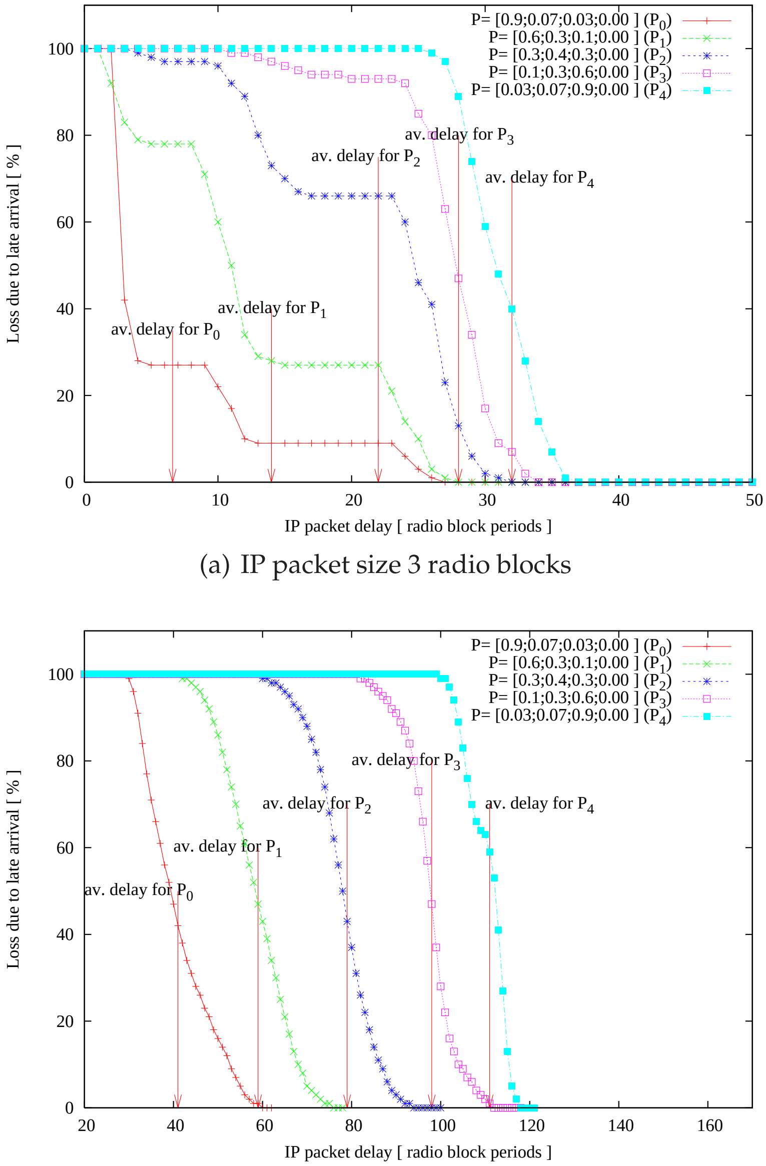
On the contrary, in case of small IP packets this relation is not linear in the entire delay domain. There are places, denoted here as zero benefit zones, which with the increase of acceptable IP packet delay return zero loss performance improvement. This is caused by the effect of the ARQ feedback loop delay. I in the case of small IP packets, the delay associated with the ARQ feedback is larger than delay caused by slowing IP packet transmission originated from radio block retransmissions. This phenomenon is quite important as the adaptation mechanisms at the Application layer are not aware of this nonlinear behaviour. Therefore, it is important to pass the message that would make the adaptation algorithms aware of this issue, if such cross layer communication is possible.

Once could say the the use of average delay component in case of small IP packets is highly un-recommended, considering the high level of uncertainty. However, it is important that the influence of ARQ loop on both average and distribution of IP packet delay can be quite profound. Thus, having an indicator of this performance is always better that being unaware of its existence. The potential usage of this descriptor can be implemented at different layers of the protocol stack. For example, a Multimedia application can adjust the size of its data packets accordingly to expected RLC delay performance. The overall throughput, affecting the quality of the transmitted content, can be kept on the same level by increasing the rate of releasing these packets.

4.7 Summary

This chapter presented a new method of analysing the influence of ARQ methods on the IP packet delay. A number of extensive simulations have been performed to determine how reliable the average value is as the indicator of the RLC delay performance.

The proposed method is characterised by the ability to capture the influence of Hybrid Type II and III ARQ loop on IP packet delay. Additionally, it considers the mechanism of re-sequencing radio blocks with respect to the priority mechanism



(a) IP packet size 3 radio blocks

(b) IP packet size 30 radio blocks

Figure 4.9: *IP packet loss caused by its late arrival as a function of Maximum acceptable packet delay at RLC level - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$.

based on the number of already performed transmission attempts. The proposed method uses input parameters that are easy to obtain in a real system, making it applicable to systems that exist in the real world.

The tests performed on the proposed model revealed that the average value can represent the influence of the ARQ based system on IP packet delay for medium and large sized packets. The variance and shape of the delay characteristic are such that the mean delay can be used as a good performance indicator.

The situation looks different in the case of small IP packets, where the analysed delay often has a highly non-Gaussian and non-uniform distribution. This means that the average value can not be fully trusted as the delay descriptor of the RLC layer. This is a result of high differences between mean and median values of distributions of a particular IP packet size.

Following that, the ARQ component of IP packet delay distribution has been analysed. It has been shown that in the case of large IP packets the dynamic behaviour of ARQ loop does not introduce unexpected behaviour. Hence, it does not interfere with adaptation mechanism at the higher layers. When small packets are transmitted, the ARQ loop introduces a non-linear component to the relationship between loss and maximum acceptable delay. This may confuse adaptation mechanisms at the Multimedia or VoIP application so it is essential to send an appropriate message to this application that makes it aware of the potential problems, if a cross layer communication is possible.

CHAPTER 5

Novel Delay Prediction Technique

In this chapter two new methods of predicting the ARQ loop component of the average IP packet delay are presented. These methods are Brute Force (BF) and Low Computation Complexity (LCC). The proposed algorithms are examined with respect of the difference between the results from simulations and analytical BF and LCC calculations. The simulation results come from the implementation of the ARQ loop analysis methodology proposed in this thesis. Finally, on the basis of the accuracy and computation complexity, one of these algorithms is selected as a candidate for a deployment in a real network.

5.1 Introduction

The last chapter presented a new methodology for analysing the BEC influence on IP packet delay. The input variables can be readily gathered from an existing system. The PHY influence is modelled in a fashion that enables it to be used not only for a posteriori analysis, like the complex Markov chain models, but also for the analysis of expected performance. Combining these factors together allows for a mathematical model to be developed that could predict average expected IP packet delay as a function of its size (in radio blocks) , k , radio channel condition, represented by $\overline{P}$, and ARQ feedback delays, $\overline{\Delta}$.

An obvious approach is to first carefully analyse the entire process of IP packet transmission and then model this process with mathematical equations. Once this is done, the compound mathematical description can then be used to estimate the expected delay.

Since the prediction method developed in this chapter is based on the model presented in the previous chapter, it follows the same assumptions as that model. In addition to this, an assumption about the radio channel behaviour has to be made. It is assumed that the radio channel influence during the transmission of a single IP packet is stochastic, meaning that the vector describing this channel, $\overline{P}$, does not change significantly during transmission. Moreover, it is assumed that there is an external predictor which delivers the estimation of the radio channel condition for the following transmission, $\widehat{\overline{P}[z+1]}$. One such solution is the idea that the estimation of the next radio channel condition is equal to conditions experienced in the last IP packet transmission, $\widehat{\overline{P}[z+1]} = \overline{P}[z]$[1].

It is important to state that in the rest of this chapter it is assumed that a perfect predictor is used, meaning that $\widehat{\overline{P}[z+1]} = \overline{P}[z+1]$. Hence, when the accuracy of the proposed prediction method is analysed, it takes into account only the imperfection of the prediction algorithms. The issue of finding the most accurate predictor of the $\overline{P}$ vector is beyond the scope of this thesis and will not be addressed here.

5.2 Brute Force approach

The name of this technique implies that the computation complexity involved in this approach is not considered as an important issue. However, because the process of sending an IP packet is quite intensive, this method is actually characterised by high computation complexity.

In this approach, the process of sending an IP packet through the model is going to be investigated first. Next the mathematical description of this process will be proposed with the expectation that this rigorous description will yield a solution to the problem of predicting the delay of IP packet transmitted over the system.

5.2.1 Flow of events associated with the transport of IP packet

The flow of the events can be captured easily by looking at figure 4.3. First, the $z'th$ IP packet is fragmented into a number of radio blocks, according to the fragmentation process described in Chapter 2 . As a result of this fragmentation a number of radio blocks, k is created which need to be transmitted successfully, for the IP packet in question, $IP[z]$, to be considered as received correctly.

Secondly, each of these radio blocks can have three transmission attempts. The number of attempts rely on the condition of the radio channel, $\overline{P}[z]$. Obviously, if $\overline{P}$ represents a channel with a high probability of successful transmission, most

[1] $\overline{P}[z]$ represents the value of the $\overline{P}$ for the $z'th$ IP packet. However, the estimation of this value, $\widehat{\overline{P}[z]}$, can be calculated using a few previously transmitted packets, $z-1, z-2, z-3, ...$. (e.g. Exponentially Weighted Moving-Average (EWMA) algorithm can be used to calculate the $\widehat{\overline{P}[z]}$

of $IP[z]$'s radio blocks will experience only one transmission attempt.

Therefore, if the IP packet consists of k radio blocks and each of them can have three transmission attempts, the total number of possible transmission scenarios is 3^k. Each of these combinations is associated with a certain pattern of the number of transmission attempts of each particular radio block being a part of the analysed IP packet.

The probability of a certain pattern happening and the total IP packet delay associated with this pattern is closely linked to the average IP packet delay.

It is important to remember that each transmission attempt consumes one time slot from the radio resource. Thus, the radio block that is experiencing two retransmissions slows the whole transmission by two timeslots in relation to the ideal case where no retransmission is needed.

Additionally, the priority of radio blocks that have experienced the highest number of transmission attempts should be considered. The higher priority given to those radio blocks that have experienced the higher number of transmission attempts minimise the time of average IP packet delays. All IP packets require all radio blocks carrying this particular IP packet to be delivered successfully. Thus, by giving the fastest access to the radio resources to those radio blocks that already spent a lot of time (due to retransmission) in the ARQ loop, the average time of delivery of all radio blocks being a part of the particular IP packet is shorter.

Finally, in the case of more that one radio block being a part of analysed IP packet the position of radio blocks in the transmission queue is important. That position determines the minimum time which the analysed radio block needs to spend in the queue before it can be scheduled to be sent over the ARQ loop. Thus, the radio block that is first in the queue has zero delay associated with the time spent in the transmission queue. Whereas the last radio block of a particular IP packet has to wait until all other radio blocks being a part of this IP packet are sent to the ARQ loop. Thus, this position phenomenon needs to be taken into account as well.

5.2.2 Mathematical description of these events

Each of the factors involved is discussed and specified in a more rigorous mathematical fashion. This will allow an algorithm to be specified for prediction of the average delay of the transmitted packet.

5.2.2.1 Representation of all possible scenarios

The mathematical representation of all possible scenarios is a matrix, named $\mathbb{R}(k)$. Each row, denoted by 'i', represents one possible transmission scenario, the columns, denoted 'j', show the order of radio blocks in the transmission buffer. Each element takes the value of the number of transmission attempts associated with the specific radio block within each scenario. Hence the matrix has the following form:

$$\mathbb{R}(k) = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,k} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ r_{3^k,1} & r_{3^k,2} & \cdots & r_{3^k,k} \end{bmatrix} \tag{5.1}$$

Where:

- $r_{i,j}$ represents the number of transmission attempts experienced by j^{th} radio block of the i^{th} possible retransmission scenario. In the case of a retransmission free scenario, all $r_{i,j}$ of a particular 'i' will be equal to one. For example, if an IP packet consists of four radio blocks and the particular i^{th} transmission of this packet is experiencing the following radio block pattern: first radio block 2 transmission attempts, second radio block 1 transmission attempts, third radio block 2 transmission attempts and the fourth radio block 3 transmission attempts, then the components of matrix $\mathbb{R}(k)$ will have the following values: $r_{i,1} = 2$, $r_{i,2} = 1$, $r_{i,3} = 2$, $r_{i,4} = 3$.

- $i \in \{1, 2, 3, \cdots, 3^k\}$ represents one of the 3^k possible scenarios of transmitting an IP packet which has a size of k radio blocks.

- $j \in \{1, 2, 3, \cdots, k\}$ represents the position of the radio block in the transmission queue, in the i^{th} transmission scenario.

5.2.2.2 Probability of different scenarios

Since the matrix $\mathbb{R}(k)$ stores all possible 3^k scenarios of IP packet transmission (represented by rows of this matrix), it is possible to calculate the probability of each of this scenario to happen. The probability of the i^{th} transmission scenario occurring is the product of the probabilities of successful transmission of all its constituent radio blocks. The results of these calculations for each individual i^{th} IP packet transmission scenario are then stored in a vector denoted by $\overline{S}(k)$, which is 3^k elements large.

The probabilities of successful transmissions of a radio block at a particular attempt is stored in the vector $\overline{P}$, which is one of the initial parameters. However, these probabilities have to be linked appropriately with the status of a particular radio block. Therefore, the matrix $\mathbb{R}(k)$, is used to determined the number of transmission attempts associated with this particular radio block. Consequently, each element of $\mathbb{R}(k)$ is an index of an element of vector $\overline{P}$. Hence, $\overline{S}(k)$ has the following form:

$$\overline{S}(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_{3^k}(k) \end{bmatrix} \tag{5.2}$$

Where:

- $s_i(k) = \prod_{j=1}^{k} p_{r_{i,j}}$ represents the probability that i^{th} constellation of radio blocks will occur during the transmission of an IP packet, whose size is k radio blocks.
- values of $p_{r_{i,j}}$ are taken from vector $\overline{P}$.
- values of $r_{i,j}$ elements (which are indexes of p elements) are taken from matrix $\mathbb{R}(k)$.

5.2.2.3 Delay associated with different scenarios

When the IP packet is fragmented into radio blocks, these radio blocks are sent to the transmission buffer where they wait for access to the radio channel. The scale of the associated queuing delay depends on the position of a radio block in the transmission buffer, as the last radio block has to wait at least $k-1$ time-slots to be transmitted when the first radio block experiences zero delay. Also, the retransmission of errored blocks will not be performed instantaneously, and the radio block will experience some additional delay related to this phenomena. Finally, the radio block with a larger number of transmission attempts has a higher priority in obtaining access to the radio resources, compared to a radio block with a lower number of transmission attempts. This priority mechanism prevents dead lock situations from happening, yet, it postpones the transmission of all radio blocks queued in the transmission buffer. These three phenomena have different effects on the total delay of the transmitted IP packet and can also affect the delay of subsequence IP packets

Starting with the representation of the total delay, since all possible scenarios have to be considered, the natural representation of the total delay is a vector that stores delays for every possible constellation of k radio blocks. The vector size is 3^k and the row index 'i' indicates a particular scenario. Thus,

$$\overline{\Delta}_{Total}(k) = \begin{bmatrix} \delta_1^{Total}(k) \\ \delta_2^{Total}(k) \\ \vdots \\ \delta_{3^k}^{Total}(k) \end{bmatrix} \tag{5.3}$$

Where:

- $\delta_i^{Total}(k)$ represents the delay of an IP packet, of size k radio blocks, when its radio blocks have experienced the transmission attempt scenario stored the i^{th} row of the $\mathbb{R}$ matrix.

The radio blocks queued in the transmission buffer experience a delay that is determined by the position of a particular block in the queue. Consequently, the delay associated with this phenomena, denoted by $\delta_{i,j}^s$, is equal to the index of the radio block decremented by 1, since the delay of the first radio block is zero. Hence, $\delta_{i,j}^s = j - 1$.

The process of requesting the retransmission of errored radio blocks takes time, and this time delay has to be taken into the account as well. The data to model this problem is stored in the vector $\overline{\Delta}$. If a radio block is transmitted once, then it experiences a delay equal to $\delta_1^a = \delta_1 + \delta_s$, where δ_s represents the time necessary for a successful transmission across the medium. This time is equal to one, and any other delay is expressed as a multiple of it. For a higher number of transmission attempts the situation looks like:

$$\begin{aligned} \delta_2^a &= \delta_1^a + \delta_2 + \delta_s \\ \delta_3^a &= \delta_2^a + \delta_3 + \delta_s \\ \delta_e^a &= \delta_3^a + \delta_e \end{aligned} \tag{5.4}$$

The vector $\overline{\Delta}_a$ therefore represents the accumulated influence of the retransmission delay on the radio blocks being transmitted once, twice, etc...

$$\overline{\Delta}_a = \begin{bmatrix} \delta_1^a \\ \delta_2^a \\ \delta_3^a \\ \delta_e^a \end{bmatrix} \tag{5.5}$$

Using this vector, it is easy to find the delay of radio blocks caused by non-instantaneous retransmission. For a particular transmission scenario, indicated by 'i', and a particular radio block labelled 'j', we can find the associated number of retransmissions, $r_{i,j}$. Hence, by linking the retransmission number with the relevant delay associated with it, we can obtain the retransmission influence on the delay of the considered radio block. Thus, $\delta_{i,j}^r = \delta_{r_{i,j}}^a$.

To combine the influence of these two phenomena on the total IP packet delay, a matrix is created, denoted by $\mathbb{D}_t$, representing the sum of these delays. The elements of this matrix are $\delta^t_{i,j} = \delta^s_{i,j} + \delta^r_{i,j}$.

$$\mathbb{D}_t = \begin{bmatrix} \delta^t_{1,1} & \delta^t_{1,2} & \cdots & \delta^t_{1,k} \\ \delta^t_{2,1} & \delta^t_{2,2} & \cdots & \delta^t_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \delta^t_{3^k,1} & \delta^t_{3^k,2} & \cdots & \delta^t_{3^k,k} \end{bmatrix} \tag{5.6}$$

The last radio block to reach the receiver effectively determines the total delay of a particular IP packet. Thus, each row of the matrix $\mathbb{D}_t$ is searched for the largest element. This element represents the delay of the last radio block related to the i^{th} transmission scenario. The vector $\overline{\Delta}_{Max-row-t}(k)$ stores these delays.

$$\overline{\Delta}_{Max-row-t}(k) = \begin{bmatrix} \delta^{Max-row-t}_1(k) \\ \delta^{Max-row-t}_2(k) \\ \vdots \\ \delta^{Max-row-t}_{3^k}(k) \end{bmatrix} \tag{5.7}$$

Where:

- $\delta^{Max-row-t}_i(k)$ is the maximum value from the i^{th} row of the $\mathbb{D}_t$ matrix.

In the case where the retransmitted radio blocks compete for access to the radio resources with the radio blocks in the transmission buffer, they slow down other radio blocks within the same stream of IP packets.

Retransmission of a radio block has priority access, and takes one time-slot, thus, the entire IP packet is slowed by one time-slot. Moreover, in the case of a radio block experiencing two retransmission attempts, which is equal to three transmission attempts, the IP packet is slowed by two time-slots. Hence, knowing the number of transmission attempts of a particular radio block allows for the influence of this radio block on the IP packet transmission delay to be calculated.

Since these higher priority retransmitted radio blocks will slow down the entire transmission of the packet, the influence of all such radio blocks needs to be determined. Therefore, the following vector, named $\overline{\Delta}_{Sum-q}(k)$, stored the accumulated delay caused by priority retransmission:

$$\overline{\Delta}_{Sum-q}(k) = \begin{bmatrix} \delta^{Sum-q}_1(k) \\ \delta^{Sum-q}_2(k) \\ \vdots \\ \delta^{Sum-q}_{3^k}(k) \end{bmatrix} \tag{5.8}$$

Where:

- $\delta_i^{Sum-q}(k) = \sum_{j=1}^{k} (r_{i,j} - 1)$

These different contributions can now be combined to find a technique of calculating $\overline{\Delta}_{Total}(k)$.

By adding the vector $\overline{\Delta}_{Max-row-t}(k)$, to the vector $\overline{\Delta}_{Sum-q}(k)$ the total delay for each possible transmission case is obtained.

$$\overline{\Delta}_{Total}(k) = \overline{\Delta}_{Sum-q}(k) + \overline{\Delta}_{Max-row-t}(k) \tag{5.9}$$

5.2.2.4 Main formula

With vectors $\overline{S}(k)$ and $\overline{\Delta}_{Total}(k)$ defined, the average delay of the transmitted IP packet can be computed. This is done by calculating the weighted average of delays of all possible scenarios. Thus, the final formula for the average delay computed with the Brute Force method, denoted as $f_{BF}(k)$ looks like:

$$f_{BF}(k) = \sum_{i=1}^{3^k} \left\{ \delta_i^{Total}(k) \cdot s_i(k) \right\} \tag{5.10}$$

5.2.3 Brute Force (BF) algorithm

The analytical solution for finding the average expected IP packet delay has been determined. It is also possible to implement this solution in an algorithm as follows.

————————-

Step 0:

————————-

START

————————-

Step 1:

————————-

Input the values of:

- IP packet size, k
- ARQ loop feedback, $\overline{\Delta}$
- Radio channel descriptor, $\overline{P}$

from the pre-set data stream input.

————————-

Step 2:

————————-

Create the matrix $\mathbb{R}(k)$:

$$\mathbb{R}(k) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 2 \\ 1 & 1 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & r_{i,j} & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ r_{3^k,1} & r_{3^k,2} & \cdots & r_{3^k,k} \end{bmatrix} \tag{5.11}$$

Where:

- each row denotes one of 3^k possible radio block transmission attempt combinations of an IP packet transmission of the size of k radio blocks.
- each matrix element, $r_{i,j} \in \{1, 2, 3\}$, represents the number of transmission attempts experienced by the j^{th} radio block in the i^{th} constellation.

Step 3:

Create the vector $\overline{S}(k)$

$$\overline{S}(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_{3^k}(k) \end{bmatrix} \tag{5.12}$$

Where:

- $s_i(k) = \prod_{j=1}^{k} p_{r_{i,j}}$. Whereas, $p_{r_{i,j}}$ are elements of the input vector $\overline{P}$.

Step 4:

Create the vector $\overline{\Delta}_a$

$$\overline{\Delta}_a = \begin{bmatrix} \delta_1^a \\ \delta_2^a \\ \delta_3^a \\ \delta_e^a \end{bmatrix} \tag{5.13}$$

Where:

- $\delta_1^a = \delta_1 + \delta_s$

- $\delta_2^a = \delta_1^a + \delta_2 + \delta_s$
- $\delta_3^a = \delta_2^a + \delta_3 + \delta_s$
- $\delta_e^a = \delta_3^a + \delta_e$

and,

- δ_i are elements of the input vector $\overline{\Delta}$
- δ_s represent the time, expressed in the number of radio block periods, necessary for transmission of a single radio block across the medium. Since all δ_i elements are relative to the δ_s, the value of δ_s is one.

————————-

Step 5:

————————-

Create the matrix $\mathbb{D}_t(k)$:

$$\mathbb{D}_t(k) = \begin{bmatrix} \delta_{1,1}^t & \delta_{1,2}^t & \cdots & \delta_{1,k}^t \\ \delta_{2,1}^t & \delta_{2,2}^t & \cdots & \delta_{2,k}^t \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{3^k,1}^t & \delta_{3^k,2}^t & \cdots & \delta_{3^k,k}^t \end{bmatrix} \tag{5.14}$$

Where:

- $\delta_{i,j}^t = \delta_{r_{i,j}}^a + j - 1.$

————————-

Step 6:

————————-

Create the vector $\overline{\Delta}_{Max-row-t}(k)$:

$$\overline{\Delta}_{Max-row-t}(k) = \begin{bmatrix} \delta_1^{Max-row-t}(k) \\ \delta_2^{Max-row-t}(k) \\ \vdots \\ \delta_{3^k}^{Max-row-t}(k) \end{bmatrix} \tag{5.15}$$

Where:

- $\delta_i^{Max-row-t}(k) = max\left\{\delta_{i,1}^t, \delta_{i,2}^t, \cdots, \delta_{i,k}^t\right\}.$

————————-

Step 7:

————————-

Create the vector $\overline{\Delta}_{Sum-q}(k)$:

$$\overline{\Delta}_{Sum-q}(k) = \begin{bmatrix} \delta_1^{Sum-q}(k) \\ \delta_2^{Sum-q}(k) \\ \vdots \\ \delta_{3^k}^{Sum-q}(k) \end{bmatrix} \tag{5.16}$$

Where:

- $\delta_i^{Sum-q}(k) = \sum\limits_{j=1}^{k} (r_{i,j} - 1)$

——————-

Step 8:

——————-

Create the vector $\overline{\Delta}_{Total}(k)$ as the sum of two vectors, $\overline{\Delta}_{Sum-q}(k)$ and $\overline{\Delta}_{Max-row-t}(k)$:

$$\overline{\Delta}_{Total}(k) = \overline{\Delta}_{Sum-q}(k) + \overline{\Delta}_{Max-row-t}(k) \tag{5.17}$$

Where:

- $\delta_i^{Total}(k)$ is the i^{th} element of the vector $\overline{\Delta}_{Total}(k)$

——————-

Step 9:

——————-

Calculate the value of average IP packet delay caused by the transmission in the ARQ loop, denoted by $f_{BF}(k)$

$$f_{BF}(k) = \sum_{i=1}^{3^k} \left\{ \delta_i^{Total}(k) \cdot s_i(k) \right\} \tag{5.18}$$

——————-

Step 10:

——————-

Send the value of $f_{BF}(k)$ into a pre-defined output data stream.

——————-

Step 11:

——————-

END

——————-

5.2.4 Comparison of Simulation from Chapter 4 and BF based results

Using the analytical formula for the Brute Force method, a comparison of results obtained from simulation and calculation was performed for validation of the computer model. The simulation environment is SES/workbench[2], and is shown in Figure 4.3. The simulations were run with the same values of $\overline{P}$, $\overline{\Delta}$ and IP_{size} as used previously in Chapter 4.

The results shown below represent simulations and calculations for the following settings:

- IP packet size, IP_{size}, varying in the range of 1 to 12 radio blocks, with 10,000 transmissions performed for each packet size.

- The $\overline{P}$ vector is represented in five forms, denoted P_0 , P_1 , P_2 , P_3 , P_4 , that cover 5 different radio channel conditions. More details about these vectors can be found in section 4.5.

- The $\overline{\Delta}$ vector is set to have only one value $\overline{\Delta} = [0; 8; 13; 0]^T$ to test one ARQ delay settings. This setting was chosen as it represents a commonly used configuration.

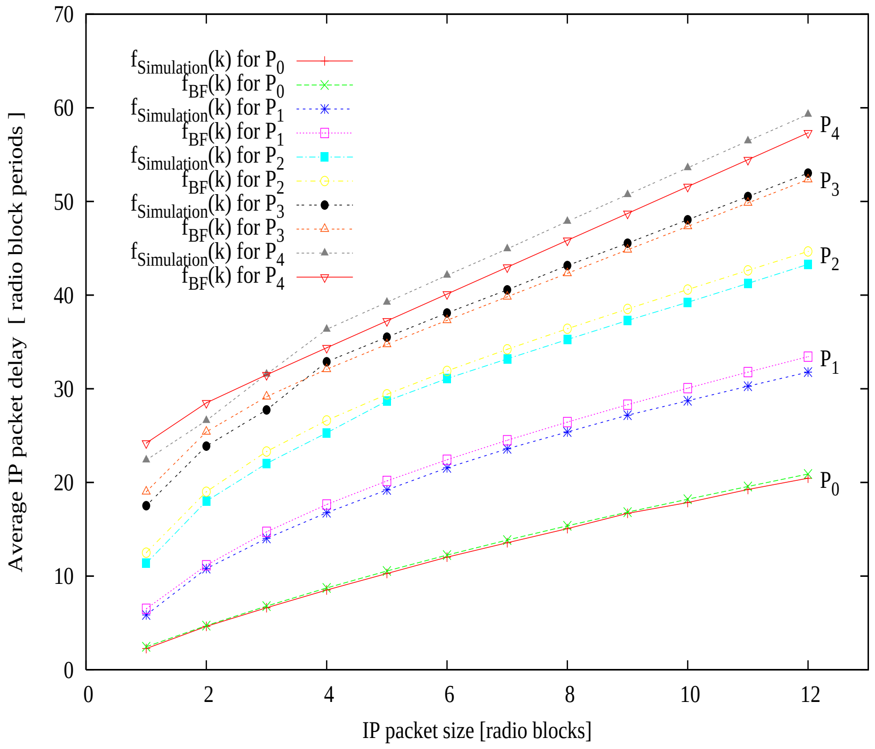


Figure 5.1: *Simulated and calculated average IP packet delay vs its size for five different settings of $\overline{P}$ vector.*

The results, shown in Figure 5.1, indicate that the proposed methodology performs well in each different radio channel cases. The figure 5.2 illustrates the difference between results of simulation and calculation. It strengthens the claim

[2] For more information abut SES/workbench please refer to `http://www.hyperformix.com/`

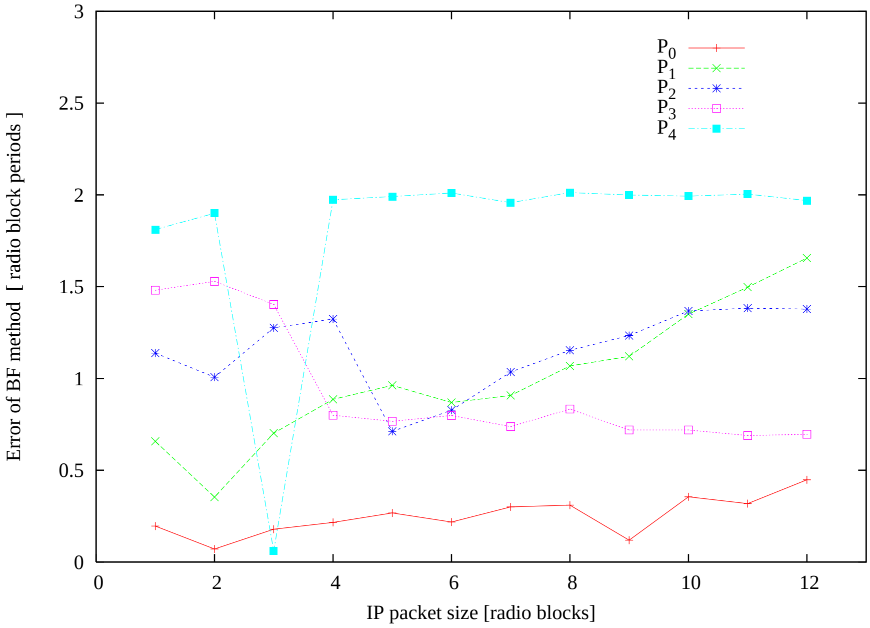


Figure 5.2: *BF error - difference between simulation and BF calculation results,* $abs(f_{Simulation}(k) - f_{BF}(k))$*, for five different settings of* $\overline{P}$ *vector.*

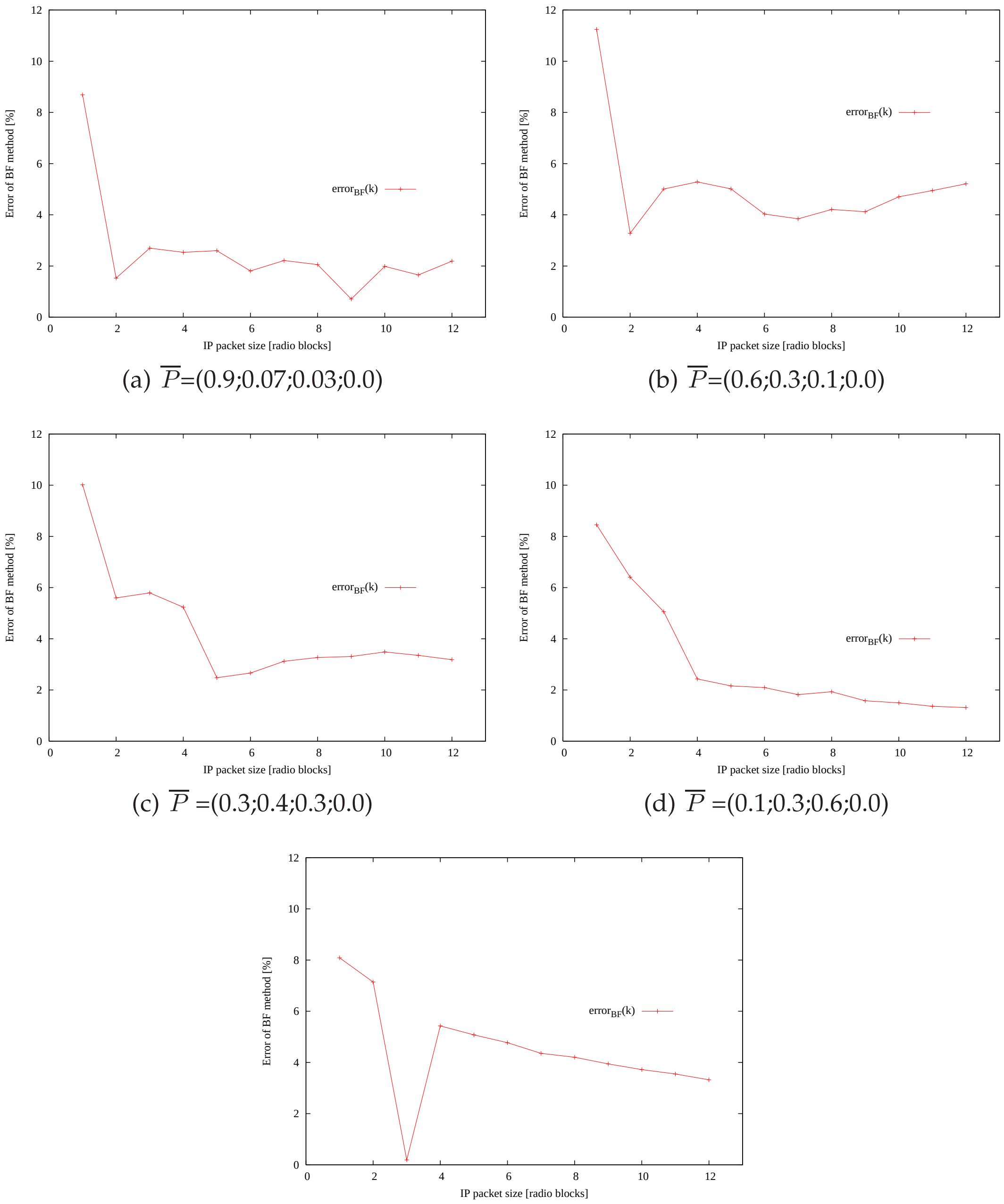
that the BF method offers good accuracy, since the difference between simulation and calculation is less than 2 radio block periods. The abnormal shape of results for radio channel represented by P4 vector is due to the fact that the graph presents absolute value and in the case of P4 radio channel the BF method overestimates the expected average ARQ component of IP packet delay for small IP packet sizes and underestimates it for medium and large ones. The spike on the graph is where the error passes through zero.

The imperfection of the delay estimation comes from the fact that the elements of the $\overline{\Delta}_{Sum-q}(k)$ vector is calculated from the influence that each IP packet radio block constellation has on the average packet delay. Each of these elements, $\delta_i^{Sum-q}(k)$, shows how retransmissions related with each i^{th} constellation slows down the entire IP packet transmission. But it does not consider the fact that some radio blocks of that particular constellation can postpone a transmission of a newly scheduled IP packet. In such a case this new packet will not start its transmission, which results in not affecting this packet by the full value of delay presented by, $\delta_i^{Sum-q}(k)$. This phenomenon is the reason of overestimating the expected average IP packet delay by the BF techniques.

With the favourable results for one $\overline{\Delta}$ vector it is now necessary to investigate the error introduced by the BF method for a higher variety of ARQ delay vectors. A number of simulations have been performed and compared to relevant results calculated from the BF formula 5.19. These results have been plotted for different error transmission and ARQ loop delay vectors. These vectors have the same form as in section 4.5.

$$error_{BF}(k) = \left| \frac{f_{sim}(k) - f_{BF}(k)}{f_{sim}(k)} \right| \cdot 100\% \tag{5.19}$$

The full set of these results can be found in the Appendix F. A representative example is presented in figure 5.3.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 5.3: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

As can be seen in figure 5.3 the error introduced by this method is quite low. For small packets it is around 10 %, whereas in case of medium IP packet size its accuracy is higher, as the error is kept below 5%.

The values of delay analysed in this work represent the most optimistic scenario as queuing at the LLC buffer and core IP network delays are not considered.

Additionally, each cellular network has its own protocol mechanisms that may affect the delay. For example, the time unit in GPRS/EGPRS is 20 ms, while in UMTS it is 10 ms. That simple issue greatly affects the delay of IP packets. If the delay budget at the RLC is 100ms, then from case P_0 in Figure 5.1, VoIP packets in GPRS/EGPRS must be less than two radio blocks, while in UMTS the packets can be greater than four radio blocks long. Thus, UMTS can send higher quality[3] VoIP than GPRS/EGPRS within the same delay budget and independent of the higher bandwidth offered by UMTS.

5.2.5 Limitations of Brute Force algorithm

The accuracy of this technique is assessed by comparing it to simulation results of an identical system architecture. These results show that the differences between simulation and calculation are marginal. The highest error is experienced for an exceptionally high ratio of retransmission case (P_4), although for a low and medium retransmission level the error introduced by the proposed technique is minor.

This low error is a promising feature. However, the complexity of the BF algorithm grows exponentially with the size of the packet. Thus, the use of this method is limited to IP packets composed of approximately 12 radio blocks[4], and the low complexity of this method is bounded at six radio blocks. This is a serious drawback and some other method for predicting the average IP packet delay for larger packets is needed.

5.3 Low Complexity approach

Since the shape of the average IP packet delay possesses similar properties across different simulation settings, it is worth investigating this further. It would be wise to exploit these similarities to find a mathematical description that can be used in the prediction of the average IP packet delay.

Additionally, it is important to keep in mind that the BF method can offer relatively good accuracy and be quite useful if the size of the packet is kept at reasonably low levels. In realistic terms, this means that the packet size must be between one and six radio blocks.

[3] The quality of VoIP is mostly affected by the delay and loss of VoIP application frames. Thus, the system that offers lower delay will be perceived as offering better voice quality. Since UMTS time unit is 10 ms and EGPRS/GPRS time unit is 20ms, then for the same P vector the UMTS will offer twice smaller delay than EGPRS/GPRS kind of system.

[4] The 12 radio blocks can represent the payload of 264 bytes VoIP packet transmitted over EGPRS network with MCS 1 (MCS 1 is used in the case of poor radio reception) or a standard TCP segment of 1500 bytes transported over EGPRS network with MCS 8 (MCS 8 is used in the case of good radio reception).

5.3.1 Study of the shape of the relationship between an average IP packet delay and its size

Consider the simple set of transmission scenarios presented in figure 5.4.

If it is assumed that the radio channel is error free and all radio blocks need only one transmission attempt to successfully reach the receiver. Then, the relationship between the IP packet size, k, and its average delay is a linear function, $f_{ef}(k) = k$.

If this case is extended to include *ideal SR-ARQ*, where the retransmission delay is set to *zero*. Nonetheless, there are some errors present, $\overline{P} = [0.6; 0.3; 0.1; 0.0]^T$ which cause extra resource occupation resulting in a higher slope of a still linear function, $f_{ier}(k) = k \cdot B$.

The last transmission scenario to consider has both non zero feedback delay and losses included, $\overline{P} = [0.6; 0.3; 0.1; 0.0]^T$ and $\overline{\Delta} = [0; 8; 13; 0]^T$. However, the shape of such a scenario is no longer linear. On the contrary it contains a very strong non-linear part at the very beginning of its path, close to the point of origin.

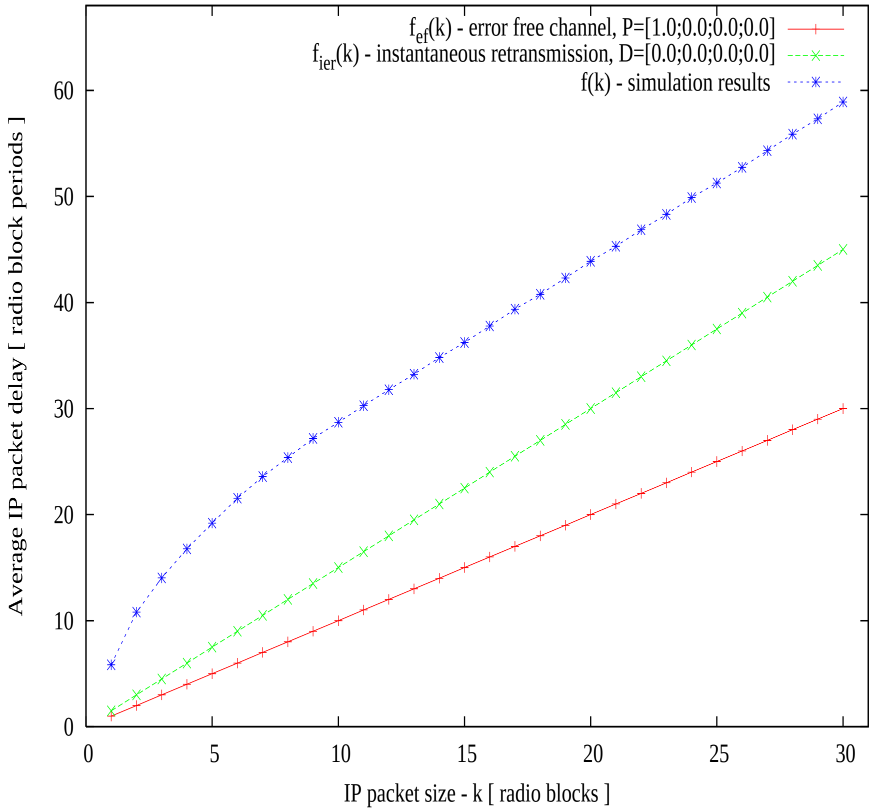


Figure 5.4: *Example of results for* $\overline{P}$ *=(0.6;0.3;0.1;0)* $\overline{\Delta}$ *=(0;8;13;0)*

Looking at these sets of different transmission scenarios it can be said that the general shape of the average IP packet delay versus its size is composed of three major functions.

1. An offset between the values from the realistic transmission scenario and

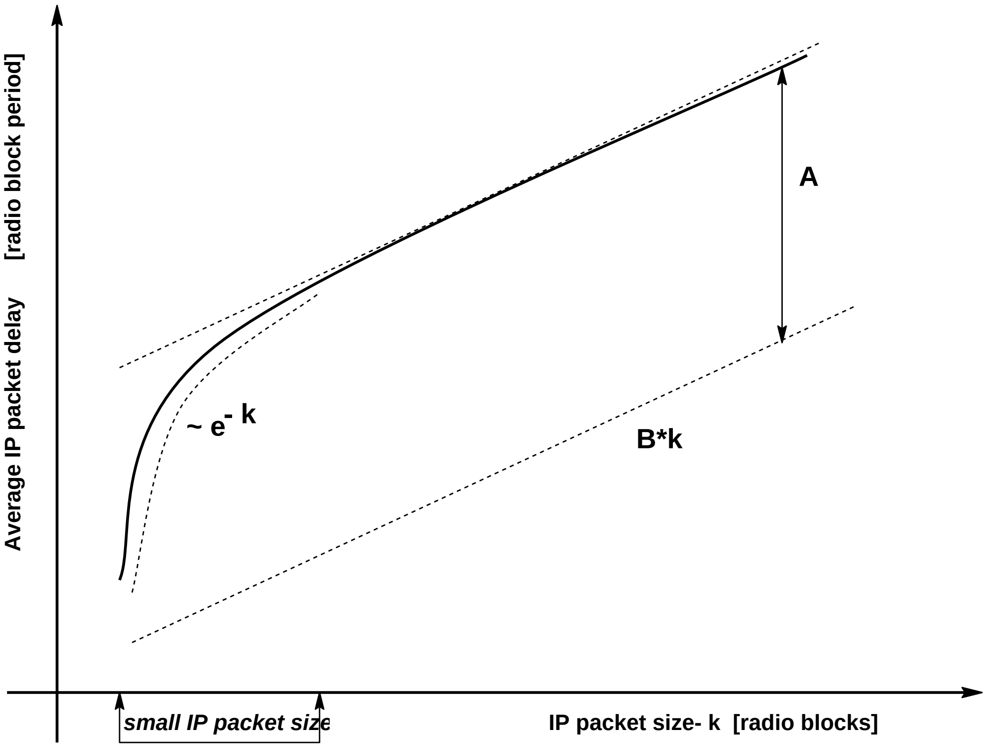


Figure 5.5: *A generic characteristic of an average delay of IP packet within the considered ARQ model*

the idealised *ideal SR-ARQ* case. This offset is represented by a constant function, $f_{offset}(k) = A$.

2. The linear function associated with the fact of multiple resource occupation by a radio block being sent more than once by the BEC mechanism. This function is also linear, but it depends on the average usage of the radio resource by radio blocks, $f_{resource-usage-efficiency}(k) = B \cdot k$.

3. The non-linear part which is quite dominant for small values of the IP packet size and nearly negligible for large IP packet size, $f_{non-linear}(k) \approx e^{-k}$.

5.3.2 Mathematical description of the major components shaping the relation between an average IP packet delay and its size

The major components contributing to the final average IP packet delay characteristic have been specified. The proposed delay function is now described mathematically.

Since all three components contribute to the overall delay characteristic independently, it seems obvious to add them together in order to obtain the function describing the total average delay. Thus,

$$f(k) = f_{offset}(k) + f_{resource-usage-efficiency}(k) + f_{non-linear}(k) \tag{5.20}$$

However, the proposed function contains three independent components and it would be difficult to parameterise these functions having only knowledge about a few values of the average delay from the BF algorithm. Hence, it is necessary to reduce the number of the components needed to be parametrised. One intuitive solution is to look at the dynamism of $f(k)$ function and on the basis of its derivative find the form of the seeking function.

Following this idea, a new approach is proposed. This approach exploits the fact that the BF method offers prediction of the $f(k)$ function involving low computation complexity for small values of IP packets, $k \in \{1, 2, 3, 4, 5, 6\}$, as the size of matrices used for calculation of the IP packet k radio blocks large is 3^k. Taking this into account the approximation of the $f(k)$ function can be expressed as the sum of the average IP packet delay for packet size equal to one and the following average delay progress function, $\overline{\Delta f(l)}$. In this way there are only two components that have to be parametrised, it is possible because of the use of the average delay value offered by BF method.

$$\overline{f(k)} = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{\Delta f(l)} \right) \tag{5.21}$$

$f_{BF}(1)$ is taken directly from the BF method, whereas the form and parameters of $\overline{\Delta f(l)}$ have to be determined from the BF method for small packet sizes and input parameters: k, $\overline{P}$, $\overline{\Delta}$.

A detailed description of the process of finding the form of $\overline{\Delta f(l)}$ is given in Appendix A.

The final form of the prediction of $f(k)$, denoted $f_{CD:m,n}(k)$, is based on $f_{BF}(1)$ and two other points obtained from the BF method, $\{m, f_{BF}(m)\}$ and $\{n, f_{BF}(n)\}$ and has the following form:

$$f_{CD:m,n}(k) = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{B} + \overline{C_{m,n}} \cdot e^{-\overline{D_{m,n}} \cdot l} \right) \tag{5.22}$$

Where:

- $\overline{B} = \sum_{u=1}^{3} (p_u \cdot u)$
- $\overline{D_{m,n}} = \frac{ln\left(\frac{\Delta f_{BF}(n) - \overline{B}}{\Delta f_{BF}(m) - \overline{B}}\right)}{m-n}$
- $\overline{C_{m,n}} = \frac{\Delta f_{BF}(m) - \overline{B}}{e^{-\overline{D_{m,n}} \cdot m}}$
- p_u are elements of $\overline{P}$ vector
- $\Delta f_{BF}(n) = f_{BF}(n) - f_{BF}(n-1)$

- $\Delta f_{BF}(m) = f_{BF}(m) - f_{BF}(m-1)$
- $m, n \in \{2, 3, 4, 5, 6\}$ and $m \neq n$

5.3.3 Low Computation Complexity (LCC) algorithm

As the analytical formula for estimation of the average expected IP packet delay in a low complexity manner is known, it is possible to present an algorithm that will allow this to be implemented in a model.

——————-

Step 0:

——————-

START

——————-

Step 1:

——————-

Input the values of:

- IP packet size, k
- Curve fitting points: m and n, where $m, n \in \{2, 3, 4, 5, 6\}$ and $m \neq n$
- ARQ loop feedback, $\overline{\Delta}$
- Radio channel descriptor, $\overline{P}$

——————-

Step 2:

——————-

Run the *BF algoritm* for $k \in \{1, m, m-1, n, n-1\}$ and store these results at the following variables:

$$\begin{array}{l} f_{BF}(1) \\ f_{BF}(m) \\ f_{BF}(m-1) \\ f_{BF}(n) \\ f_{BF}(n-1) \end{array} \tag{5.23}$$

——————-

Step 3:

——————-

Calculate

$$\begin{array}{l}\overline{B} = \sum_{u=1}^{3} (p_u \cdot u) \\ \Delta f_{BF}(n) = f_{BF}(n) - f_{BF}(n-1) \\ \Delta f_{BF}(m) = f_{BF}(m) - f_{BF}(m-1) \\ \overline{D_{m,n}} = \frac{ln\left(\frac{\Delta f_{BF}(n) - \overline{B}}{\Delta f_{BF}(m) - \overline{B}}\right)}{m-n} \\ \overline{C_{m,n}} = \frac{\Delta f_{BF}(m) - \overline{B}}{e^{-\overline{D_{m,n}} \cdot m}}\end{array} \tag{5.24}$$

Where:

- p_u are elements of $\overline{P}$ vector

——————–

Step 4:

——————–

Calculate

$$f_{CD:m,n}(k) = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{B} + \overline{C_{m,n}} \cdot e^{-\overline{D_{m,n}} \cdot l}\right) \tag{5.25}$$

——————–

Step 5:

——————–

Send the value of $f_{CD:m,n}(k)$ into a pre-defined output data stream.

——————–

Step 6:

——————–

END

——————–

5.3.4 Comparison of Simulation from Chapter 4 and LCC based results

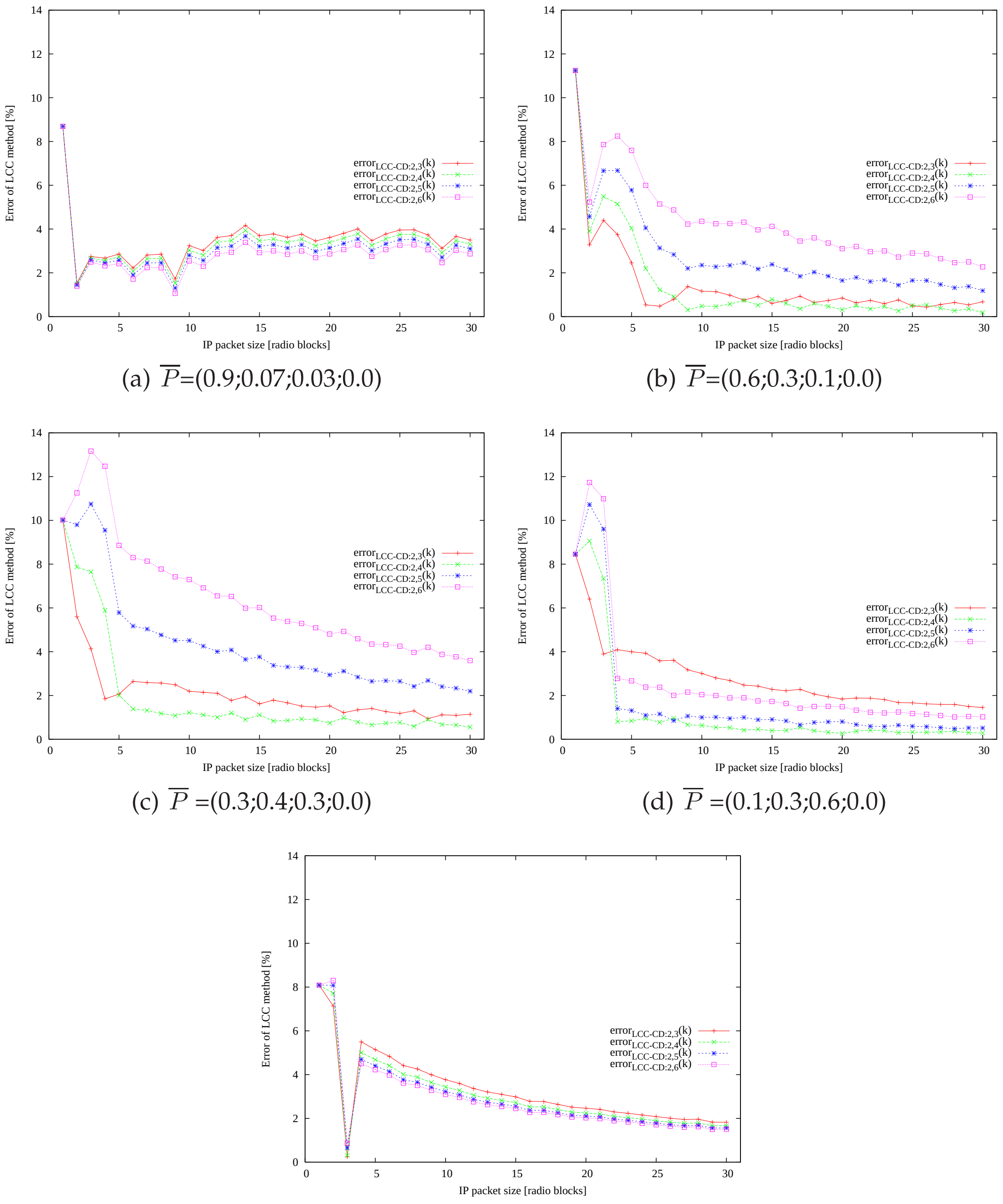
The comparison of the simulation and calculation results of the LCC method has two purposes. Firstly, to track and analyse the level of accuracy offered by the LCC method. Secondly, to choose n and m packet sizes which minimise the average error introduced by the LCC method.

The first task is achieved by running a series of simulations and comparing the results with the results obtained from the LCC algorithm calculations. This comparison is presented in the form of LCC error, calculated accordingly to the formula presented in equation 5.26. Both, simulations and LCC calculations were performed for the same input settings as in section 5.2.4. Hence, the IP packet size varies between 1 and 30 radio blocks. Five different radio channel scenarios

($\overline{P}$) and fifteen different ARQ loop delay vectors ($\overline{\Delta}$) have been simulated and compared to LCC. All of these 75 different settings are included in the Appendix G. An example set of results displaying a general trend is shown in figure 5.6.

$$error_{LCC-CD:m,n}(k) = \left| \frac{f(k) - f_{CD:m,n}(k)}{f(k)} \right| \cdot 100\% \tag{5.26}$$

As can be seen in figure 5.6 the accuracy of the LCC method is of the same order of magnitude as in the BF case. It returns an error prediction below 6% for IP packets larger than six radio blocks. Due to the fact that the LCC offers much lower computation complexity indicates that LCC outperforms the BF approach.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 5.6: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

The second task in this section is to address the problem of choosing values of IP packet size for the computation of the $f_{CD:m,n}(k)$ function. Since n and m can take any value between two and five, there is a set of different solutions for C and D parameters. This will affect the final version of the approximation function. Therefore, there is a need to examine different combinations of n and m values to find one that introduces the smallest error between the delay function $f(k)$, obtained from simulation, and the approximation of this function $f_{CD:m,n}(k)$, obtained from calculation.

In order to achieve this an LCC error is computed for all possible combinations of IP packet sizes, radio channel conditions and ARQ loop delays, as outlined in section 4.5. This error is then averaged for each combination of $\overline{P}$ and $\overline{\Delta}$ for values between seven and thirty, as presented in equation 5.27. The averaging process is performed for values higher than six since it was shown in section 4.6 that the use of average IP packet size is validated for middle and large IP packet sizes. Whereas, the maximum analysed IP packet size is thirty because for that value of packet size both simulation and LCC results do not experience changes in the error value, both lines follow the same trajectory.

$$e_{CD:n,m} = \frac{1}{23} \cdot \sum_{k=7}^{30} (error_{LCC-CD:m,n}(k)) \tag{5.27}$$

The results of these calculations for all 75 different $\overline{\Delta}$ and $\overline{P}$ vector constellations are presented in Table 5.1. The last row of this table shows an average value of each column, representing an average error introduced by the LCC algorithm based on a relevant pair of n and m IP packet sizes.

The lowest error is introduced by the LCC algorithm based on $n = 2$ and $m = 5$, represented by the $f_{CD:5,2}(k)$. However, the LCC algorithm using $n = 2$ and $m = 4$, $f_{CD:4,2}(k)$, has only slightly higher error but lower computation complexity. The LCC method based on the equation $f_{CD:5,2}(k)$ has to use matrices of the size 3^5, whereas the $f_{CD:4,2}(k)$ approach needs only matrices of the maximum size of 3^4. This marginal difference in terms of introduced error and lower computation complexity leads to the choice of $n = 2$ and $m = 4$.

This choice is further supported by the analysis of the average, standard deviation and maximum values of the $e_{CD:n,m}$ per transmission scenario, shown in figure 5.2. In the case of the radio channel represented by P_0 and P_4, all average values are nearly the same. In the P_1 scenario the best value offers $f_{CD:2,6}(k)$ but the improvement is slight and the computation complexity associated with this function is much bigger than in case of $f_{CD:2,4}(k)$. In the case of P_2, the $f_{CD:2,4}(k)$ has the smallest error and still holds its low computation complexity. Finally, in the case of P_3, the $f_{CD:2,5}(k)$ slightly outperforms $f_{CD:2,4}(k)$ but with the expense of higher complexity.

Thus, taking into account the balance between accuracy and computation complexity of the analysed LCC functions the equation proposed for use in the LCC technique is the following:

$$f_{CD:4,2}(k) = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{B} + \overline{C_{4,2}} \cdot e^{-\overline{D_{4,2}} \cdot l} \right) \tag{5.28}$$

5.3.5 Conclusions

The Low Computation Complexity algorithm for prediction of an average IP packet delay in cellular data networks caused by the use of SR-ARQ has been proposed.

The complexity of the proposed algorithm is kept at a low level, comparable to the BF approach. Moreover, this complexity is maintained constant, regardless of the size of the packet being analyzed. In contrast, the Brute Force algorithm, proposed previously, has exponentially growing complexity.

This low complexity makes the LCC technique implementable in a real system, where a dedicated node, like a BSC in the EGPRS case, has to maintain a large number of such predictions. This low complexity allows for both fast prediction for a selected connection and high prediction capacity in terms of the number of users that a node using LCC approach needs to service.

Having knowledge about the average IP packet delay at RLC level, it is possible to enrich the MAC's scheduling algorithm. Since RLC is capable of delivering the prediction of the average delay of IP packet for a single radio channel, MAC can dynamically adjust the temporary access to the radio resources in order to fulfill the QoS contract, if promised. Moreover, the knowledge about the delay performance of the IP packet at RLC layer can be a good indicator of the connection quality for higher protocol layers. The knowledge about C/I, BER or BLER values has limited applicability for higher protocol layers. On the contrary, the estimation of mapping of all of previously mentioned factors on the delay performance of the analysed IP packet may be directly used by those higher layers. The influence of ARQ loop on the IP packet delay performance cannot be compensated by the MAC scheduling algotirhm offering more of the radio resources. Since delay associated with the feedback depends not on the accessible bandwidth but on the internal ARQ protocol loop design, the ARQ loop inherited IP packet delay component can be the variable describing the minimum average IP packet delay performance. It may replace or complement the C/I, BER or BLER parameters, adding a new IP performance orientated dimension to the description of the wireless connection.

$\overline{P}$ and $\overline{\Delta}$ constalation	$e_{CD:2,3}$	$e_{CD:2,4}$	$e_{CD:2,5}$	$e_{CD:2,6}$
-	[%]	[%]	[%]	[%]
$P_0 : \overline{P} = [0.9; 0.07; 0.03; 0]$ &				
$\overline{\Delta} = [0; 8; 13; 0]$	3.53	3.32	3.08	2.81
$\overline{\Delta} = [0; 10; 15; 0]$	3.23	3.09	2.94	2.77
$\overline{\Delta} = [0; 12; 17; 0]$	2.84	2.76	2.67	2.57
$\overline{\Delta} = [0; 14; 19; 0]$	2.41	2.39	2.35	2.31
$\overline{\Delta} = [0; 16; 21; 0]$	1.32	1.54	1.77	2.00
$\overline{\Delta} = [0; 13; 23; 0]$	1.51	1.55	1.58	1.62
$\overline{\Delta} = [0; 15; 25; 0]$	1.18	1.25	1.32	1.40
$\overline{\Delta} = [0; 17; 27; 0]$	0.92	1.01	1.1	1.19
$\overline{\Delta} = [0; 19; 29; 0]$	0.63	0.73	0.85	0.97
$\overline{\Delta} = [0; 21; 31; 0]$	0.48	0.56	0.67	0.80
$\overline{\Delta} = [0; 18; 33; 0]$	0.53	0.5	0.52	0.57
$\overline{\Delta} = [0; 20; 35; 0]$	0.67	0.61	0.6	0.63
$\overline{\Delta} = [0; 22; 37; 0]$	0.78	0.69	0.62	0.59
$\overline{\Delta} = [0; 24; 39; 0]$	0.93	0.81	0.73	0.67
$\overline{\Delta} = [0; 26; 41; 0]$	0.49	0.62	0.81	1.01
$P_1 : \overline{P} = [0.6; 0.3; 0.1; 0]$ &				
$\overline{\Delta} = [0; 8; 13; 0]$	0.77	0.47	1.9	3.44
$\overline{\Delta} = [0; 10; 15; 0]$	1.77	0.43	1.17	2.90
$\overline{\Delta} = [0; 12; 17; 0]$	2.78	1.31	0.4	2.26
$\overline{\Delta} = [0; 14; 19; 0]$	3.73	2.18	0.52	1.62
$\overline{\Delta} = [0; 16; 21; 0]$	4.6	2.97	1.17	1.03
$\overline{\Delta} = [0; 13; 23; 0]$	5.67	3.79	1.7	0.71
$\overline{\Delta} = [0; 15; 25; 0]$	6.44	4.52	2.36	0.76
$\overline{\Delta} = [0; 17; 27; 0]$	7.17	5.21	2.96	1.12
$\overline{\Delta} = [0; 19; 29; 0]$	7.79	5.79	3.49	1.47
$\overline{\Delta} = [0; 21; 31; 0]$	8.39	3.36	4.02	1.88
$\overline{\Delta} = [0; 18; 33; 0]$	9.33	7.09	4.54	2.23
$\overline{\Delta} = [0; 20; 35; 0]$	9.84	7.58	5.01	2.58
$\overline{\Delta} = [0; 22; 37; 0]$	10.29	8.03	5.43	2.93
$\overline{\Delta} = [0; 24; 39; 0]$	10.72	8.45	5.83	3.24
$\overline{\Delta} = [0; 26; 41; 0]$	11.13	8.84	6.21	3.59
$P_2 : \overline{P} = [0.3; 0.4; 0.3; 0]$ &				
$\overline{\Delta} = [0; 8; 13; 0]$	1.6	0.88	3.28	5.32
$\overline{\Delta} = [0; 10; 15; 0]$	2.2	0.53	3.23	5.59
$\overline{\Delta} = [0; 12; 17; 0]$	2.77	0.17	3.13	5.77
$\overline{\Delta} = [0; 14; 19; 0]$	3.35	0.23	2.97	5.85
$\overline{\Delta} = [0; 16; 21; 0]$	3.88	0.6	2.8	5.90
$\overline{\Delta} = [0; 13; 23; 0]$	4.61	1.15	2.22	5.15
$\overline{\Delta} = [0; 15; 25; 0]$	5.01	1.4	2.16	5.30
$\overline{\Delta} = [0; 17; 27; 0]$	5.41	1.67	2.06	5.38
$\overline{\Delta} = [0; 19; 29; 0]$	5.77	1.92	1.96	5.45
$\overline{\Delta} = [0; 21; 31; 0]$	6.02	2.07	1.96	5.60
$\overline{\Delta} = [0; 18; 33; 0]$	6.7	2.63	1.33	4.78
$\overline{\Delta} = [0; 20; 35; 0]$	6.9	2.73	1.37	4.98
$\overline{\Delta} = [0; 22; 37; 0]$	7.11	2.85	1.38	5.13
$\overline{\Delta} = [0; 24; 39; 0]$	7.36	3.02	1.31	5.19
$\overline{\Delta} = [0; 26; 41; 0]$	7.53	3.12	1.32	5.32

$\overline{P}$ **and** $\overline{\Delta}$ **constalation**	$e_{CD:2,3}$	$e_{CD:2,4}$	$e_{CD:2,5}$	$e_{CD:2,6}$
-	[%]	[%]	[%]	[%]
$P_3 : \overline{P} = [0.1; 0.3; 0.6; 0]\,\&$				
$\overline{\Delta} = [0; 8; 13; 0]$	2.15	0.43	0.75	1.50
$\overline{\Delta} = [0; 10; 15; 0]$	2.43	0.49	0.88	1.78
$\overline{\Delta} = [0; 12; 17; 0]$	2.5	0.36	1.19	2.21
$\overline{\Delta} = [0; 14; 19; 0]$	2.47	0.14	1.57	2.71
$\overline{\Delta} = [0; 16; 21; 0]$	2.4	0.1	1.94	3.19
$\overline{\Delta} = [0; 13; 23; 0]$	4.11	1.77	0.15	0.90
$\overline{\Delta} = [0; 15; 25; 0]$	4.06	1.56	0.20	1.35
$\overline{\Delta} = [0; 17; 27; 0]$	4.09	1.44	0.44	1.69
$\overline{\Delta} = [0; 19; 29; 0]$	4.19	1.42	0.58	1.91
$\overline{\Delta} = [0; 21; 31; 0]$	4.31	1.41	0.69	2.10
$\overline{\Delta} = [0; 18; 33; 0]$	5.44	2.69	0.80	0.44
$\overline{\Delta} = [0; 20; 35; 0]$	5.43	2.57	0.57	0.75
$\overline{\Delta} = [0; 22; 37; 0]$	5.36	2.38	0.28	1.11
$\overline{\Delta} = [0; 24; 39; 0]$	5.22	2.14	0.07	1.52
$\overline{\Delta} = [0; 26; 41; 0]$	5.19	2.00	0.27	1.80
$P_4 : \overline{P} = [0.03; 0.07; 0.9; 0]\,\&$				
$\overline{\Delta} = [0; 8; 13; 0]$	2.7	2.47	2.31	2.22
$\overline{\Delta} = [0; 10; 15; 0]$	3.14	2.86	2.68	2.57
$\overline{\Delta} = [0; 12; 17; 0]$	3.65	3.34	3.14	3.01
$\overline{\Delta} = [0; 14; 19; 0]$	3.71	3.37	3.15	3.01
$\overline{\Delta} = [0; 16; 21; 0]$	3.64	3.28	3.04	2.88
$\overline{\Delta} = [0; 13; 23; 0]$	4.57	4.25	4.04	3.91
$\overline{\Delta} = [0; 15; 25; 0]$	4.47	4.13	3.90	3.76
$\overline{\Delta} = [0; 17; 27; 0]$	4.15	3.77	3.53	3.37
$\overline{\Delta} = [0; 19; 29; 0]$	4.11	3.71	3.45	3.28
$\overline{\Delta} = [0; 21; 31; 0]$	4.33	3.92	3.64	3.46
$\overline{\Delta} = [0; 18; 33; 0]$	5.53	5.15	4.91	4.76
$\overline{\Delta} = [0; 20; 35; 0]$	5.53	5.13	4.87	4.71
$\overline{\Delta} = [0; 22; 37; 0]$	5.56	5.15	4.88	4.70
$\overline{\Delta} = [0; 24; 39; 0]$	5.33	4.89	4.61	4.42
$\overline{\Delta} = [0; 26; 41; 0]$	5.01	4.56	4.26	4.06
Average	*4.30*	*2.67*	*2.23*	*2.85*
Std. Dev.	*2.55*	*2.11*	*1.56*	*1.66*
Maximum	*11.13*	*8.84*	*6.21*	*5.9*

Table 5.1: *LCC errors for all combinations of delay,* $\overline{\Delta}$, *and radio channel,* $\overline{P}$, *vectors.*

$\overline{P}$ constalation	$e_{CD:2,3}$	$e_{CD:2,4}$	$e_{CD:2,5}$	$e_{CD:2,6}$
-	[%]	[%]	[%]	[%]
$P_0 : \overline{P} = [0.9; 0.07; 0.03; 0]$				
Average	1.43	1.43	1.44	1.46
Standard Deviation	1.05	0.99	0.91	0.83
Maximum	3.53	3.32	3.08	2.81
$P_1 : \overline{P} = [0.6; 0.3; 0.1; 0]$				
Average	6.7	4.87	3.12	2.12
Standard Deviation	3.37	2.90	1.98	0.98
Maximum	11.13	8.84	6.21	3.59
$P_2 : \overline{P} = [0.3; 0.4; 0.3; 0]$				
Average	5.08	1.66	2.17	5.38
Standard Deviation	1.95	1.05	0.75	0.32
Maximum	7.53	3.12	3.28	5.9
$P_3 : \overline{P} = [0.1; 0.3; 0.6; 0]$				
Average	3.96	1.39	0.69	1.66
Standard Deviation	1.25	0.89	0.53	0.73
Maximum	5.44	2.69	1.94	3.19
$P_4 : \overline{P} = [0.03; 0.07; 0.9; 0]$				
Average	4.36	4.00	3.76	3.61
Standard Deviation	0.90	0.86	0.83	0.81
Maximum	5.56	5.15	4.91	4.76

Table 5.2: *Average, Standard Deviation and Maximum of LCC errors for five radio channel vectors, $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.*

5.4 Final comparison of Simulation, BF and LCC results

The accuracy analysis of BF and LCC algorithms presented in this chapter has been so far based on the comparison of the errors introduced by both techniques. In both cases the level of these errors were kept on a low level mostly below 5 %. Nonetheless, it is important to investigate the accuracy of these methods in a broader scope. Thus, the accuracy analysis in this section focuses on the relationship between values of the average ARQ component of IP packet delay obtained from simulation and the same values obtained as a result of use of BF and LCC prediction techniques. Hence, for each $\overline{P}$ and $\overline{\Delta}$ constellation specified in section 4.5 the graph representing valuses of simulation, BF and LCC method for different curve fitting points is plotted.

All of these scenarios can be found in Appendix F, whereas an exemplary figures for the $\overline{P} \in (P_0, P_1, P_2, P_3, P_4)$ and $\overline{\Delta} = [0; 8; 13; 0]^T$ settings are shown in figure 5.7.

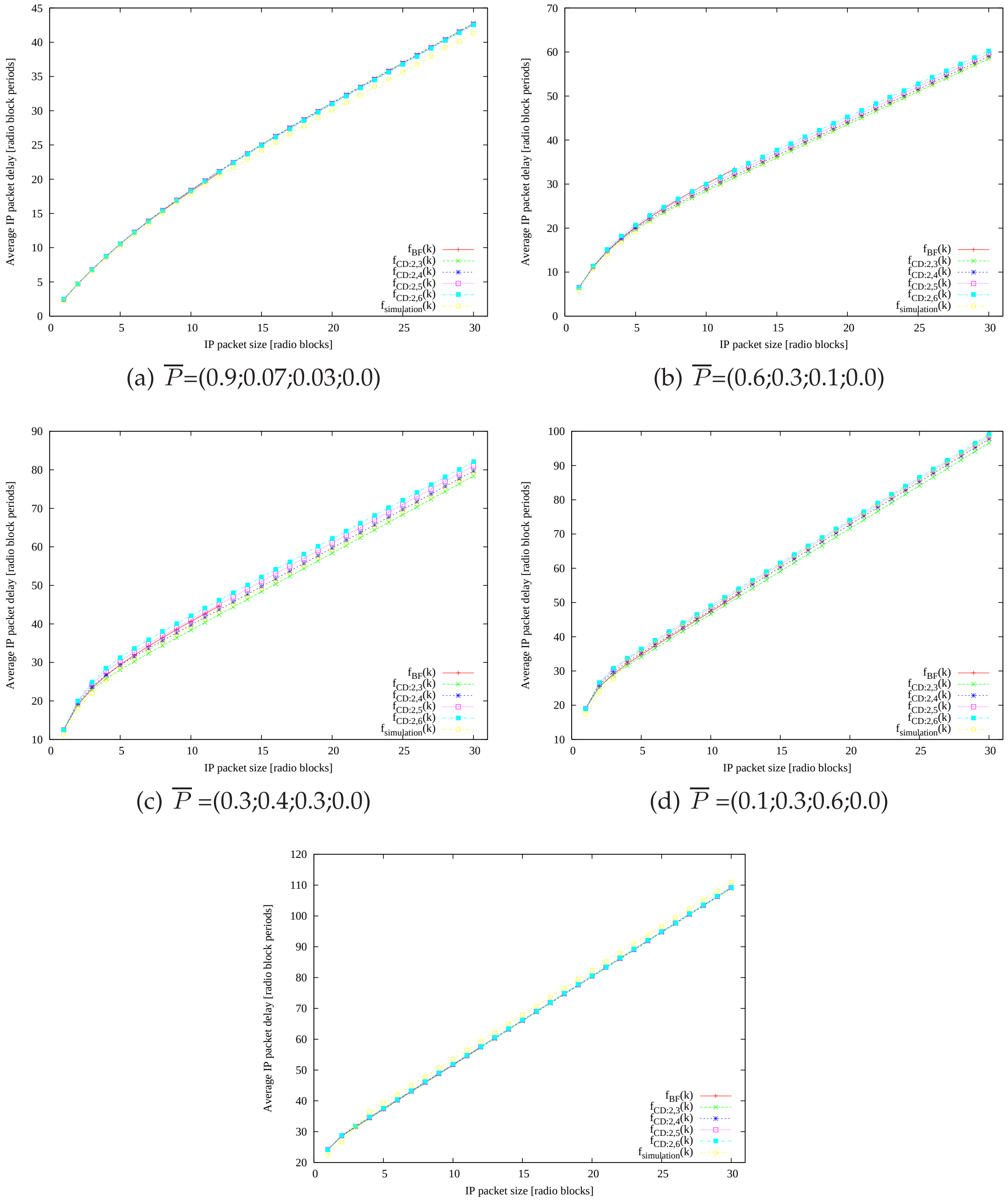
It is clearly visible that the proposed prediction techniques tracks the shape of the average ARQ delay component. Hence, these graphs show that both BF and LCC algorithms return values with minimal error level and can be used as the RLC delay performance descriptors.

5.5 Summary

This chapter presented two new methods for the prediction of the expected average IP packet delay characteristic in Hybrid Type I/II/III ARQ systems. These new methods are called Brute Force and Low Computation Complexity algorithms.

The first one, BF, has its roots in the step by step analysis of sending an IP packet over a radio channel in a mobile data system. It is characterised by quite good accuracy. However, its main drawback is the increase in complexity that corresponds to an increase in the size of the analysed IP packets. Realistically speaking, any analysis of IP packet size larger than six radio blocks is not feasible from a complexity point of view.

This second method uses extrapolation from two points from the BF method to approximate the rest of the delay characteristic. This approximation offers a good accuracy level and is free from the problem of increasing complexity with increasing IP packet size. Thus, the LCC method is suitable for terminal equipment with limited energy and computation capabilities. Additionally, the low computation requirements of LCC algorithm makes it a good candidate for being used in VoIP and Multimedia applications, where the algorithm must be both



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure 5.7: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

relatively accurate and fast.

CHAPTER 6

LCC Implementation in EGPRS

6.1 Overview

This chapter explains how the LCC technique could be implemented in an EGPRS network and what adjustments are necessary to implement the LCC method. Firstly, the simplified network topology is presented with emphasis put on the nodes that are part of the air interface and that require some modification in order to implement the LCC method. Following that, the required modifications are presented. At the next step, the process of gathering the data necessary for computing the $\overline{P}$ vectors is presented. Finally, the use of $\overline{P}$, $\overline{\Delta}$ and IP packet size is shown.

6.2 EGPRS

A detailed description of the (E)GPRS network can be found in [10, 55]. A short summary is given in section 2.3.1.2 and the general architecture of (E)GPRS systems is presented in Figure 6.1. In brief, the path of an IP packet being transported from the Internet to a Mobile Station (MS) over (E)GPRS is the following. The packet arrives at one of (E)GPRS GGSN edge nodes, where it is forwarded to the SGSN that is currently associated with the MS. The SGSN, in turn, forwards the packet to the BSC currently managing the MS, and then to the BTS. Finally, the BTS transmits the packet as radio blocks to the destination MS.

As the topic of this thesis is to investigate the influence of ARQ mechanisms on IP packet transmission, the nodes that have to be considered closely are BSC and BTS, from the network side, and MS, from the user side. The BSC receives an LLC frame that contains an IP packet being sent from the SGSN. Following that, the BSC node creates a number of radio blocks according to the chosen MCS and

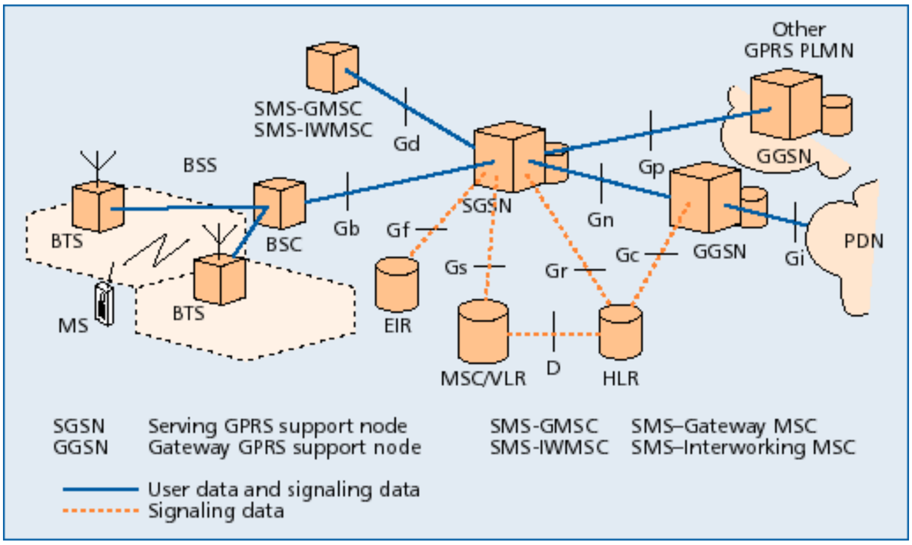


Figure 6.1: *(E)GPRS system architecture. [28]*

the size of this particular packet. These radio blocks are then sent to the BTS and from there are transmitted over the air interface to the destination MS.

The LCC method requires the following input parameters: IP packet size, k, Radio channel description, $\overline{P}$, ARQ loop description, $\overline{\Delta}$. Thus, the BSC, BTS and MS have to be modified in order to have access to those parameters.

The easiest place to start is to calculate the size of transmitted IP packet. This can be done at the BSC by comparing of the size of the IP packet to the payload size of the radio block, which is determined by the chosen MCS. The number of radio blocks necessary, k, can be found by simple division. The relevant equation is given in section 4.5.1.

The next challenge that has to be addressed is to find the value of $\overline{P}$ vector. This can be done at the BSC. There are two parameters that need to be set before the $\overline{P}$ vector is calculated. Firstly, the predictor has to know what the expected time window (TW) is. In other words, what time horizon needs to be considered for prediction of the $\overline{P}$ vector. Secondly, the system has to operate with the same MCS during the computation of the $\overline{P}$ vector. This is necessary to keep the relation between MCS payload and IP packet size fixed. This condition is valid when Link Adaptation (LA) procedure in EGPRS is turned off or if the speed of the LA algorithm is dictated by the size of time window (TW_{size}). In the GPRS system the problem of selecting MCS does not exist, as all data transmission in deployed networks is performed with Coding Schemes 2 (CS2).

Once the MCS and TW_{size} is known, the next step is to perform the transmission of data over the wireless link and calculate the number of radio blocks sent successfully at the first, second and third transmission attempts and the number of radio blocks that were not received successfully after three transmission attempts. This data is then stored in a matrix $pdf_{MCSx}[z]$, which keeps the distribution for each transmission window, denoted by index z for MCSx. Following that, the value of $\overline{P}$ for previously chosen MCSx, $\overline{P_{MCSx}}$, is predicted on the basis of n last time window distributions, $\overline{P_{MCSx}}[z+1] = f(pdf_{MCSx}[z], pdf_{MCSx}[z-1], ...pdf_{MCSx}[z-n+1])$

At the next step, the condition of the radio link is checked and a new MCSx is selected. If a prediction of the $\overline{P}$ vector calculated for the newly selected MCS already exists, it can be used to predict the influence of ARQ loop on average IP packet delay. If there is no data regarding the selected MCS, then the prediction of ARQ loop influence can not be performed, as there is no history of previous transmissions with this particular MCS. The situation is much simpler if LA is turned off since only one MCS is used for all transmissions. It is the same scenario in the case of GPRS transmissions, for the reasons explained earlier.

The process of $\overline{P}$ vector prediction is shown in figure 6.2.

The last variable necessary to use the LCC technique is the $\overline{\Delta}$ vector. This vector represents the delays associated with each transmission attempt. The data necessary to create $\overline{\Delta}$ vector can be collected at the BSC node. Each radio block which is transmitted more than once has to spend an extra time between the moments when it is received with errors at the MS and transmitted again at the BSC. This additional time can be easily measured for every radio block transmission attempt and stored in a temporary vector $\overline{\Delta_{temp}}$. This vector stores the sum of all delay experienced by the radio blocks which were received without errors at the first, second and third transmission attempts. When a large number of samples is collected each element is divided by the population of radio block for one, two and three transmission attempts. This represents the average ARQ loop influence on each transmission attempt and is stored in $\overline{\Delta}$ vector.

6.3 Expected BLER in real networks

Since the access to the equipment from a real network is generally very limited or impossible to obtain, it is necessary to find ways to approximate the expected values of BLER and consequently the $\overline{P}$ vector. Having these values would allow tests to be run to look at the expected delay performance dimensioned in seconds. This section aims to deliver the most likely BLER and $\overline{P}$ values for EGPRS type of systems and analyse the delay characteristic of these systems for typical conditions.

As the Hybrid Type I ARQ does not store previously transmitted radio blocks, the process of finding BLER and values for the $\overline{P}$ vector differs from the case considering Hybrid Type II/III ARQ. This fundamental difference is the reason behind separating the analysis of BLER and $\overline{P}$ vector for Hybrid Type I ARQ and Hybrid Type II/III ARQ.

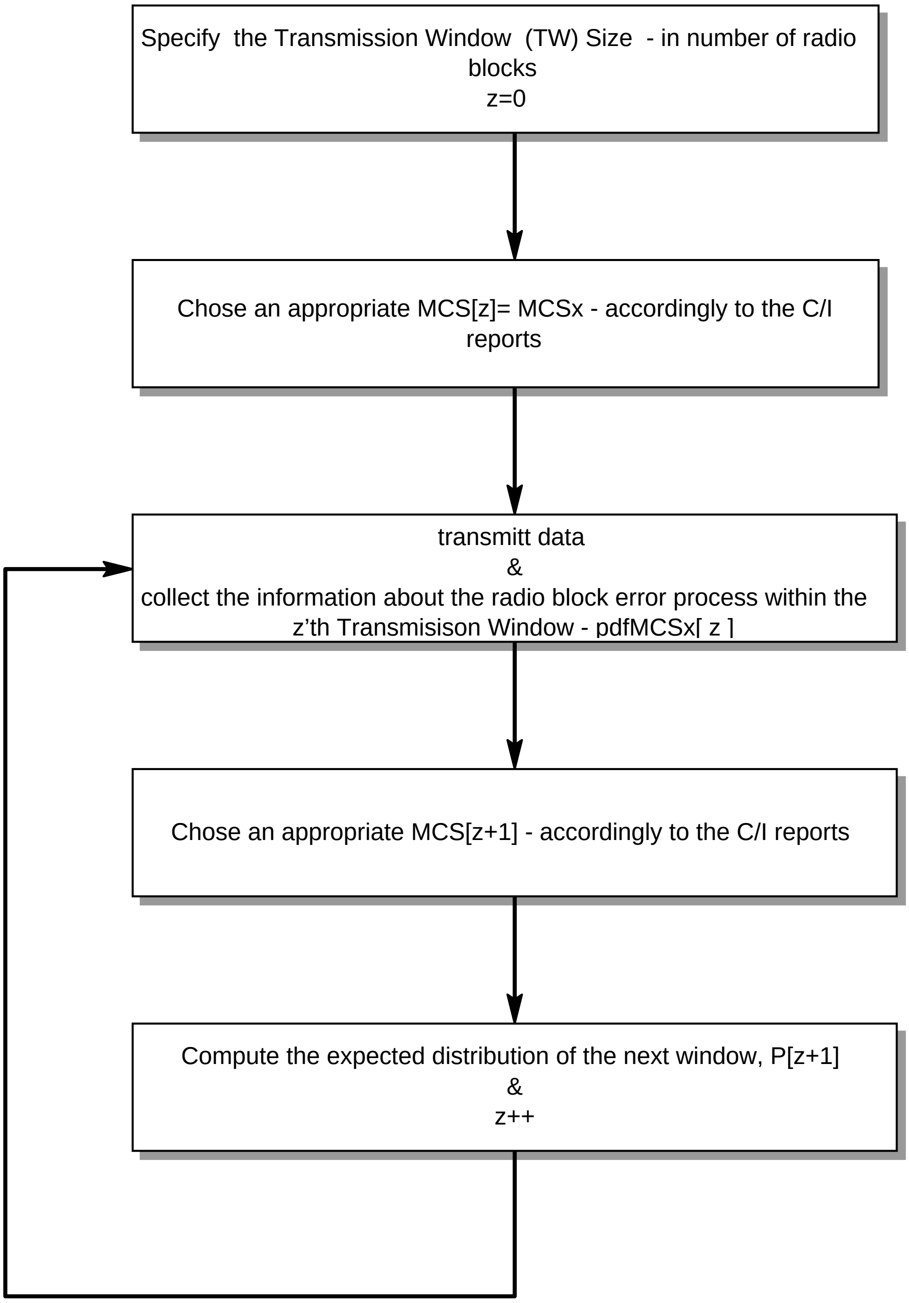

Specify the Transmission Window (TW) Size - in number of radio blocks
z=0
Chose an appropriate MCS[z]= MCSx - accordingly to the C/I reports
transmitt data
&
collect the information about the radio block error process within the z'th Transmisison Window - pdfMCSx[z]
Chose an appropriate MCS[z+1] - accordingly to the C/I reports
Compute the expected distribution of the next window, P[z+1]
&
z++

Figure 6.2: *Process of gathering data for $\overline{P}$ vector prediction*

6.3.1 Hybrid Type I ARQ

This sub-section presents methods for estimating the expected values for BLER and $\overline{P}$ vectors for Hybrid Type I ARQ. Thus, this section answers the following questions:

- How to compute values of $\overline{P}$ vector ?
- How to compute the BLER after each transmission attempt?
- How to compute the initial error rate – BLER at the first transmission attempt?

Firstly, let's focus on the methodology of finding the value of the $\overline{P}$ vector for a given BLER. The $\overline{P}$ vector stores probability associated with successful reception of radio blocks at the first attempt, p_1, the second attempt, p_2, the third attempt, p_3, transmission attempt and probability of failure after three transmission attempts, p_e. Thus, the simplest way of describing the relationsip between BLER and elements of $\overline{P}$ vectors is the following:

$$\overline{P} = \begin{bmatrix} p_1 = 1 - BLER_1 \\ p_2 = BLER_1 * (1 - BLER_2) \\ p_3 = BLER_1 * BLER_2 * (1 - BLER_3) \\ p_e = BLER_1 * BLER_2 * BLER_3 \end{bmatrix} \tag{6.1}$$

Where:

- $BLER_1$ represents BLER after single transmission attempt
- $BLER_2$ represents BLER after two transmission attempts
- $BLER_3$ represents BLER after three transmission attempts

Secondly, it is necessary to find the method of finding the BLERs at each transmission attempts. In case of Hybrid Type I ARQ it is relatively simple, as consequent transmissions do not have any relationship with each other, hence:

$$\begin{aligned} BLER_1 &= (BLER_1)^1 \\ BLER_2 &= (BLER_1)^2 \\ BLER_3 &= (BLER_1)^3 \end{aligned} \tag{6.2}$$

The relationship between $BLER_1$ and $BLER_2$ and $BLER_3$ is known, but still the $BLER_1$ is not known. Thus, at the third step it is necessary to find $BLER_1$ for a given radio channel and MCS. This can be achieved by using information about the history of C/I measurement over a selected period of time in combination with information about the BLER performance of the selected MCS, the speed of the terminal, the type of environment the transmission is operating in, and the presence or absence of Frequency Hopping. An example of a probability density function of C/I for EGPRS system is given in [10] and is presented in the figure 6.4. An example of the BLER(C/I) function for different MCSs is given in [3] and is presented in figure 6.3.

Once the data from these graphs are tabularised (for each C/I the value of the pdf distribution and the BLER are taken from the graph and associated with this C/I - meaning there are two functions pfd(C/I) and BLER(C/I)), it is possible to calculate the $BLER_1$ according to the following formula:

$$BLER_1 = \int_0^\infty \{pdf(C/I) \cdot BLER(C/I)\} \, d(C/I) \tag{6.3}$$

Once $BLER_1$ is known the other BLERs and the $\overline{P}$ vector can be calculated.

For example the value of $BLER_1$ for MCS4, taken from figure 6.3, and given the pdf(C/I) presented in figure 6.4 is 14%. Hence $BLER_2$=0.02 (2%) and $BLER_3$=0.003 (0.3%).

Therefore:

$$\begin{aligned} p_1 &= 0,859164 (\approx 86\%) \\ p_2 &= 0,138043 (\approx 13.7\%) \\ p_3 &= 0,002786 (\approx 0.3\%) \\ p_e &= 0,000007804 (\approx 0\%) \end{aligned} \tag{6.4}$$

The values presented above are just an example of $\overline{P}$ vector, since the BLER characteristic depends on many factors like radio planning of the cell, chosen MCS, velocity of the moving terminal, type of surrounding environment, etc. Thus, it is hard to say that there is one typical BLER value for EGPRS type of networks.

Due to the lack of *typical* $BLER_1$, the delay performance of Hybrid Type I ARQ will be tested on a range of initial block error rates, $BLER_1$, varying from 0.1% up to 30%. These scenarios are presented in table 6.1. The figures 6.5 and 6.6 show the delay performance of three different MCSs (MCS2, MCS4 and MCS9) under BLER conditions specified in table 6.1. The IP packet size has been limited to 500 Bytes, as in [2] it is assumed as an average IP packet size used for carrying Multimedia type traffic. The value of $\overline{\Delta}$ represents the most commonly used size of the ARQ loop transmission window, N_{Poll}, equal to 10 radio blocks.

It is not surprising that the ARQ component of average IP packet delay is the lowest in the case of MCS9 for the same radio conditions, $\overline{P}$. It is important to remember that when MCS9 is used it has to be supported by a very high C/I level, otherwise the error rate will be very high. In the case when a very high error rate is experienced by MCS9, it may be better to use a different MCS, eg:

MCS4 or MCS2. These MCSs can keep very low BLER for much lower C/I, but they also offer significantly lower throughput. This lower throughput with low error rate may perform worse than the high throughput offered by MCS9, even when there is a high error rate. Figure 6.7 shows such a case. It is assumed that all 3 MCSs operate within the same radio conditions and MCS2 experiences minimal error rate, BLER1=1%. The MCS4 BLER is higher and equal to 15% and finally MCS9 BLER is equal to 30%. In such a scenario it is better to transport all IP packet smaller then 150 Bytes using MCS2. All IP packets that are larger that 150 Bytes should be transmitted by MCS9, since for these packets this MCS offers the smaller ARQ delay component.

On the same figure 6.7, the performance of the LCC algorithm is compared to the simulation results. The graph shows that the difference between simulation results and LCC values is very small. This shows that, in the case of ARQ based IP packet delays that are expected in EGPRS networks, the LCC algorithm offers a good accuracy.

Initial error rate	$\overline{P}$			
$(BLER_1)$	p_1	p_2	p_3	p_e
$1 * 10^{-2}$	$99 * 10^{-2}$	$1 * 10^{-2}$	≈ 0	≈ 0
$5 * 10^{-2}$	$95 * 10^{-2}$	$4 * 10^{-2}$	$1 * 10^{-2}$	≈ 0
$1 * 10^{-1}$	$90 * 10^{-2}$	$9 * 10^{-2}$	$1 * 10^{-2}$	≈ 0
$15 * 10^{-1}$	$85 * 10^{-2}$	$12 * 10^{-2}$	$2 * 10^{-2}$	$1 * 10^{-2}$
$2 * 10^{-1}$	$80 * 10^{-2}$	$16 * 10^{-2}$	$3 * 10^{-2}$	$1 * 10^{-2}$
$2.5 * 10^{-1}$	$75 * 10^{-2}$	$19 * 10^{-2}$	$4 * 10^{-2}$	$2 * 10^{-2}$
$3 * 10^{-1}$	$70 * 10^{-2}$	$21 * 10^{-2}$	$6 * 10^{-2}$	$3 * 10^{-2}$

Table 6.1: *$\overline{P}$ vectors for different initial error rate*

6.3.2 Section Hybrid Type II/III ARQ

The estimation of expected $\overline{P}$ vectors for Hybrid Type II/III ARQ is much more difficult than for the Type I case. The relationships between BLER at each transmission attempts and the $\overline{P}$ vector is the same as in the case of Hybrid Type I ARQ, equation 6.1. However, since the previously unsuccessfully received radio blocks are stored at the receiver and are combined with new transmission attempts of these radio blocks, the relationship between BLERs is not as shown in equation 6.2. The radio block error process is not memoryless any more, and $BLER_2$ and $BLER_3$ cannot be calculated by simple multiplications. Instead, each BLER has to be calculated independently.

One way of achieving this is to run a link level simulation of the EGPRS air interface and plot a graph of BLER vs number of transmission attempts for a given static C/I. Such a graph can be found in [2] and is shown in figures 6.8

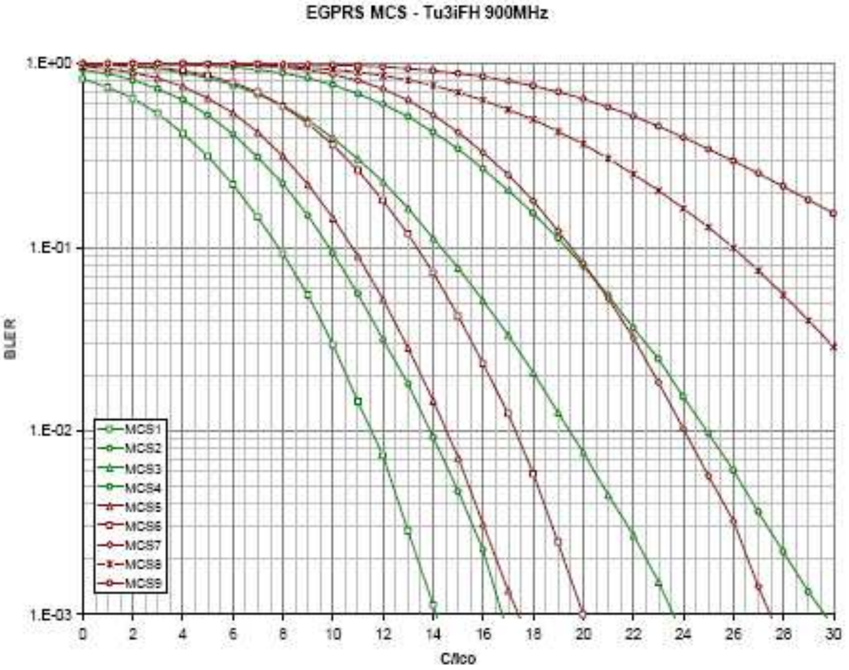


Figure 6.3: *BLER for different MCSs in EGPRS - TU3iFH 900MHz[3]*

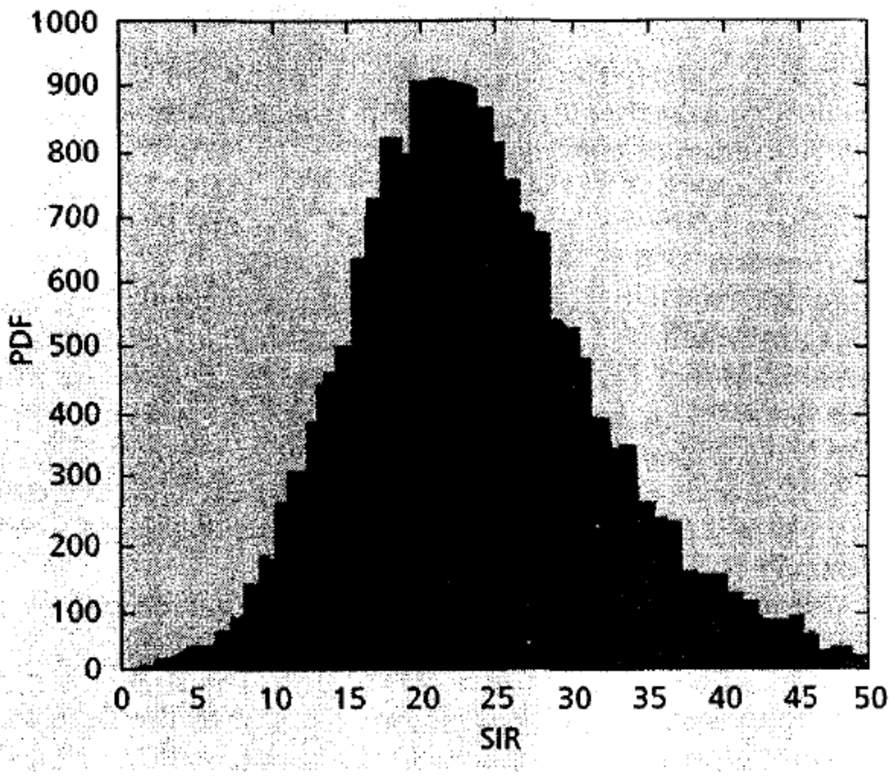
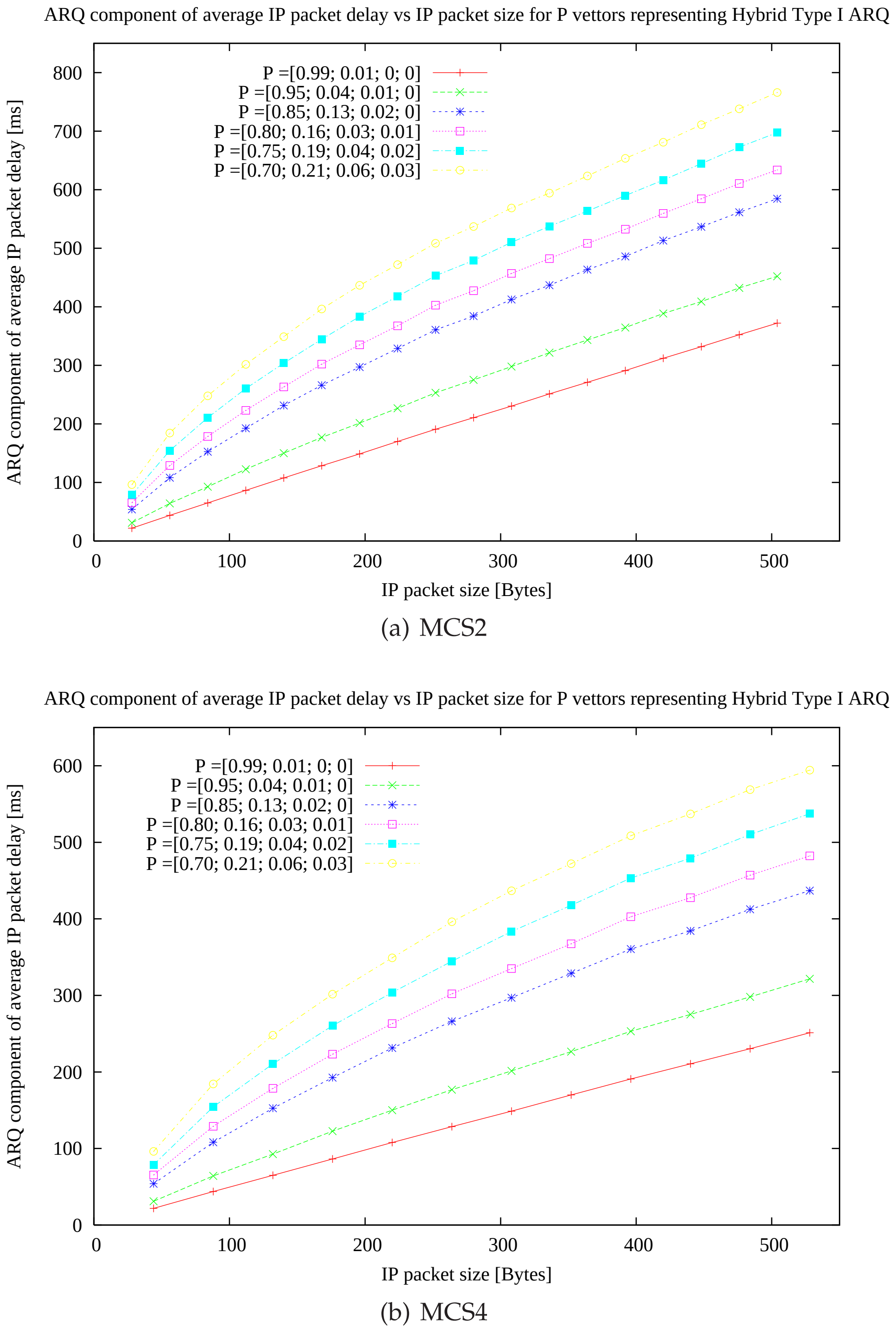


Figure 6.4: *A typical channel quality distribution for EDGE [10]*



(a) MCS2

(b) MCS4

Figure 6.5: *ARQ component of average IP packet delay - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P}$ *specified in Table 6.1 for MCS2(a) and MCS4(b).*

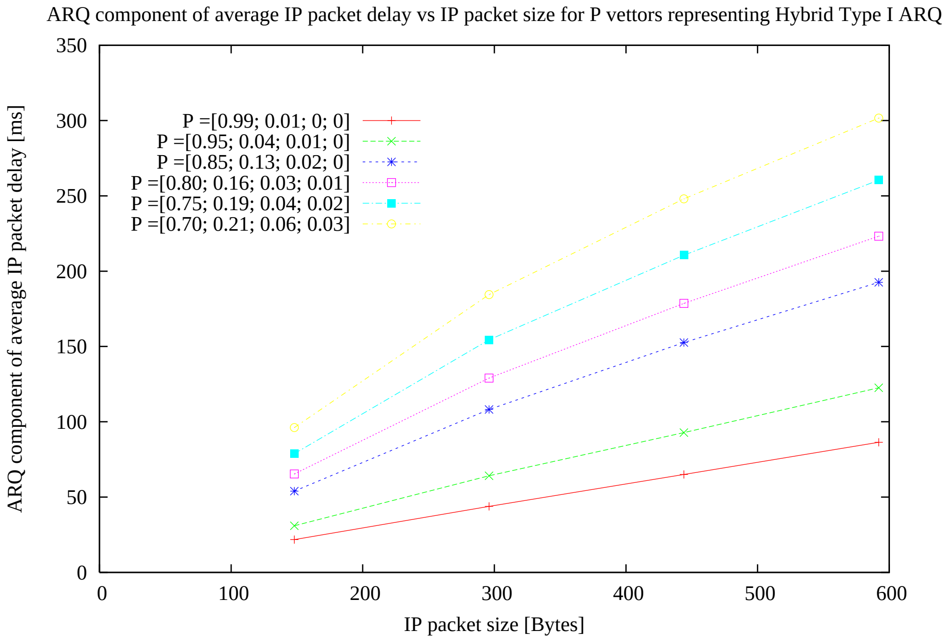


Figure 6.6: *ARQ component of average IP packet delay - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P}$ *specified in Table 6.1 for MCS9*

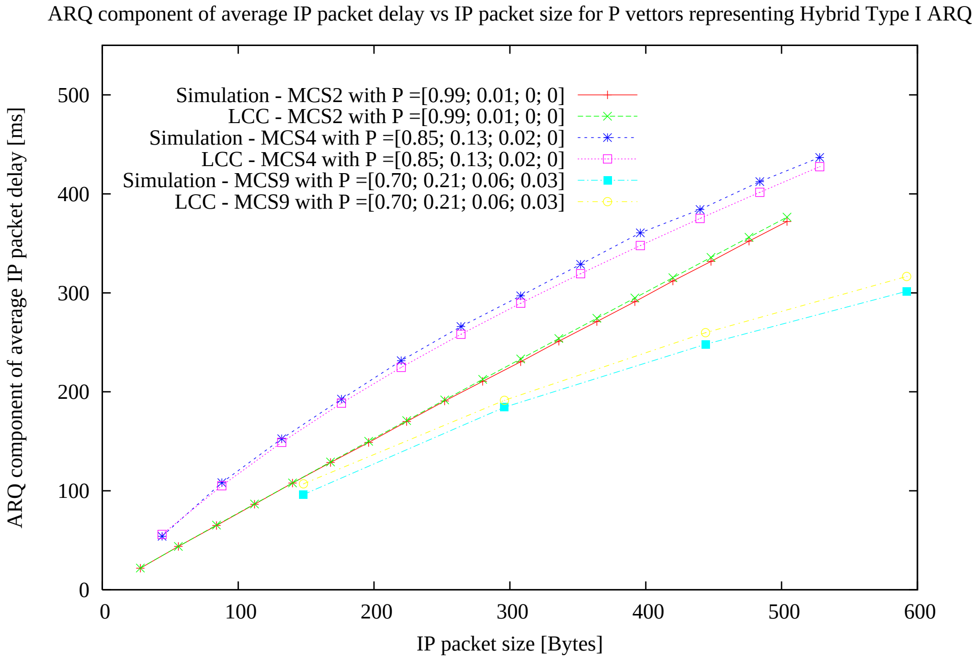


Figure 6.7: *ARQ component of average IP packet delay - comparison of simulation results with LCC for* $\overline{\Delta} = [0; 8; 13; 0]$ *and selected* $\overline{P}$ *and MCSs.*

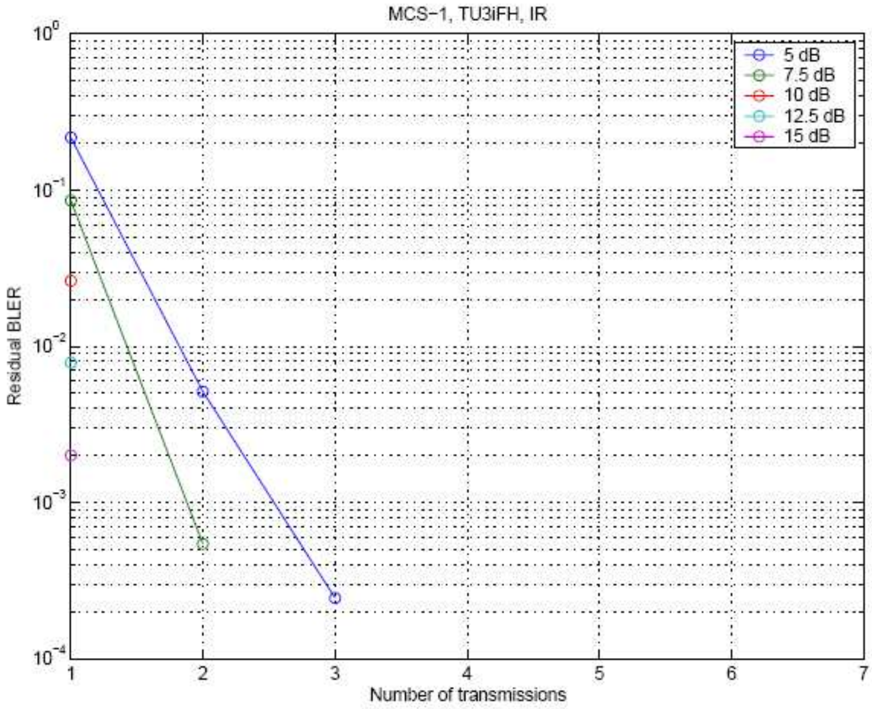


Figure 6.8: *BLER as a function of number of transmission attempts with IR [2]for MCS1, TU3iFH.*

and 6.9. Assuming that a static C/I is achieved for the transmission period it is possible to determine the BLER at each transmission attempt. It is obvious that the static channel is not achievable in the case of a mobile data network radio channel, but this section aims only to estimate possible shapes of the $\overline{P}$ vector.

Let's analyse the case of MCS5 with C/I of 7.5 dB. The BLER at each transmission attempt can be found from the figure 6.9, and are as follows:

$$\begin{aligned} BLER_1 &= 3 * 10^{-1}(30\%) \\ BLER_2 &= 2 * 10^{-2}(2\%) \\ BLER_3 &= 1.5 * 10^{-4}(\approx 0\%) \end{aligned} \tag{6.5}$$

Now, using these values for BLER and formula 6.1 it is possible to calculate the value of $\overline{P}$ vector representing this transmission scenario. The calculation is as follows:

$$\begin{aligned} p_1 &= 1 - BLER_1 = 0.7(70\%) \\ p_2 &= BLER_1 \cdot (1 - BLER_2) = 0.294(\approx 29\%) \\ p_3 &= BLER_1 \cdot BLER_2 \cdot (1 - BLER_3) = 0.006(\approx 1\%) \\ p_e &= BLER_1 \cdot BLER_2 \cdot (1 - BLER_3) = 0.0000009(\approx 0\%) \end{aligned} \tag{6.6}$$

Table 6.2 shows the calculated values of $\overline{P}$ vectors for MCS1 and MCS5 for a few different C/I values. It is interesting to know that $p_x > p_{x+1}$ does not always hold for the $\overline{P}$ vector. In the case of MCS5 with 5 dB C/I the shape of the distribution is different because, although only 40% of blocks are successfully received at

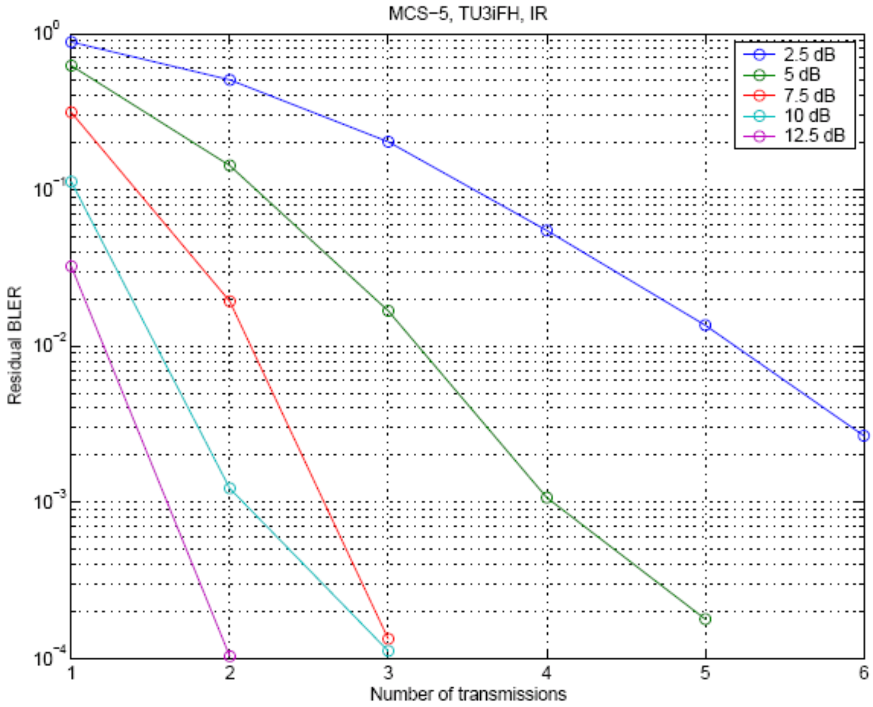


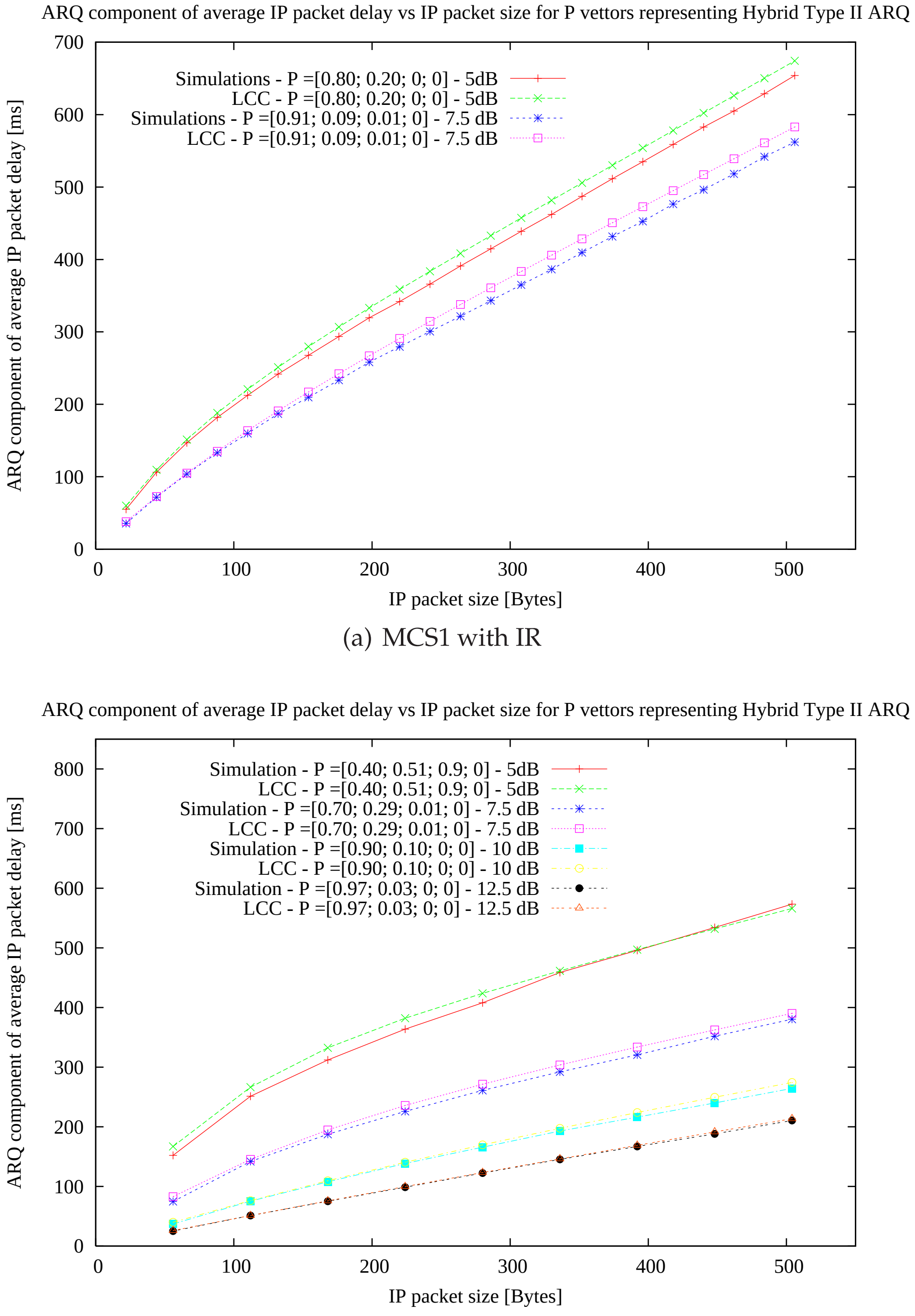
Figure 6.9: *BLER as a function of number of transmission attempts with IR for MCS5, TU3iFH[2].*

the first transmission, this value jumps to 91% after 2 retransmissions and to 100% after 3 transmission attempts. The new ARQ loop analysis method proposed in this thesis can easily capture this phenomenon by simple data logging. In fact it can work with any given shape of the $\overline{P}$ vector. When combined with the LCC method, it is possible to predict the ARQ component of average IP packet delay on the basis of any given $\overline{P}$ vector, as has been shown on chapter 5.

Having a few examples of possible $\overline{P}$ vectors, it is now possible to calculate this ARQ delay contribution when MCS1 and MCS5 are used with the BLER specified in Table 6.2. This is shown in figure 6.10 where the ARQ component of average IP packet delay is plotted as a function of IP packet size.

Since both graphs have $\overline{P}$ calculated on the basis of the same power, environment, terminal velocity etc, it is possible to compare their delay performance for different IP packet sizes. For example, in the case of a channel with C/I of 5dB, the use of MCS1 for small IP packets (up to 100 Bytes long) introduces a lower average IP delay component than in the case of MCS5. For bigger IP packet sizes, MCS1 has nearly the same performance as MCS5, whereas for much larger packet sizes, the MCS5 delay performance is superior to MCS1. When the C/I is equal to 7.5 dB, MCS5 is clearly superior to MCS1. This is due to the low error rate and higher throughput of MCS5.

The figure 6.10 also shows the accuracy of the LCC technique. For both, MCS1 and MCS5, the LCC technique predicted the ARQ component of IP packet delay accurately, as the curve of LCC prediction closely follows the simulation results.



(a) MCS1 with IR

(b) MCS5 with IR

Figure 6.10: *Average IP packet delay - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P}$ *specified in Table 6.2.*

MCSx	Initial error rate	$\overline{P}$			
	$(BLER_1)$	p_1	p_2	p_3	p_e
MCS5 - 12.5 dB	$3 * 10^{-2}$	$97 * 10^{-2}$	$3 * 10^{-2}$	≈ 0	≈ 0
MCS5 - 10 dB	$10 * 10^{-2}$	$90 * 10^{-2}$	$10 * 10^{-2}$	≈ 0	≈ 0
MCS5 - 7.5 dB	$30 * 10^{-2}$	$70 * 10^{-2}$	$29.4 * 10^{-2}$	$0.6 * 10^{-2}$	≈ 0
MCS5 - 5 dB	$60 * 10^{-2}$	$40 * 10^{-2}$	$51 * 10^{-2}$	$9 * 10^{-2}$	≈ 0
MCS1 - 7.5 dB	$9 * 10^{-2}$	$91 * 10^{-2}$	≈ 0	≈ 0	≈ 0
MCS1 - 5 dB	$20 * 10^{-2}$	$80 * 10^{-2}$	$20 * 10^{-2}$	≈ 0	≈ 0

Table 6.2: *$\overline{P}$ vectors for different initial error rate*

6.4 Conclusion

This chapter described the methodology of gathering input data$\overline{P}$, $\overline{\Delta}$ and k, for the LCC technique. Since it is generally very hard to have access to a real EGPRS network, a method of estimating the expected $\overline{P}$ vectors was shown. Following that, simulations were performed based on previously estimated $\overline{P}$ vectors and graphs showing the relationship between the ARQ component of average IP packet delay and the size of these packets were plotted. The results obtained from the simulations were then compared to the values predicted by the LCC analytical algorithm, which showed only small difference between the results obtained from simulations and from the LCC algorithm.

It is worth mentioning here that in the case of Hybrid Type II/III ARQ the BLER after three transmission attempts is very low. This validates the assumption about negligible error rate of radio blocks after three transmission attempts, made for the simulations performed in chapters 4 and 5. Additionally, it was shown that the LCC method can be used for both Hybrid Type I ARQ and Hybrid Type II/III ARQ. However, in the case of Type I ARQ the overall performance can be degraded by the slower LA algorithm necessary for collecting radio block error characteristic at the same MCS. This is not the case in Hybrid Type II/III ARQ where adaptation is done by the retransmission itself as each transmission attempt increases the error resilience of transmitted radio blocks.

CHAPTER 7

Conclusions and Future work

7.1 Conclusions

The work in this thesis has explored the influence of the ARQ loop on the delay of IP packets transmitted over mobile data networks.

A novel methodology for analysing ARQ loop delay performance has been proposed. This new approach allows testing of advanced ARQ techniques like Hybrid Type I/II/III ARQ with respect to their influence on the IP packet transmission delay. Its IP centric design allows use of the delay component introduced by an ARQ loop as a performance descriptor of the RLC layer. This descriptor may be useful in cross layer performance optimization, when higher layers adjust their congestion and flow control policies, like TCP, or change the size of their packets, eg: Multimedia Application that is capable of creating smaller or larger packets while maintaining the same throughput. Additionally, the input data representing the PHY layer ($\overline{P}$) ARQ loop delays ($\overline{\Delta}$) and size of transmitted IP packet (k) are easily gathered in a real system. This feature is important once the model is used beyond pure analysis of ARQ loop influence. When used for the prediction of the ARQ loop influence on the IP packet transmission, it is essential to be able to collect the input data in a simple manner.

A series of tests run on the computer based simulation model have revealed that the average value of IP packet delay caused by an ARQ loop can be considered as a good performance descriptor of the RLC layer for packets of medium or large size. This means that the packets should be larger than 6 radio blocks. On the contrary, in the case of small IP packets, the average delay has to be used with awareness of its potential misrepresentation. This is because the ARQ loop delay component distribution for a given IP packet size is in most cases non-uniform and non-Gaussian like in its shape. Thus, it is quite likely that the real delay experienced by the analysed packet will never have the value represented by the average delay. Therefore, the average value of this delay component can misinform adaptation mechanisms at higher protocol layers, if this average delay is used by them as a descriptor of the RLC layer delay performance. However, having an RLC delay performance descriptor can be useful. When small IP packets are transmitted the average delay ARQ related component should be accompanied by the additional message signalling limited trust to the average delay descriptor.

Another important finding is the non-linear behaviour of the relationship between the average ARQ component of IP packet delay and its size. It is often assumed that this relation has a linear characteristic, but this is true for only larger IP packet sizes. However, the average ARQ inherited delay between small IP packets has a higher slope than in the case of medium and large packets. The source of this comes from the domination of the ARQ loop retransmission delays over the delay caused by average radio block occupation of radio resources. The higher the retransmission feedback delays, the higher the domination of ARQ feedback for small IP packets. In the extreme case of ideal SR-ARQ, this ARQ feedback loop delay has no influence on the average delay, which holds a linear characteristic across the full range of IP packet sizes.

Based on the ARQ loop analysis described above, two methods for prediction of the ARQ loop component of average IP packet delay have been proposed.

The first method, called Brute Force (BF), is based on a careful analysis of all the major processes contributing to the ARQ based average IP packet delay. The results given by the BF method offer a good level of prediction accuracy. However, the computation complexity of the BF technique grows exponentially with the size of the analysed packet. This complexity limits the use of the BF method in a system that requires a real time prediction.

The second method, named Low Computation Complexity (LCC), uses the knowledge about a generic shape of the average delay as a function of IP packet size and the values of the average ARQ component of IP packet delay obtained from the BF method for small packet sizes. These delays that are computed for small packets are then used for curve fitting, to reduce complexity for medium and large IP packets. The error introduced by the LCC method is of the same order of magnitude as in the BF method. Nonetheless, the LCC algorithm complexity is significantly reduced.

Taking into account features like level of the prediction error, algorithm computation complexity and easy access to prediction input values it is concluded that the LCC algorithm makes a good candidate to be deployed in a real network.

7.2 Future Work

The following are possible directions to extend the work presented in this thesis:

Deployment of the LLC technique in an existing mobile data network. This would show how much the assumptions from the ARQ loop model being considered have to be relaxed to match results from a real network. Once this is done, the search for an accurate predictor of the $\overline{P}$ vector would be carried out. Following that, the accuracy of the prediction of the ARQ component of IP packet delay in a real system could be tested for a variety of radio channel scenarios, $\overline{P}$, and ARQ loop delays, $\overline{\Delta}$.

The use of the prediction of the ARQ component of average IP packet delay . The knowledge and experience from the previous paragraph would allow more advanced experiments aiming to utilise the expected ARQ loop influence on the IP packet delay to be run. These experiments can be classified into three groups, named after the protocol which could use the prediction given by the LCC algorithm. They are:

- **LCC based MAC enhancement:** The LCC prediction of the average ARQ loop component of IP packet delay can be used for allocating the amount of radio resources/radio channels given to a particular connection in order to maintain the pre-agreed level of QoS. Basically, according to the expected delay performance at the RLC layer, which represents a single MAC channel scenario, the analysed connection can get access to an appropriate amount of MAC channels. Thus, the network would be able to utilise its resources more efficiently, from the user demand point of view.
- **LCC based TCP enhancement:** The RLC delay descriptor could be used as a parameter for describing a lower delay bound at the wireless part of the peer-to-peer connection. This descriptor could be exploited to tune the behaviour of the TCP congestion avoidance mechanism, which often misinterprets the long segment delivery as a loss at the link layer.
- **LCC based Application enhancement:** The average IP packet delay can be used by an application to adapt the size of packets from voice and video codecs. The RLC delay descriptor shows the lower bound for the average IP packet delay. Thus, one of the ways to minimise the delay is to send voice or multimedia data encapsulated in IP packets with a smaller size, while keeping the throughput of this connection at a constant rate by increasing the rate of released packets.

The proposed model of ARQ loop analysis and the associated algorithms for the prediction of the ARQ loop component of average IP packet delay open a new perspective on the ARQ loop influence. The simplicity of the ARQ model and the LCC prediction algorithm allows them to be used and implemented in a real network. This is usually not the case with other models, which are often based on a detailed radio channel model and use sophisticated mathematical tools to describe the delay properties of the carried traffic. The works presented in this thesis, therefore offers a powerful new tool that can help to improve the performance of mobile data networks.

BIBLIOGRAPHY

[1] Not Just a Flash in the Pan. *The Economist - The Economist Technology Quarterly 2006*, 378(8468):23 – 25, March 2006.

[2] 3GPP. Bit rate and retransmission aspects for p-t-m MBMS in GERAN. In *3GPP TSG-GERAN nr.15; Tdoc GP-031200*, June 2003.

[3] 3GPP. SDU Requirements for PtM MBMS Radio Bearers. In *3GPP TSG-GERAN nr.15; Tdoc GP-031430*, June 2003.

[4] S. K. A. Demers and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proc. ACM SIGCOMM'89*, volume 19, pages 1–12, August 1989.

[5] M. C. C. A. DeSimone and O.-C. Yue. Throughput Performance of Transport-Layer Protocols over Wireless LANs. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM '93*, volume 1, pages 542–549, December - November 1993.

[6] R. T. C. A. Francini, F. M. Chiussi, K. D. Drucker, and N. E. Idirene. Enhanced weighted round robin schedulers for accurate bandwidth distribution in packet networks. *Computer Networks*, 37(5):561–578, November 2001.

[7] N. N. Aleksandar Neskovic and D. Paunovic. Indoor Electric Field Level Prediction Model Based on the Artificial Neural Networks. *IEEE Communications Letters*, 4(6):190 – 192, June 2000.

[8] N. N. Aleksandar Neskovic and G. Paunovic. Modern Approaches in Modeling of Mobile Radio Systems Propagation Environment. *IEEE Communication Surveys and Tutorials*, 3(3):2–12, Third Quarter 2000.

[9] A. D. J. Almudena Konrad, Ben Y. Zhao and R. Ludwig. A Markov-Based Channel Model Algorithm for Wireless Networks. *Wireless Networks*, 9(3):189 – 199, May 2003.

[10] F. M. Anders Furuskar, Sara Mazur and H. Olofsson. EDGE: Enhanced Data Rates for GSM and TDMA/136 Evolution. *IEEE Wireless Communications*, 6(3):56 – 66, June 1999.

[11] M. A. Arad and A. Leon-Garcia. Scheduled CDMA: A Hybrid Multiple Access for Wireless ATM Networks. In *Proc. Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'96*, volume 3, pages 913 – 917, October 1996.

[12] M. A. Arad and A. Leon-Garcia. A Generalized Processor Sharing Approach to Time Scheduling in Hybrid CDMA/TDMA. In *Proc. IEEE 17th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '98*, volume 3, pages 1164–1171, March – April 1998.

[13] P. S. K. Ashwin Sampath and J. M.Holtzman. Power Control and Resource Management for a Multimedia CDMA Wireless System. In *Proc. 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC'95*, volume 1, pages 21 – 25, September 1995.

[14] A. Bakre and B. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proc. 15th International Conference on Distributed Computing Systems, ICDCS'95*, pages 136 – 143, May-June 1995.

[15] H. Balakrishnan and R. Katz. ExplicitLoss Notification and Wireless Web Performance. In *Proc. IEEE GLOBECOM'98*, November 1998.

[16] J. C. Bennett and H. Zhang. WF^2Q: Worst-Case Fair Weighted Fair Queueing. In *Proc. 15th IEEE Joint Conference of the IEEE Computer Societies, INFOCOM '96*, volume 1, pages 120–128, March 1996.

[17] M. A. Bonuccelli and M. C. Clo. EDD Algorithm Performance Guarantee for Periodic Hard-Real-Time Scheduling in Distributed Systems. In *Proc. 13th International and 10th Symposium on Parallel and Distributed Processing, IPPS/SPDP'99*, pages 668 – 677, April 1999.

[18] D. H. K. T. Brahim Bensaou and K. T. Chan. Credit-Based Fair Queueing (CBFQ): A Simple Service-Scheduling Algorithm for Packet-Switched Networks. *IEEE/ACM Transactions on Networking*, 9(5):591 – 604, October 2001.

[19] L. S. Brakmo and L. L. Peterson. TCP Vegas: End-to-End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465 – 1480, October 1995.

[20] G. Brasche and B. Walke. Concepts, Services, and Protocols of the New GSM Phase 2+ General Packet Radio Service. *IEEE Communications Magazine*, 35(8):94 – 104, August 1997.

[21] K. Brown and S. Singh. M-TCP: TCP for Mobile Cellular Networks. *ACM SIGCOMM Computer Communication Review*, 27(5):19–43, July 1997.

[22] R. Caceres and L. Iftode. The Effects of Mobility on Reliable Transport Protocols. In *Proc. 14th International Conference on Distributed Computing Systems*, pages 12 – 20, June 1994.

[23] K. L. S. Carl Eklund, Roger B. Marks and S. Wang. IEEE Standard 802.16: A Technical Overview of the WirelessMAN^{TM} Air Interface for Wireless Access. *IEEE Communication Magazine*, 40(6):98–107, June 2002.

[24] L. Cerd and O. Casals. Study of the TCP Unfairness in a Wireless Environment. In *Proc. 8th IEEE International Conference on Telecommunications, ICT 2001*, June 2001.

[25] S. S. Chakraborty and M. Liinaharja. Performance Analysis of an Adaptive SR ARQ Scheme for Time-Varying Rayleigh Fading Channels. In *Proc. IEEE International Conference on Communications, ICC'01*, volume 8, pages 2478 – 2482, June 2001.

[26] J.-F. Chang and T.-H. Yang. End-to-End Delay of an Adaptive Selective Repeat ARQ Protocol. *IEEE Transactions on Communications*, 42(11):2926 – 2928, November 1994.

[27] B.-C. W. Chang-Fa Yang, Member IEEE and C.-J. Ko. A Ray-Tracing Method for Modeling Indoor Wave Propagation and Penetration. *IEEE Transactions on Antennas and Propagation*, 46(6):907 – 919, June 1998.

[28] H.-J. V. Christian Bettstetter and J. Eberspcher. GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface. *IEEE Communications Surveys*, 2(3):1 – 14, Third Quarter 1999.

[29] J. Chuang. The Effects of Time Delay Spread on Portable Radio Communications Channels with Digital Modulation. *IEEE Journal on Selected Areas in Communications*, 5(5):879 – 889, June 1987.

[30] A. M. W. Dongsoo Har and A. G. Chadney. Comment on Diffraction Loss of Rooftop-to-Street in COST 231-WalfischIkegami Model. *IEEE Transactions on Vehicular Technology*, 48(5):1451–1452, September 1999.

[31] S. L. Douglas N. Knisely, Sarath Kumar and S. Nanda. Evolution of Wireless Data Services: IS-95 to CDMA2000. *IEEE Communications Magazine*, 36(10):140 – 149, October 1998.

[32] A. Drukarev and D. J. C. Jr. Hybrid ARQ Error Control Using Sequential Decoding. *IEEE Transactions on Information Theory*, IT-29(4):521–535, July 1983.

[33] M. Elaoud and P. Ramanathan. TCP-SMART: A Technique for Improving TCP Performance in a Spottywide Band Environment. In *Proc. IEEE International Conference on Communications, ICC 2000*, volume 3, pages 1783–1787, June 2000.

[34] T. F. L. Ender Ayanoglu, Sanjoy Paul, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A Link-Layer Protocol for Wireless Networks. *ACM Wireless Networks*, 1(1):47 – 60, March 5 1995.

[35] K. Fall and S. Floyd. Simulation-based Comparisons of Tahoe, Reno and SACK TCP. *ACM Computer Communication Review*, 26(3):5–21, July 1996.

[36] R. Fantacci. Queuing Analysis of the Selective Repeat Automatic Repeat reQuest Protocol Wireless Packet Networks. *IEEE Transactions on Vehicular Technology*, 45(2):258 – 264, May 1996.

[37] R. Fantacci. Mean Packet Delay Analysis for the Selective Repeat Automatic Repeat Request Protocol with Correlated Arrivals and Deterministic and Nondeterministic Acknowledgement Delays. *Telecommunication Systems*, 9(1):41 – 57, March 1998.

[38] H. Fattah and C. Leung. An Overview of Scheduling Algorithms in Wireless Multimedia Networks. *IEEE Wireless Communications*, 9(5):76–83, October 2002.

[39] S. Floyd. TCP and Explicit Congestion Notification. *ACM Computer Communication Review*, 24(5):10–23, 1994.

[40] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE ACM Transactions on Networking*, 7(4):458–472, 1999.

[41] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. In *RFC 2582*, 1999.

[42] C. P. Fu and S. C. . Liew. TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks. *IEEE Journal on Selected Areas in Communications*, 21:216–228, February 2003.

[43] R. Ganesh and K. Pahlavan. Statistics of Short Time and Spatial Variations Measured Inwideband Indoor Radio Channels. *IEE Proceedings Microwaves, Antennas and Propagation*, 140(4):297–302, August 1993.

[44] P. A. S. Gerard J. M. Janssen and R. Prasad. Wideband Indoor Channel Measurements and BER Analysis of Frequency Selective Multipath Channels at 2.4, 4.75, and 11.5 GHz. *IEEE Transactions on Communications*, 44(10):1272–1288, October 1996.

[45] R. H. K. Giao T. Nguyen, Brian Noble and M. Satyanarayanan. A Trace-based Approach for Modeling Wireless Channel Behavior. In *Proc. Winter Simulation Conference Proceedings*, pages 597 – 604, December 1996.

[46] A. Glowacz and R. Chodorek. Behavior of TCP Westwood in Wireless Network. In *Proc. Advanced Technologies, Applications and Market Strategies for 3G and Beyond, ATAMS'2002*, December 2002.

[47] S. Golestani. A Self-clocked Fair Queueing Scheme for Broadband Applications. In *Proc. 13th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '94*, volume 2, pages 636–646, June 1994.

[48] S. Golestani. Network Delay Analysis of a Class of Fair Queueing Algorithms. *IEEE Journal on Selected Areas in Communications*, 13(6):1057–1070, August 1995.

[49] S. J. Golestani. Congestion-Free Transmission of Real-Time Traffic in Packet Networks. In *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'90*, pages 527–542, June 1990.

[50] J. Gozalvez and J. Dunlop. On the Dynamics of Link Adaptation Updating Periods for Packet Switched Systems. *Wireless Personal Communications, Kluwer Academic Publishers*, 23(1):137 – 145, October 2002.

[51] T. O. Gregory E. Bottomley and Y.-P. E. Wang. A Generalized RAKE Receiver for Interference Suppression. *IEEE Journal on Selected Areas in Communications*, 18(8):1536–1545, August 2000.

[52] O. Gurbuz and H. Owen. Dynamic Resource Scheduling Strategies for QoS in W-CDMA. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM '99*, volume 1A, pages 183 – 187, December 1999.

[53] R. Gurin and V. Peris. Quality-of-Service in Packet Networks: Basic Mechanisms and Directions. *Computer Networks*, 31(3):169–189, February 1999.

[54] K. Halford and M. Webster. Multipath measurement in wireless lans. In *Intersil Application Note AN9895*, October 2001.

[55] T. Halonen, J. Romero, and J. Melero. *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS*. John Wiley & Sons, Ltd, First edition, 2002.

[56] E. A. Hari Balakrishnan, Srinivasan Seshan and R. H. Katz. Improving TCI/IP Performance over Wireless Networks. In *Proc. 1'st International Conference on Mobile Computing and Networking*, pages 2 – 11, November 1995.

[57] S. S. Hari Balakrishnan and R. H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1(4):469 – 481, December 1995.

[58] S. S. Hari Balakrishnan, Venkata N. Padmanabhan and R. H. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6):756 –769, December 1997.

[59] M. Hata. Empirical Formula for Propagation Loss in Land Mobile Radio Services. *IEEE Transactions on Vehicular Technology*, 29(3):317–325, August 1980.

[60] S. Howard and K. Pahlavan. Doppler Spread Measurements of Indoor Radio Channel. *Electronics Letters*, 26(2):107–109, January 1990.

[61] D. Huang and J. J. Shi. Performance of TCP over Radio Link with Adaptive Channel Coding and ARQ. In *Proc. IEEE 49th Vehicular Technology Conference, VTC-1999*, volume 3, pages 2084–2088, Jul 1999.

[62] D. A. L. Ian F. Akyildiz and I. Joe. A Slotted CDMA Protocol with BER Scheduling for Wireless Multimedia Networks. *IEEE/ACM Transactions on Networking*, 7(2):146 – 158, April 1999.

[63] G. M. Ian F. Akyildiz and S. Palazzo. TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks. *IEEE/ACM Transactions on Networking*, 9(3):307–321, June 2001.

[64] ITU-T. End-user Multimedia QoS Categories. In *ITU-T Recommendation G.1010*, November 2001.

[65] J. I. J Dunlop and P. Cosimini. Estimation of the Performance of Link Adaptation in Mobile Radio Mode. In *Proc. 45th IEEE Vehicular Technology Conference, VTC'95*, volume 1, pages 326–330, July 1995.

[66] V. Jacobson. Congestion Avoidance and Control. In *ACM SIGCOMM'88*, pages 314–329, August 1988.

[67] A. A. L. Jay M. Hyman and G. Pacifici. Real-Time Scheduling with Quality of Service Constraints. *IEEE Journal on Selected Areas in Communications*, 9(7):1052–1063, September 1991.

[68] V. A. Jeonghoon MO, Richard J. La and J. Walrand. Analysis and Comparison of TCP Reno and Vegas. In *Proc. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '99*, volume 3, pages 1556 – 1563, March 1999.

[69] J.-C. C. Jui-Hung Yeh and C.-C. Lee. WLAN Standards. *IEEE Potentials*, 22(4):16 – 22, October - November 2003.

[70] G. M. K. Claffy and K. Thompson. The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone. In *Proc. The Internet Summit, INET'98*, July 1998.

[71] H. O. K. Takagaki and M. Murata. Analysis of a Window-based Flow Control Mechanism Based on TCP Vegas in Heterogeneous Network Environment. In *Proc. IEEE International Conference on Communications, ICC 2001*, volume 10, pages 3224–3228, June 2001.

[72] Y. T. Kai Xu and N. Ansari. TCP-Jersey for Wireless IP Communications. *IEEE Journal on Selected Areas in Communications*, 22(4):747–756, May 2004.

[73] C. R. Kalmanek and H. Kanakia. Rate Controlled Servers for Very High-Speed Networks. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM '90*, pages 12–20, December 1990.

[74] J. G. Kim and M. M. Krunz. Delay Analysis of Selective Repeat ARQ for a Markovian Source Over a Wireless Channel. *IEEE Transactions on Vehicular Technology*, 49(5):1968–1981, September 2000.

[75] K. C. K.K. Leung, P.F. Driessen and X. Qiu. Link Adaptation and Power Control for Streaming Services in EGPRS Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 19(10):2029 – 2039, October 2001.

[76] J. Kolakowski and J. Cichocki. *UMTS Systemy Telefonii Komorkowej Trzeciej Generacji*. Wydawnictwo Komunikacji i Lacznosci, First edition, 2003.

[77] A. G. Konheim. A Queueing Analysis of Two ARQ Protocols. *IEEE Transactions on Communications*, 28(7):1004 – 1014, July 1980.

[78] A. Kunz and C. Stepping. Overview and Implementation of Scheduling Algorithms for Wireless Environments. In *Proc. 5th European Personal Mobile Communications Conference*, pages 441 – 446, April 2003.

[79] J. H.-M. S. Kwok-Wai Cheung and R. D. Murch. A New Empirical Model for Indoor Propagation Prediction. *IEEE Transactions on Vehicular Technology*, 47(3):996 – 1001, August 1998.

[80] J. Lai and N. B. Mandayam. Performance of ReedSolomon Codes for Hybrid-ARQ over Rayleigh Fading Channels Under Imperfect Interleaving. *IEEE Transactions on Communications*, 48(10):1650 – 1659, October 2000.

[81] T. V. Lakshman and U. Madhow. The Performance of TCP/IP for Networks with High Bandwidth-delay Products and Random Lloss. *IEEE/ACM Transactions on Networking*, 5:336–350, 1997.

[82] M. C. Lawton and J. P. McGeehan. The Application of a Deterministic Ray Launching Algorithm for the Prediction of Radio Channel Characteristics in Small-Cell Environments. *IEEE Transactions on Vehicular Technology*, 43(4):955–969, November 1994.

[83] F. Lefevre and G. Vivier. Understanding TCPs Behavior over Wireless Links. In *Proc. Communications Vehicular Technology,SCVT-2000*, pages 123–130, October 2000.

[84] C. Liu and R. Jain. Congestion Coherence: A Local Enhancement for TCP over Wireless Links. In *Proc. 36th Hawaii International Conference on System Sciences, HICSS-36*, January 2003.

[85] S. F. M. Mathis, J. Mahdavi and A. Romanow. TCP Selective Acknowledgment Options. In *RFC 2018*, October 1996.

[86] D. M. Mandelbaum. An Adaptive-Feedback Coding Scheme Using Incremental Redundancy. *IEEE Transactions on Information Theory*, 20(3):388 – 389, 1974.

[87] U. K. Matthias Patzold and F. Laue. A Deterministic Digital Simulation Model for Suzuki Processes with Application to a Shadowed Rayleigh Land Mobile Radio Channel. *IEEE Transactions on Vehicular Technology*, 45(2):318–331, May 1996.

[88] U. K. Matthias Patzold and F. Laue. An Extended Suzuki Model for Land Mobile Satellite Channels and its Statistical Properties. *IEEE Transactions on Vehicular Technology*, 47(2):617–630, May 1998.

[89] L. B. Michele Rossi and M. Zorzi. Exact Statistics of ARQ Packet Delivery Delay over Markov Channels with Finite Round-trip Delay. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM '03*, volume 6, pages 3356 – 3360, December 2003.

[90] R. R. R. Michele Zorzi and L. B. Milstein. ARQ Error Control for Fading Mobile Radio Channels. *IEEE Transactions on Vehicular Technology*, 46(2):445–455, May 1997.

[91] E. N. P. Miltiades E. Anagnostou. Performance Analysis of the Selective Repeat ARQ Protocol. *IEEE Transactions on Communications*, COM-34(2), February 1986.

[92] D. Molkdar. Review on Radio Propagation into and within Buildings. *IEE Proceedings Microwaves, Antennas and Propagation*, 138(1):61 – 73, February 1991.

[93] D. Molkdar and W. Featherstone. Impact of Imperfect Link Adaptation in EGPRS. In *Proc. 2'nd IEE International Conference on 3G Mobile Communication Technologies*, pages 277–281, March 2001.

[94] R. Morrow. *Bluetooth: Operation and Use*. McGraw-Hill Education, First edition, June 2002.

[95] R. Morrow. *Wireless Network Coexistence*. McGraw-Hill Education, First edition, August 2004.

[96] J. L. P. Vlach, B. Segal and T. Pavlasek. Cross-Floor Signal Propagation Inside a Contemporary Ferro-Concrete Building at 434, 862, and 1705 MHz. *IEEE Transactions on Antennas and Propagation*, 47(7):1230–1232, July 1999.

[97] P. N. A. Pablo Jose Ameigeiras GutiCrrez, Jeroen Wigard, H. C. Damgaardt, and P. Mogensen. Performance of Link Adaptation in GPRS Networks. In *Proc. 52'nd IEEE Vehicular Technology Conference, VTS-Fall 2000*, volume 2, pages 492–499, September 2000.

[98] P. N. A. Pablo Jose Ameigeiras Gutierrez, Jeroen Wigard, H. C. Damgaardt, and P. Mogensen. Performance of Link Adaptation in GPRS Networks. In *Proc. 52nd IEEE Vehicular Technology Conference, VTC'00*, volume 2, pages 492–499, Fall 2000.

[99] A. Parekh and R. Gallager. A Generalized Processor Sharing Approach to Flow Control InintegratedSservices Networks: The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[100] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[101] L. Pavilanskas. Analysis of TCP Algorithms in the Reliable IEEE 802.11b Link. In *Proc. 19TH EUROPEAN CONFERENCE ON MODELLING AND SIMULATION, ECMS 2005*, June 2005.

[102] P. Piggin. WiMAX in-depth. *IEE Communications Engineer*, pages 36–39, Oct/Nov 2004.

[103] I. Poole. What exactly is ... VoIP. *IEE Communications Engineer*, pages 44–45, April/May 2005.

[104] I. Poole. What exactly is ... cdma2000. *IEE Communications Engineer*, pages 46–47, February/March 2006.

[105] A. K. Pravin Bhagwa, Partha Bhattacharya and S. K. Tripathit. Enhancing Throughput over Wireless LANs Using Channel Statedependent Packet Scheduling. In *Proc. 15th IEEE Annual Joint Conference of the IEEE Computer Societies, INFOCOM '96*, volume 3, pages 1133–1140, March 1996.

[106] J. Proakis. On the Probability of Error for Multichannel Reception of Binary Signals. *IEEE Transactions on Communications*, 16(1):68–71, February 1968.

[107] M. B. Pursley and S. D. Sandberg. Incremental-Redundancv Transmission for Meteor - Burst Communications. *IEEE Transactions on Communications*, 39(5):689 – 702, May 1991.

[108] V. C. M. L. Qixiang Pang, Amir Bigloo and C. Scholefield. Service Scheduling for General Packet Radio Service Classes. In *Proc. IEEE Wireless Communications and Networking Conference, WCNC'99*, volume 3, pages 1229 – 1233, September 1999.

[109] P. Ramanathan and P. Agrawal. Adapting Packet Fair Queueing Algorithms to Wireless Networks. In *Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'98*, pages 1 – 9, October 1998.

[110] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Second edition, 2001.

[111] T. S. Rappaport and S. Saridlrii. Radio-Wave Propagation for Emerging Wireless Personal-Communication Systems. *IEEE Antennas and Propagation Magazine*, 36(5):14–24, October 1994.

[112] L. K. Rasmussen and S. B. Wicker. The Performance of Type-I Trellis Coded Hybrid-ARQ Protocols over AWGN and Slowly Fading Channels. *IEEE Transactions on Information Theory*, 40(2):418–428, March 1994.

[113] H. Z. Robert J. C. Bultitude, Pierre Melanqon, G. Morrison, and M. Prokki. The Dependence of Indoor Radio Channel Multipath Characteristics on Transmit/Receive Ranges. *IEEE Journal on Selected Areas in Communications*, 11(7):979–990, September 1993.

[114] N. Samaraweera and G. Fairhurst. Explicit Loss Indication and Accurate RTO Estimation for TCP Error Recovery Using Satellite Links. *IEE Proceedings in Communications*, 144(1):47–5, February 1997.

[115] J. Sau and C. Scholefield. Scheduling and Quality of Service in the General Packet Radio Service. In *Proc. IEEE International Conference on Universal Personal Communications, ICUPC '98*, volume 2, pages 1067 – 1071, October 1998.

[116] M. G. M. Y. S. Saverio Mascolo, Claudio Casetti and R. Wang. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In *Proc. Seventh Annual International Conference on Mobile computing, Mobicom 2001*, pages 287 – 297, July 2001.

[117] M. Shreedhar and G. Varghese. Efficient Fair Queuing Using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, June 1996.

[118] D. J. C. J. Shu Lin and M. J. Miller. Automatic-RepeatreQuest Error-Control Schemes. *IEEE Communications Magazine*, 22(12):5 – 17, December 1984.

[119] M. K. Simon and M.-S. Alouini. A Unified Approach to the Performance Analysis of Digital Communication over Generalized Fading Channels. *Proceedings of the IEEE*, 86(9):1860 – 1877, September 1998.

[120] H.-W. Son and N.-H. Myung. A Deterministic Ray Tube Method for Microcellular Wave Propagation Prediction Model. *IEEE Transactions on Antennas and Propagation*, 47(8):1344–1350, August 1999.

[121] T. N. Songwu Lu and V. Bharghavan. Design and Analysis of An Algorithm for Fair Service in Error-Prone Wireless Channels. *ACM Journal of Wireless Networks*, 6(4):323 – 343, July 2000.

[122] V. B. Songwu Lu and R. Srikant. Fair Scheduling in Wireless Packet Networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, August 1999.

[123] V. B. Songwu Lu, Thyagarajan Nandagopal. A Wireless Fair Service Algorithm for Packet Cellular Networks. In *Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'98*, pages 10 – 20, October 1998.

[124] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms. *IETF RFC 2001*, 1997.

[125] P. Stuckmann. Quality of Service Management in GPRS-Based Radio Access Networks. *Telecommunication Systems*, 19(3):515–546, April 2002.

[126] P. Stuckmann and O. Moller. Advanced Scheduling and Admission Control Techniques for Cellular Packet Radio Networks. In *Proc. 5th European Personal Mobile Communications Conference*, pages 214 – 21, April 2003.

[127] Y. G. Sudhir Dixit and Z. Antoniou. Resource Management and Quality of Service in Third-Generation Wireless Networks. *IEEE Communications Magazine*, 39(2):125 – 133, February 2001.

[128] I. S. T. S. Eugene Ng and H. Zhang. Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors. In *Proc. 17th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM '98*, volume 3, pages 1103 – 1111, March - April 1998.

[129] C. L. T. T. Tjhung and N. P. Secord. BER Performance of DQPSK in Slow Rician Fading. *Electronics Letters*, 28(18):1763–1765, August 1992.

[130] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, Fourth edition, 2003.

[131] C. Tang and V. Stolpman. An Adaptive Learning Approach to Adaptive OFDM. In *Proc. IEEE Wireless Communications and Networking Conference, WCNC'04*, volume 3, pages 1406–1410, March 2004.

[132] K. K. Tapan K. Sarkar, Zhong Ji, A. Medour, and M. Salazar-Palma. A Survey of Various Propagation Models for Mobile Communication. *IEEE Antennas and Propagation Magazine*, 45(3):51 – 82, June 2003.

[133] V. Tsaoussidis and I. Matta. Open Issues on TCP for Mobile Computing. *Wireless Communications and Mobile Computing*, 2(1):3–20, February 2002.

[134] J. B. U. Hengartner and T. Gross. TCP Vegas Revisited. In *Proc. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000*, volume 3, pages 1546 – 1555, March 2000.

[135] X. G. V. Tsaoussidis, H. Badr and K. Pentikousi. Energy/Throughput Tradeoffs of TCP Error Control Strategies. In *Proc. 5th IEEE Symposium on Computers and Communications, ISCC'00*, pages 106 – 112, 2000 July.

[136] F. Vatalaro and G. E. Corazza. Probability of Error and Outage in a Rice-Lognormal Channel for Terrestrial and Satellite Personal Communications. *IEEE Transactions on Communications*, 44(8):921 – 924, August 1996.

[137] C. V. B. Vincent J. Hernandez, Antonio J. Mendez, R. M. Gagliardi, and W. J. Lennon. Bit-Error-Rate Analysis of a 16-User Gigabit Ethernet Optical-CDMA (O-CDMA) Technology Demonstrator Using Wavelength/Time Codes. *IEEE Photonics Technology Letters*, 17(12):2784 – 2786, December 2005.

[138] K. Wesolowski. *Mobile Communication Systems*. John Wiley & Sons, Ltd, First edition, January 2002.

[139] K. Wesolowski. *Podstawy Cyfrowych Systemow Telekomunikacyjnych*. Wydawnictwo Komunikacji i Lacznosci, First edition, 2003.

[140] S. B. Wicker. High-Reliability Data Transfer Over the Land Mobile Radio Channel Using Interleaved Hybrid-ARQ Error Control. *IEEE Transactions on Vehicular Technology*, 39(1):48 – 55, 1990.

[141] S. B. Wicker. Reed-Solomon Error Control Coding for Rayleigh Fading Channels with Feedback. *IEEE Transactions on Vehicular Technology*, 41(2):124–133, May 1992.

[142] S. B. Wicker and M. J. Bartz. Type-ii Hybrid - ARQ Protocols Using Punctured MDS Codes. *IEEE Transactions on Communications*, 42(234):1431 – 1440, Feb/Mar/Apr 1994.

[143] E. H.-K. Wu and M.-Z. Chen. JTCP: Jitter-based TCP for Heterogeneous Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 22(4):757– 766, May 2004.

[144] E. O. Y. Okumura and K. Fukuda. Field Strength and its Variability in VHF and UHF Land Mobile Service. *Review Electrical Communication Laboratory*, 16(9-10):825–873, 1968.

[145] J. R. Yee and J. Edward J. Weldon. Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel. *IEEE Transactions on Communications*, 43(8):2316 – 2323, 1995.

[146] K.-W. Yip and T.-S. Ng. A Simulation Model for Nakagami-m Fading Channels, $m < 1$. *IEEE Transactions on Communications*, 48(2):214–221, February 2000.

[147] H. Zhang. Service Disciplines for Guaranteed Performance Service in Packet-Switching Networks. *Proceedings of the IEEE*, 83(10):1374–1396, October 1995.

[148] H. Zhang and D. Ferrari. Rate-Controlled Static-Priority Queueing. In *Proc. 12th Annual IEEE Joint Conference of Computer and Communications Societies, INFOCOM '93*, volume 1, pages 227 – 236, March - April 1993.

[149] H. Zhang and S. Keshav. Comparison of rate-based service disciplines. In *Proc. ACM SIGCOMM'91*, pages 113 – 121, September 1991.

[150] L. Zhang. Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks. In *Proc. ACM SIGCOMM'90*, pages 19 –29, September 1990.

[151] K. Z. Zhefeng Song and Y. L. Guan. Statistical Adaptive Modulation for QAM-OFDM Systems. In *Proc. IEEE Global Telecommunications Conference, GLOBECOM'02*, volume 1, pages 706– 710, November 2002.

[152] M. Zorzi and R. R. Rao. On the Statistics of Block Errors in Bursty Channels. *IEEE Transactions on Communications*, 45(6):660 – 667, June 1997.

[153] M. Zorzi and R. R. Rao. On Channel Modeling for Delay Analysis of Packet Communications over Wireless Links. In *Proc. 36th Annual Allerton Conference on Communications, Control and Computing*, September 1998.

[154] M. Zorzi and R. R. Rao. Lateness of Probability of a Retransmission Scheme for Error Control on a Two-State Markov Channel. *IEEE Transactions on Communications*, 47(10):1537–1548, October 1999.

[155] M. Zorzi and R. R. Rao. Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks. *IEEE Personal Communications*, 6(5):32–40, October 1999.

APPENDIX A

Approximation of the $\overline{\Delta f(l)}$ from LCC method

The approximation of the $f(k)$ function proposed in section 5.3.2 has the following form:

$$\overline{f(k)} = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{\Delta f(l)}\right) \tag{A.1}$$

Since the $f_{BF}(1)$ is accessible from BF method, the only unknown in the equation above is the $\overline{\Delta f(l)}$ function. Where, $\overline{\Delta f(l)} = \overline{f(l)} - \overline{f(l-1)}$.

As k is an integer, the derivative of $f(k)$ is $\Delta f(k)$, described by:

$$\Delta f(k) = f(k) - f(k-1) \tag{A.2}$$

Thus, the $\Delta f(k)$ can take the place of $\Delta f(l)$, as by replacing k by l they have the same form. Consequently, $\overline{\Delta f(l)}$ can be replaced by $\overline{\Delta f(k)}$.

By observation of the shape of $\Delta f(k)$, figure A.1, it is expected that a reasonable curve fit can be achieved using the following function:

$$\Delta f(k) = B + C \cdot e^{-D \cdot k} \tag{A.3}$$

Subsequently, the approximation of the $\Delta f(k)$ function,$\overline{\Delta f(k)}$, chosen in this work has the following representation:

$$\overline{\Delta f(k)} = \overline{B} + \overline{C} \cdot e^{-\overline{D} \cdot k} \tag{A.4}$$

There is a number of functions that can be used for this approximation. This particular function have been chosen due to its similarity to the function $\Delta f(k)$ shown in figure A.1. Additionally, the offset component (B) can be easily calcu-

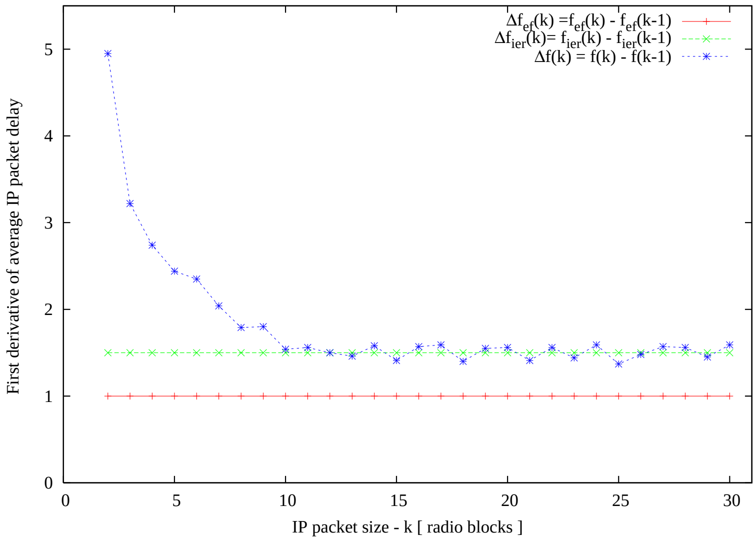


Figure A.1: *Example of first derivative of average IP packet delay - results for* $\overline{P}$ *=(0.6;0.3;0.1;0)* $\overline{\Delta}$ *=(0;8;13;0)*

lated as it matches the scenario of *ideal SR-ARQ* transmission($\Delta f_{ier}(k)$), which is presented in figure A.1. There was also an element of an intuitive choice associated with the selection of the curve fitting function form. This policy paid off, as the error level introduced by the chosen function turned out to be low, which is shown later that this thesis.

Since the B represents the static part of the $f(k)$ function progress, it is associated with the average occupation of the radio resources by radio packets. This can be easily calculated based on the $\overline{P}$ vector, representing the distribution of the transmission attempts for the whole radio blocks population. Hence, the approximation of B, $\overline{B}$, has the following form:

$$\overline{B} = \sum_{u=1}^{3} (p_u \cdot u) \tag{A.5}$$

As the BF method can offer an estimation of the $f(k)$ function,$\overline{f(k)}$, for small IP packet sizes,$k \in \{2, 3, 4, 5, 6\}$, these values can be used to approximate the $\Delta f(k)$. Therefore,

$$\overline{\Delta f(k)} = \Delta f_{BF}(k) = f_{BF}(k) - f_{BF}(k-1) \tag{A.6}$$

Hence, by merging equations A.4 and A.6:

$$\Delta f_{BF}(k) = \overline{B} + \overline{C} \cdot e^{-\overline{D} \cdot k} \tag{A.7}$$

The only unknown remaining at this moment are $\overline{C}$ and $\overline{D}$. In order to compute their values two small IP packet sizes are going to be chosen. Having done that, two independent equations emerged and they will be used to find numerical values of $\overline{C}$ and $\overline{D}$. Let's denote these two small IP packet sizes as m and n.

For $k = m$,

$$\Delta f_{BF}(m) = \overline{B} + \overline{C} \cdot e^{-\overline{D} \cdot m} \tag{A.8}$$

And for $k = n$,

$$\Delta f_{BF}(n) = \overline{B} + \overline{C} \cdot e^{-\overline{D} \cdot n} \tag{A.9}$$

Following that a number of mathematical operations are going to be performed to find equations for $\overline{C}$ and $\overline{D}$.

$$ln\left(\Delta f_{BF}(m) - \overline{B}\right) = ln\left(\overline{C}\right) + ln\left(e^{-\overline{D} \cdot m}\right) \tag{A.10}$$

$$ln\left(\Delta f_{BF}(n) - \overline{B}\right) = ln\left(\overline{C}\right) + ln\left(e^{-\overline{D} \cdot n}\right) \tag{A.11}$$

Rearranging A.10,

$$ln\left(\overline{C}\right) = ln\left(\Delta f_{BF}(m) - \overline{B}\right) - ln\left(e^{-\overline{D} \cdot m}\right) \tag{A.12}$$

Substituting A.12 into A.11,

$$ln\left(\Delta f_{BF}(n) - \overline{B}\right) = ln\left(\Delta f_{BF}(m) - \overline{B}\right) - ln\left(e^{-\overline{D} \cdot m}\right) + ln\left(e^{-\overline{D} \cdot n}\right) \tag{A.13}$$

So that,

$$ln\left(\Delta f_{BF}(n) - \overline{B}\right) - ln\left(\Delta f_{BF}(m) - \overline{B}\right) = \overline{D} \cdot m - \overline{D} \cdot n \tag{A.14}$$

Thus,

$$ln\left(\tfrac{\Delta f_{BF}(n) - \overline{B}}{\Delta f_{BF}(m) - \overline{B}}\right) = \overline{D} \cdot (m - n) \tag{A.15}$$

And,

$$\overline{D} = \frac{ln\left(\frac{\Delta f_{BF}(n) - \overline{B}}{\Delta f_{BF}(m) - \overline{B}}\right)}{m - n} \tag{A.16}$$

Then, by substituting A.16 into A.8 and basic rearrangements:

$$\overline{C} = \tfrac{\Delta f_{BF}(m) - \overline{B}}{e^{-\overline{D} \cdot m}} \tag{A.17}$$

Because the values of $\overline{C}$ and $\overline{D}$ will differ slightly for different pair of m and n, affecting the accuracy of the $\overline{f(k)}$, the special index indicating which points have been used for calculation of $\overline{C}$ and $\overline{D}$ will be added to $\overline{C}$ and $\overline{D}$ resulting in:

$$\begin{aligned}\overline{D_{m,n}} &= \frac{ln\left(\frac{\Delta f_{BF}(n)-\overline{B}}{\Delta f_{BF}(m)-\overline{B}}\right)}{m-n} \\ \overline{C_{m,n}} &= \frac{\Delta f_{BF}(m)-\overline{B}}{e^{-\overline{D_{m,n}}\cdot m}}\end{aligned} \tag{A.18}$$

Finally, the estimation of the tunction $f(k)$, $\overline{f(k)}$, based on the points $\{m, f_{BF}(m)\}$ and $\{n, f_{BF}(n)\}$ has the form presented below:

$$f_{CD:m,n}(k) = f_{BF}(1) + \sum_{l=2}^{k} \left(\overline{B} + \overline{C_{m,n}} \cdot e^{-\overline{D_{m,n}}\cdot l}\right) \tag{A.19}$$

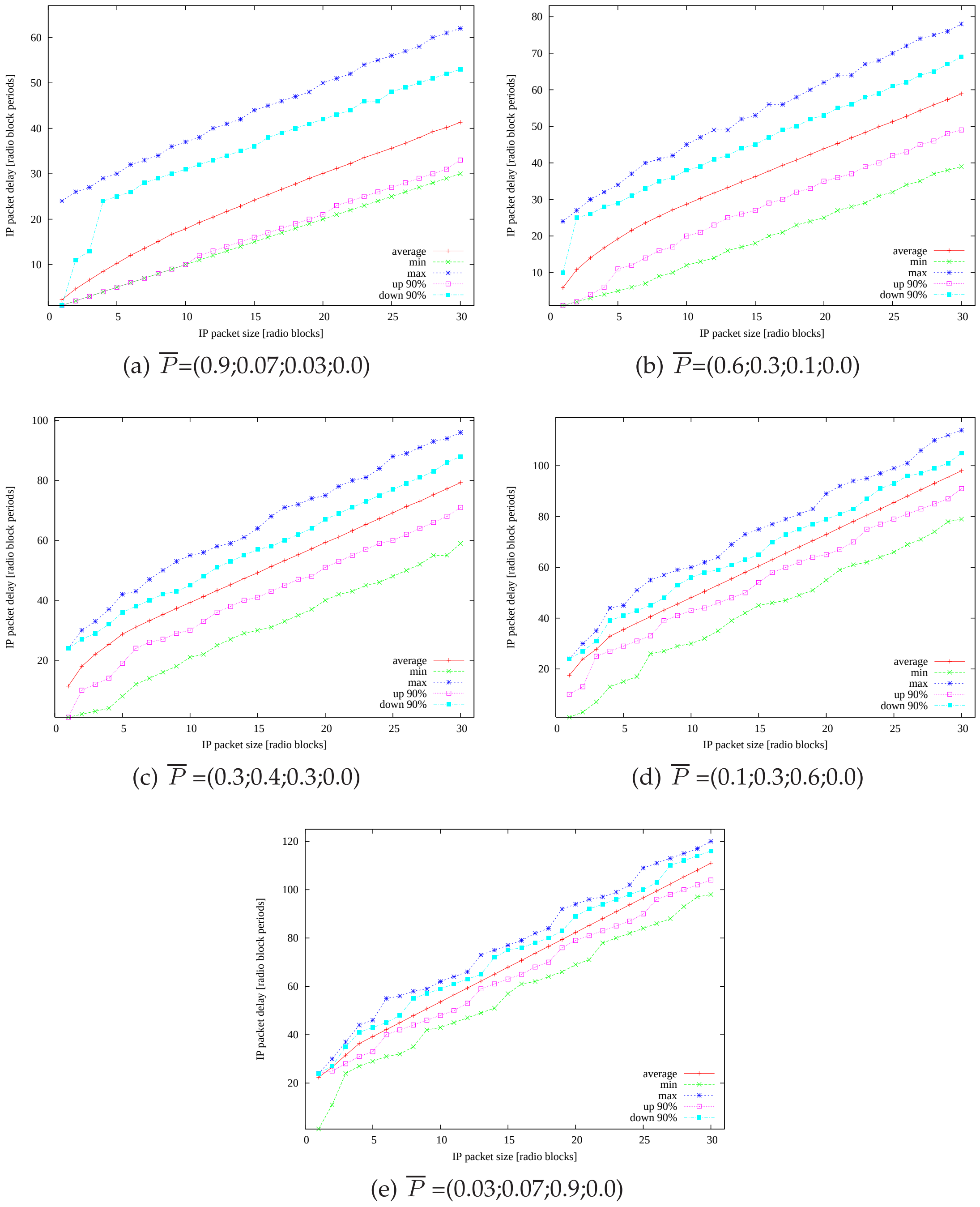
The above equation, equation A.19, is used in chapter 5 as the main part of Low Computation Complexity method.

APPENDIX B

Average IP packet delay and min, max, upper 90% and lover 90% bounds

This appendix presents the results of a comparison between the average IP packet delay and patterns of minimum and maximum values of delay experienced by a packet of a certain size, k. Additionally, two trends exposing upper and lower 90% result bounds are plotted. These additional lines illustrate how far most of the results are spread from the mean value, for a given packet size within an analysed $\overline{P}$ and $\overline{\Delta}$.

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

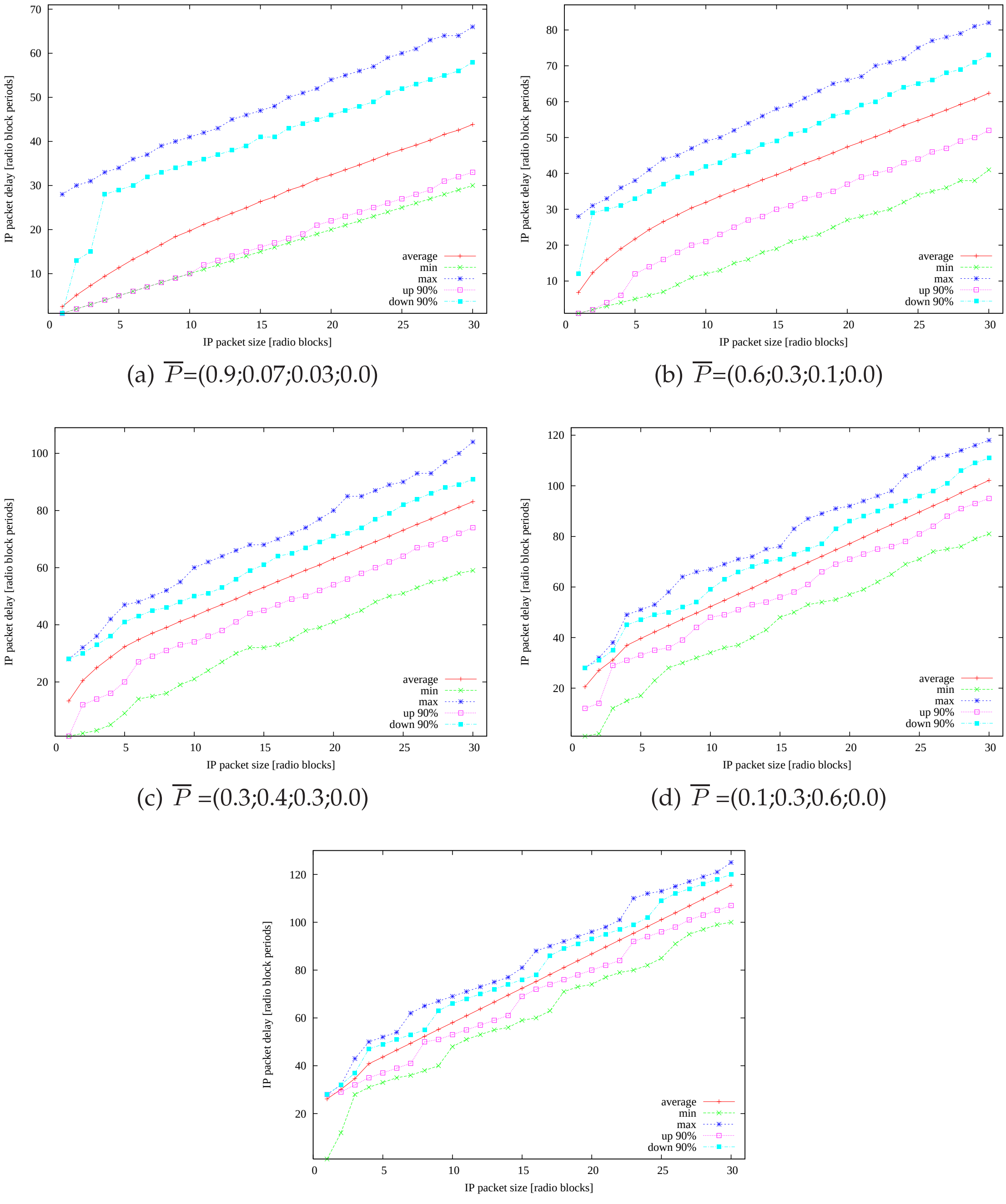
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.1: *Average IP packet delay and min, max, upper 90% anl lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

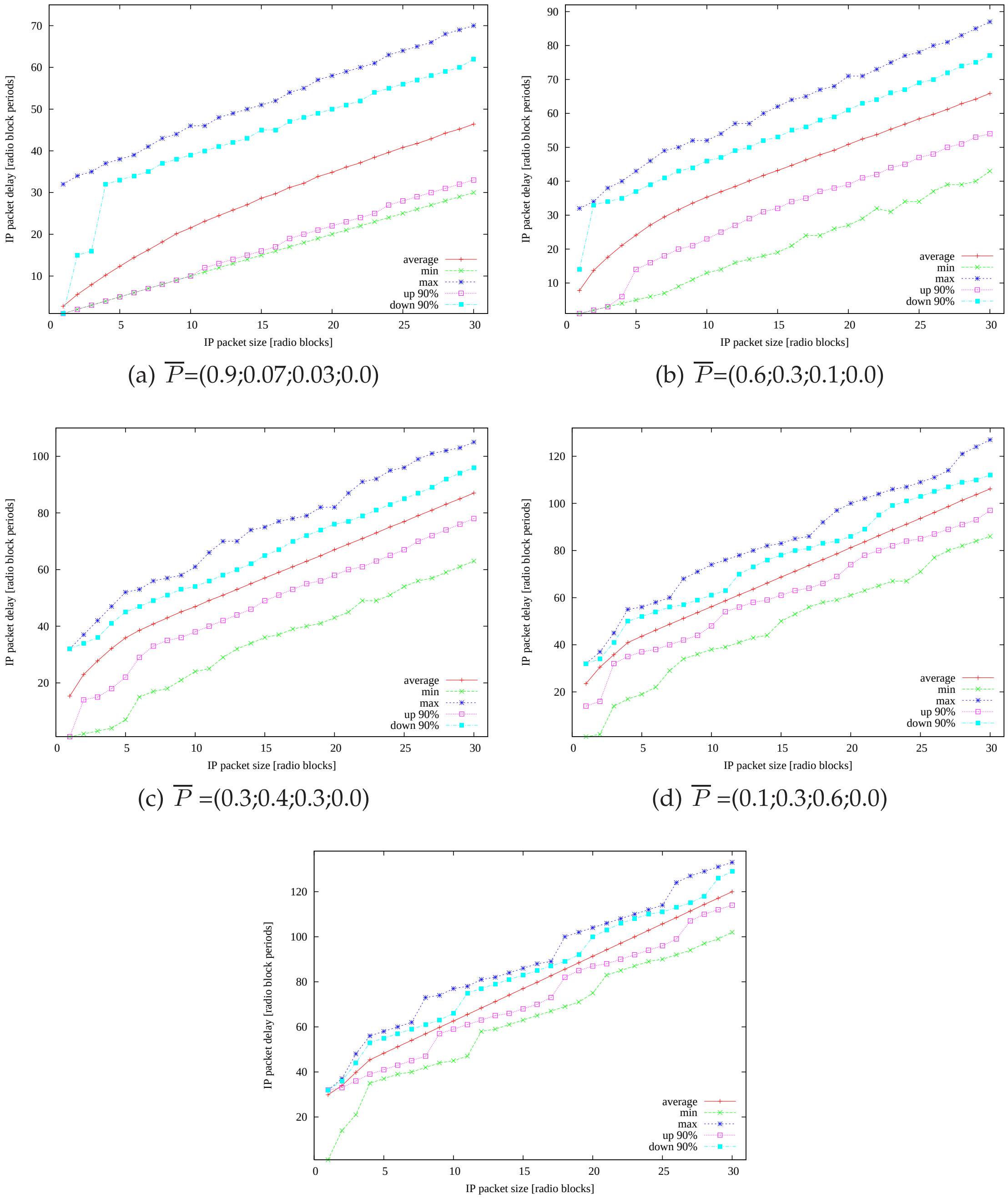
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.2: *Average IP packet delay and min, max, upper 90% anl lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

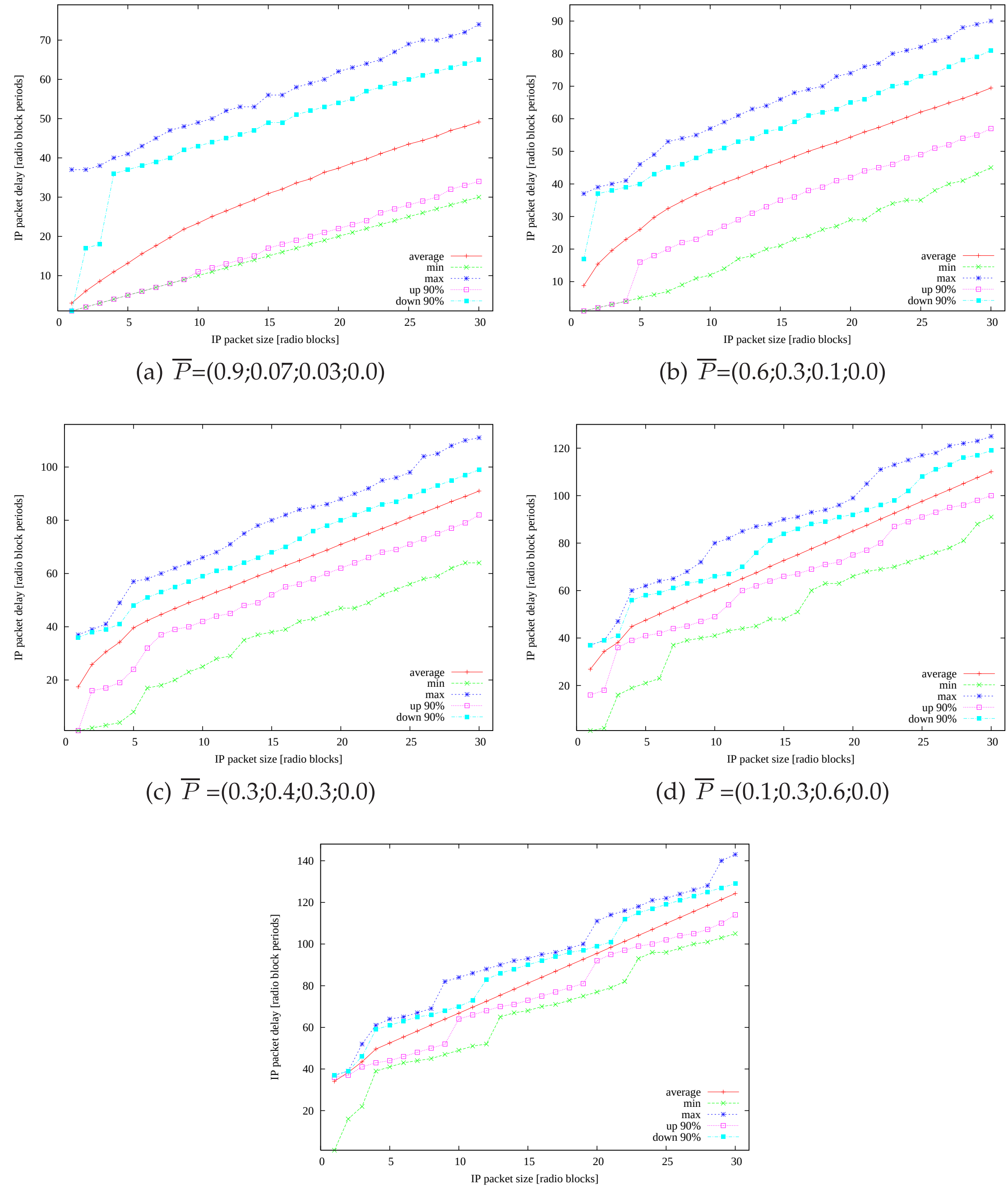
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.3: *Average IP packet delay and min, max, upper 90% anl lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

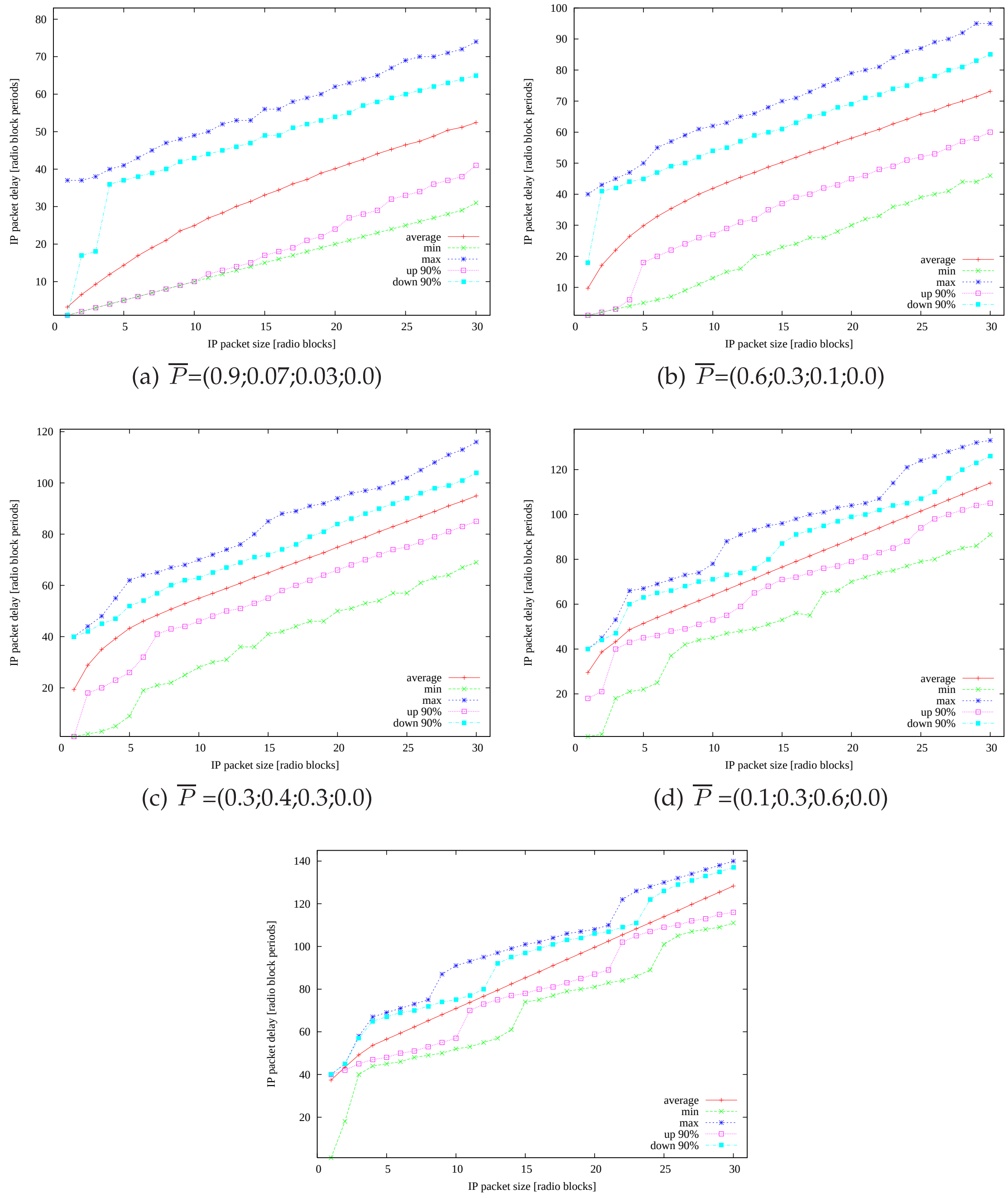
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.4: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

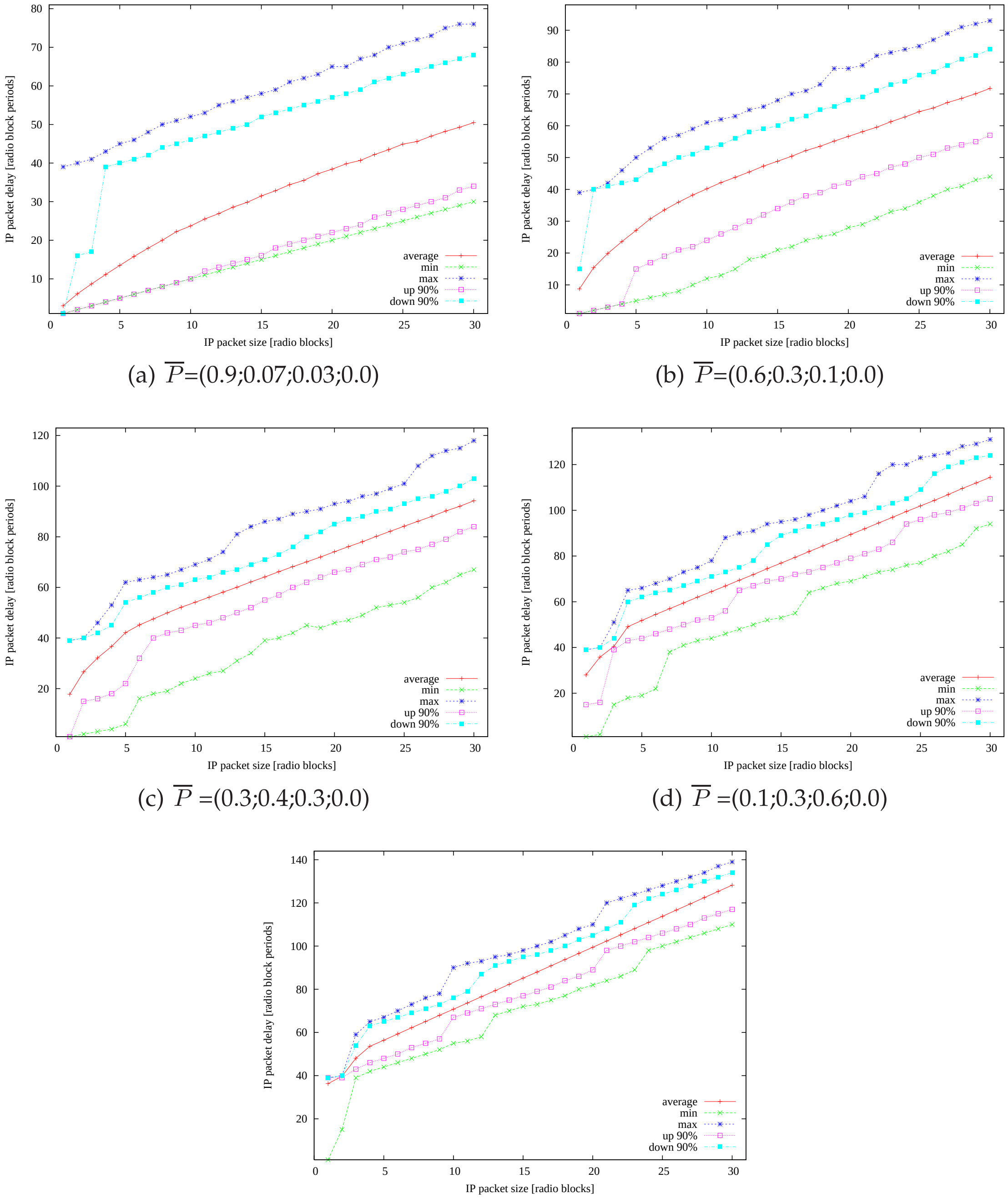
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.5: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

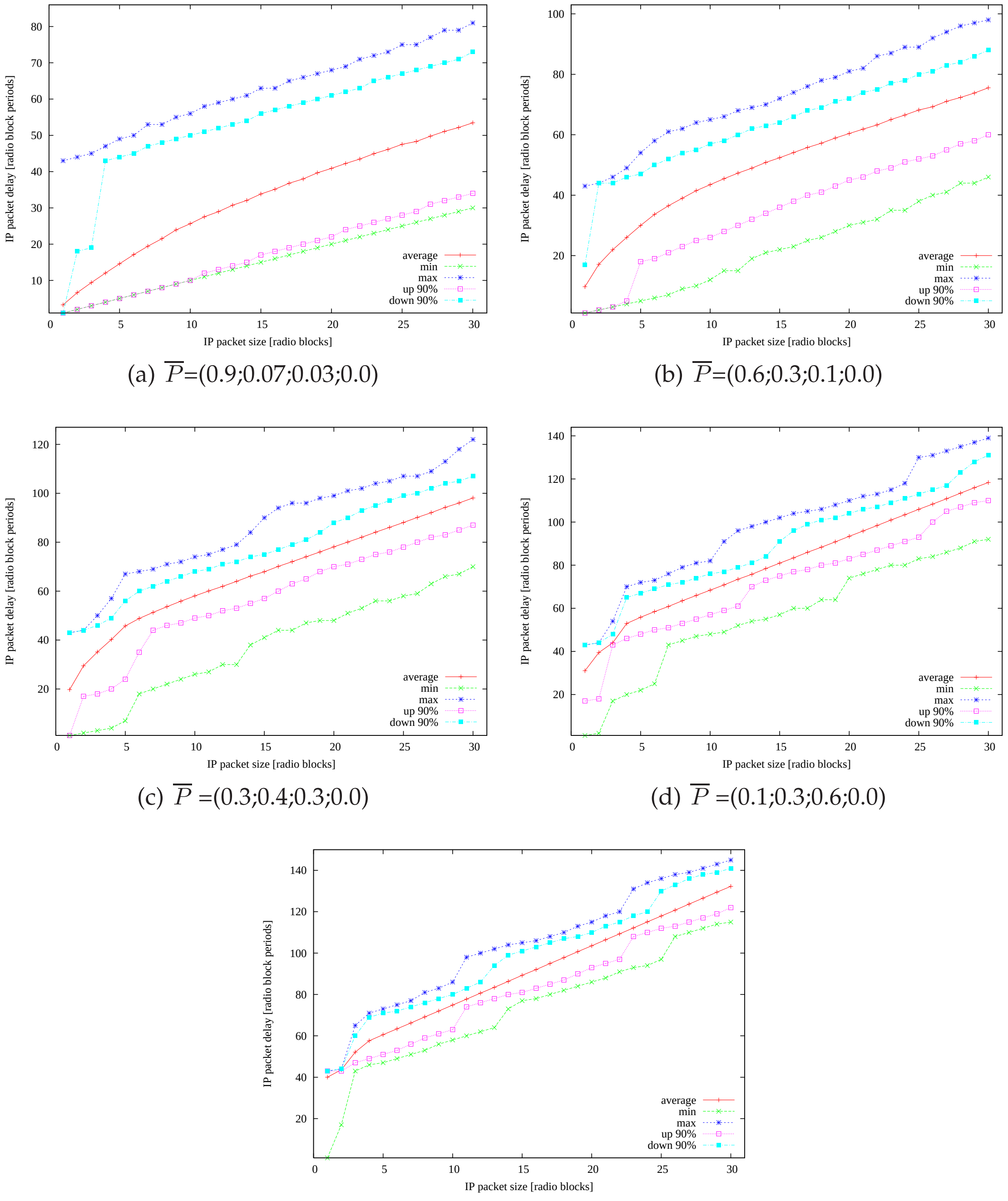
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.6: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

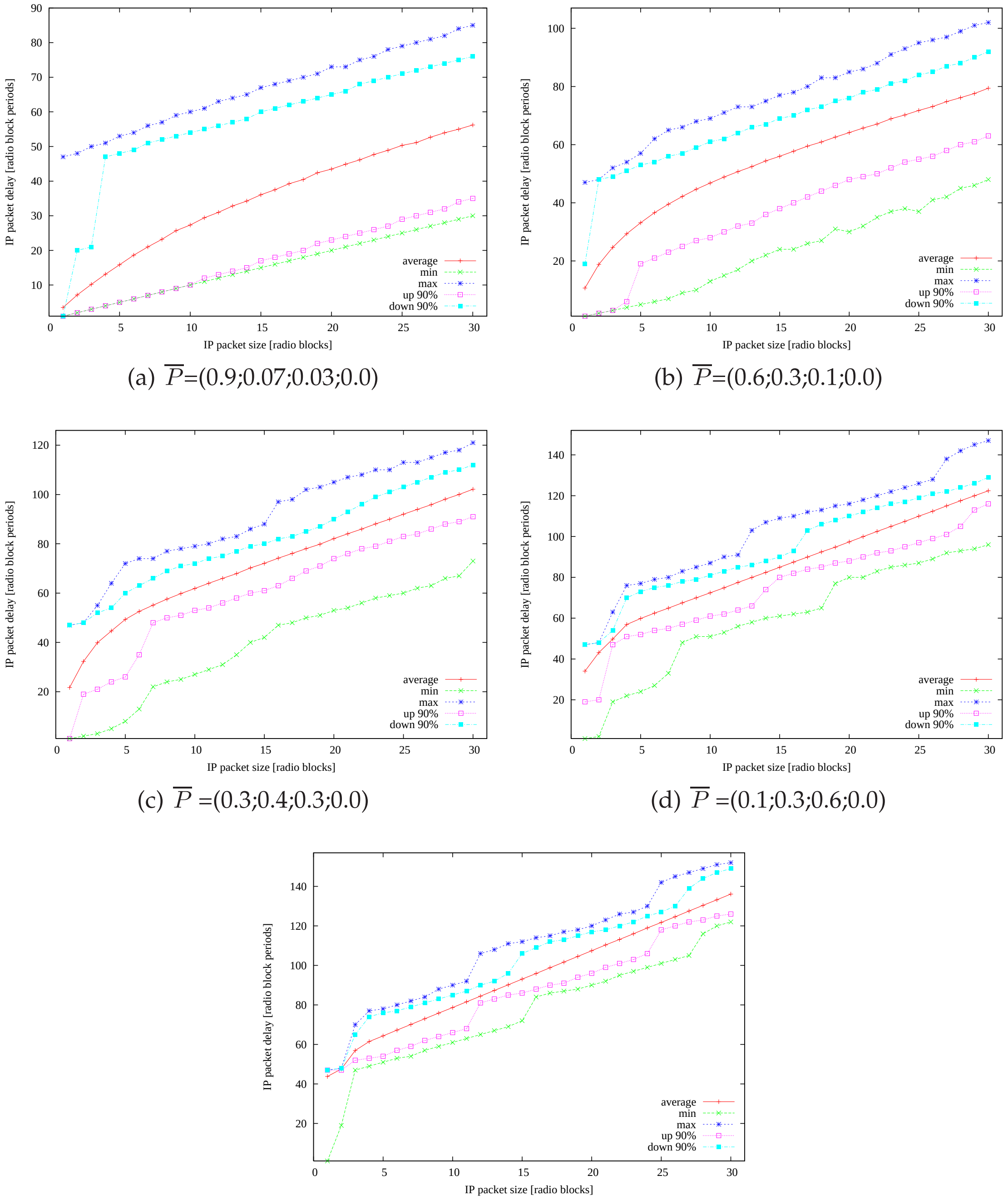
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.7: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

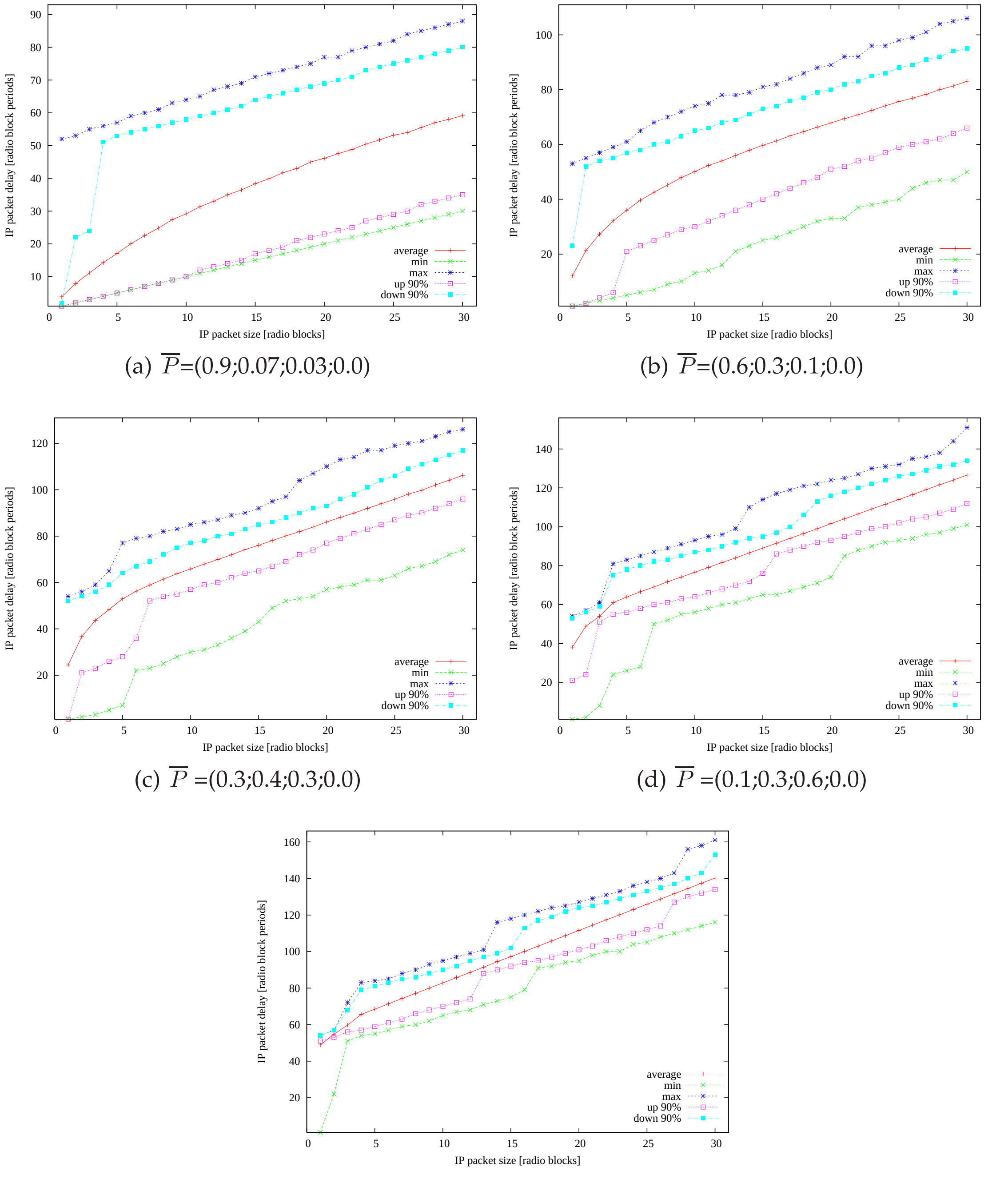
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.8: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

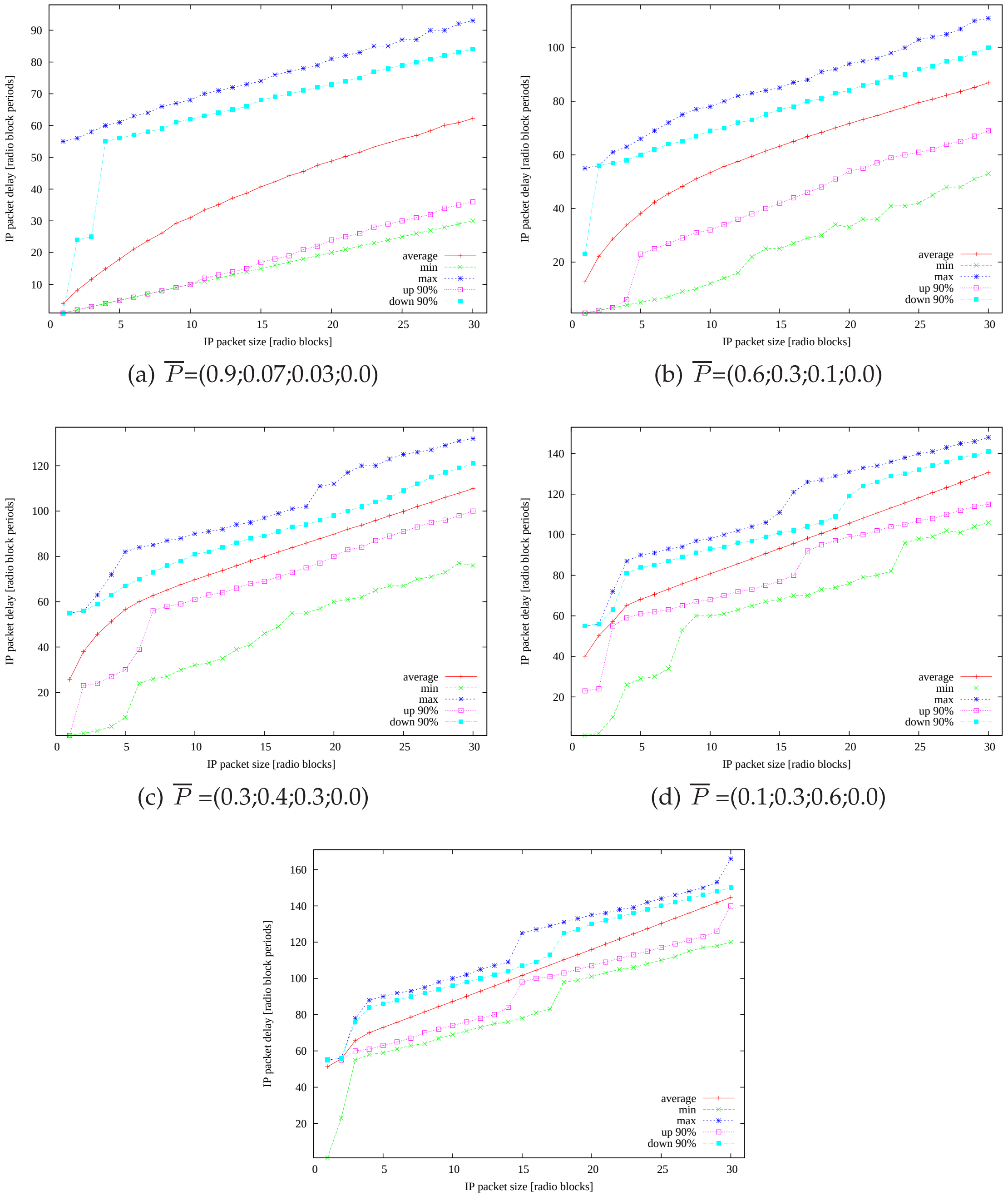
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.9: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

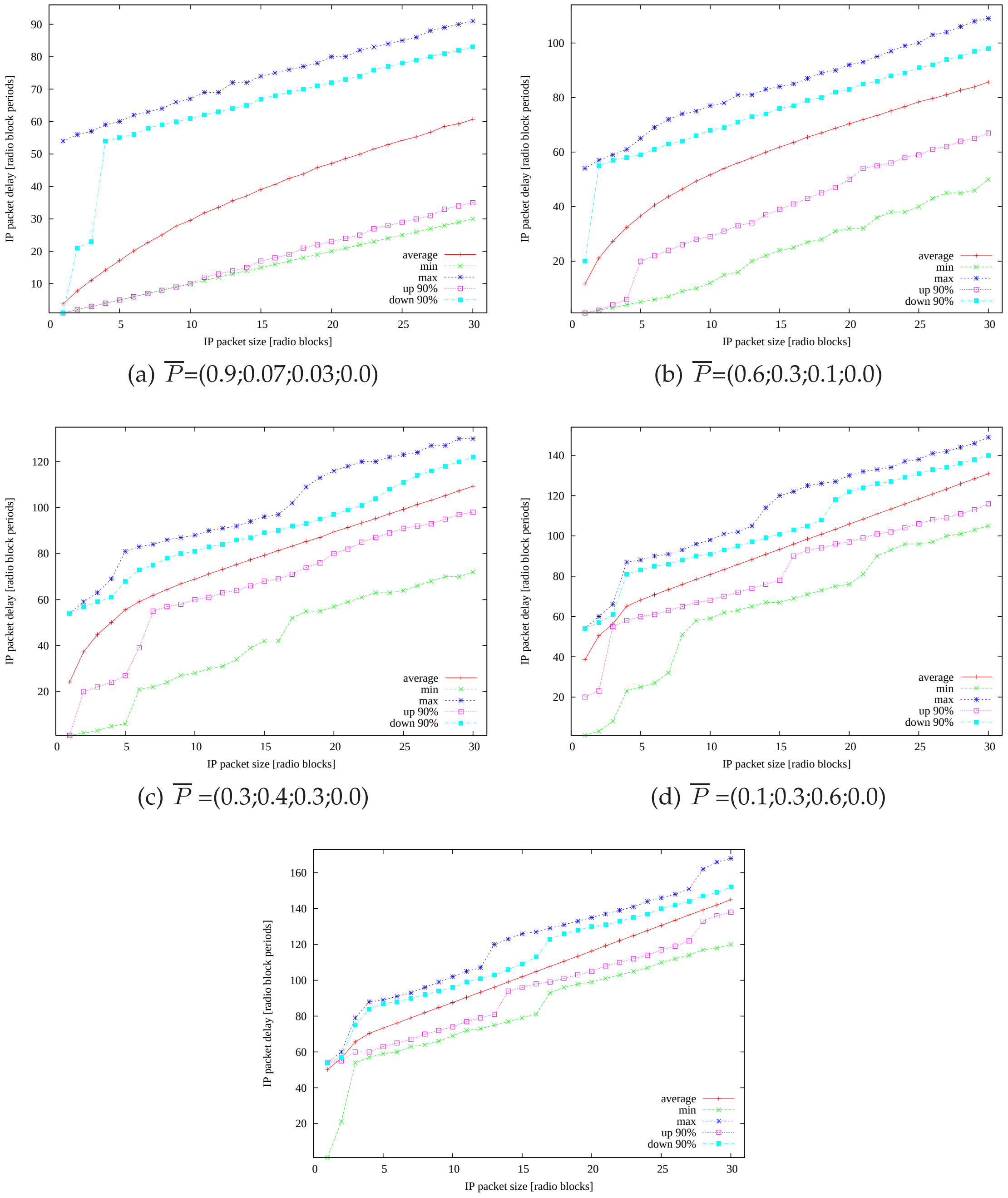
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.10: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

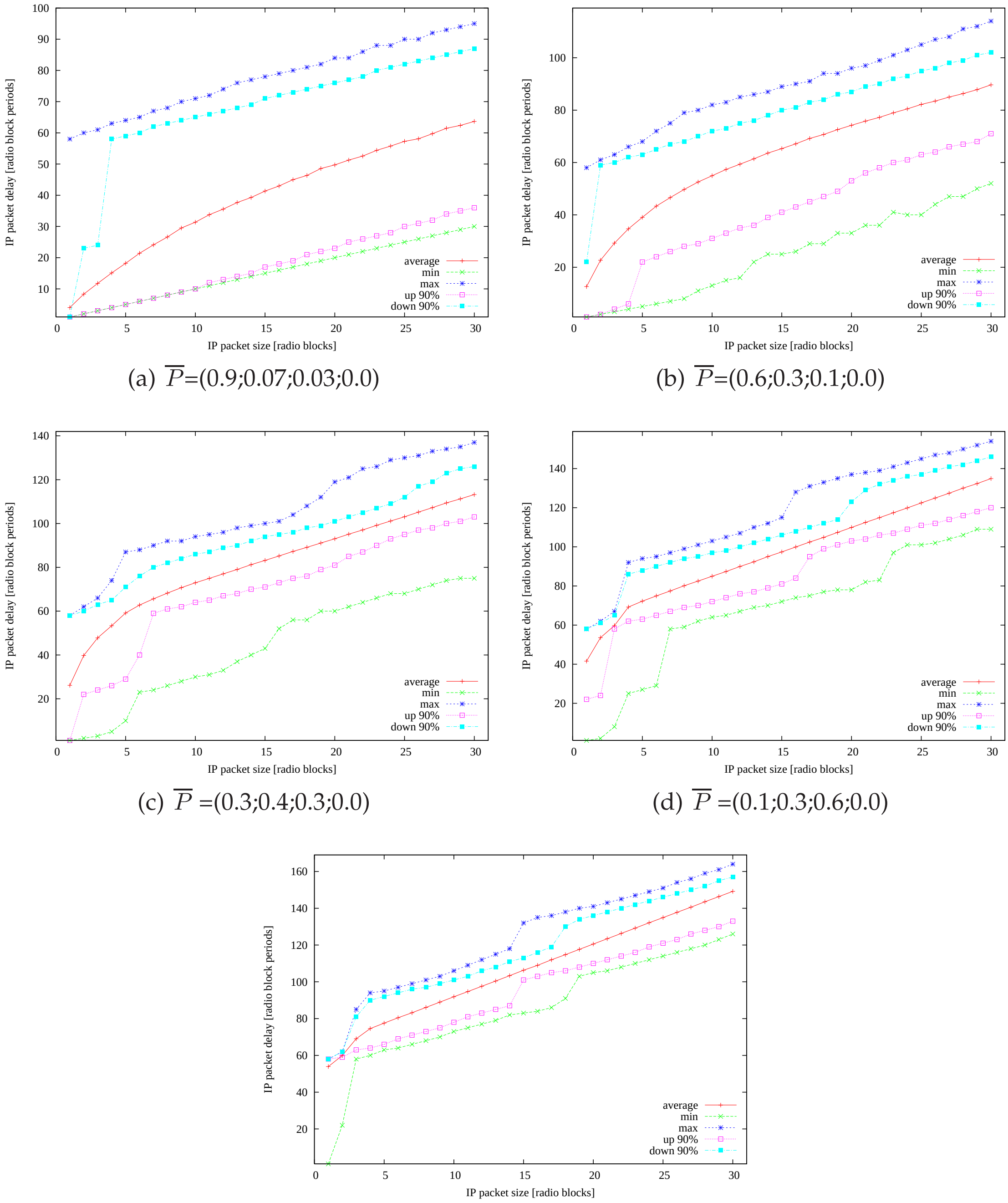
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.11: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

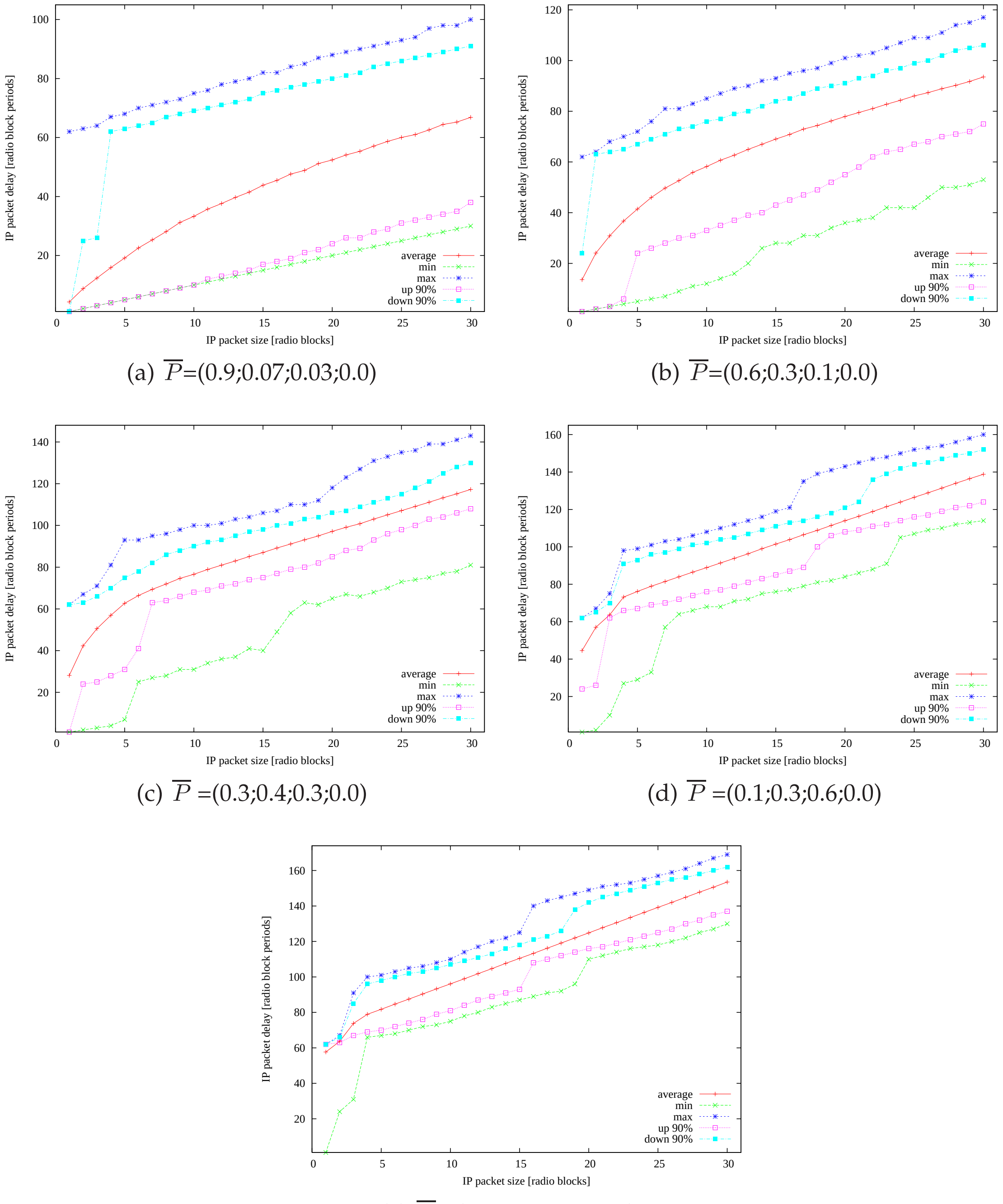
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.12: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

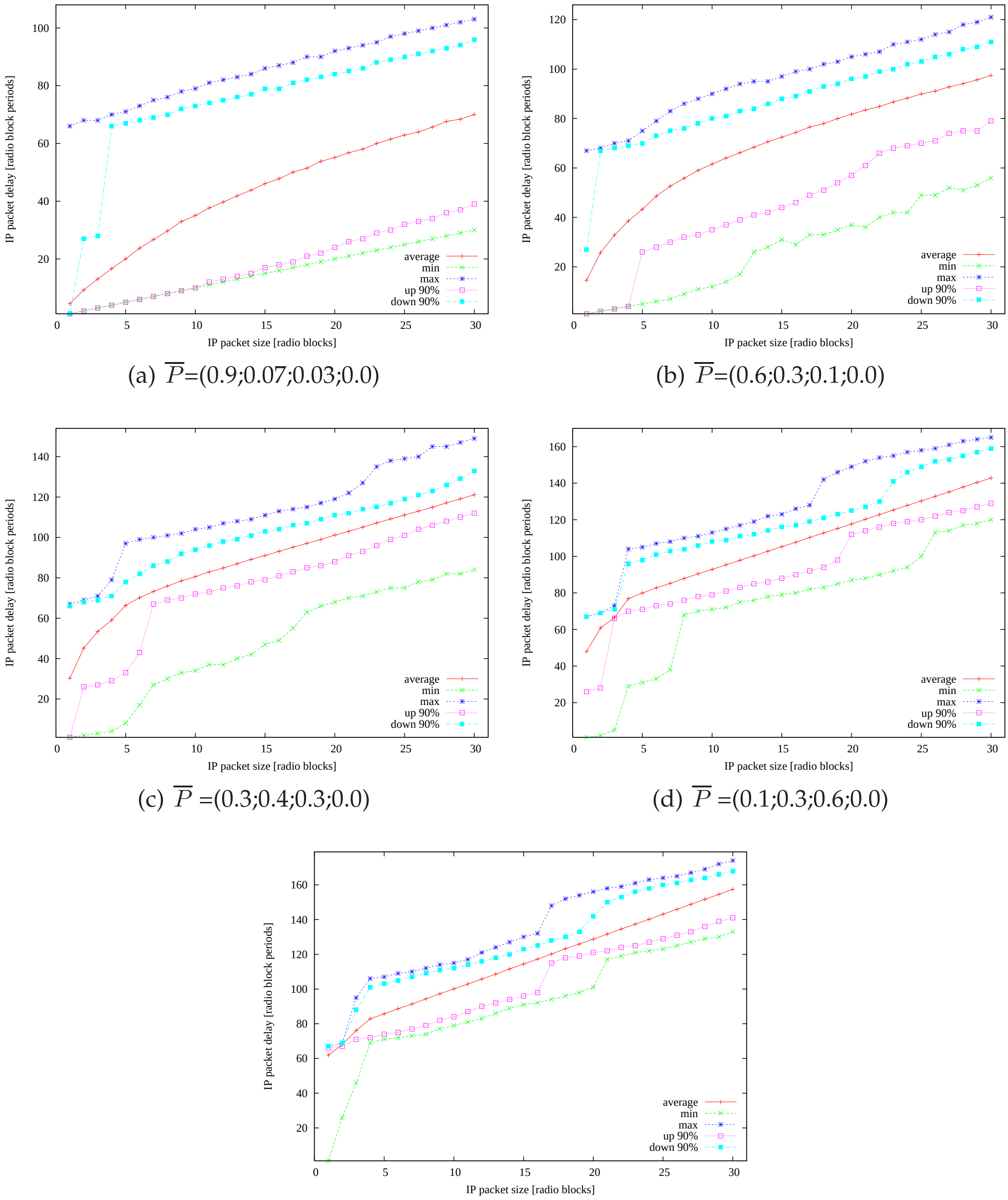
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.13: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

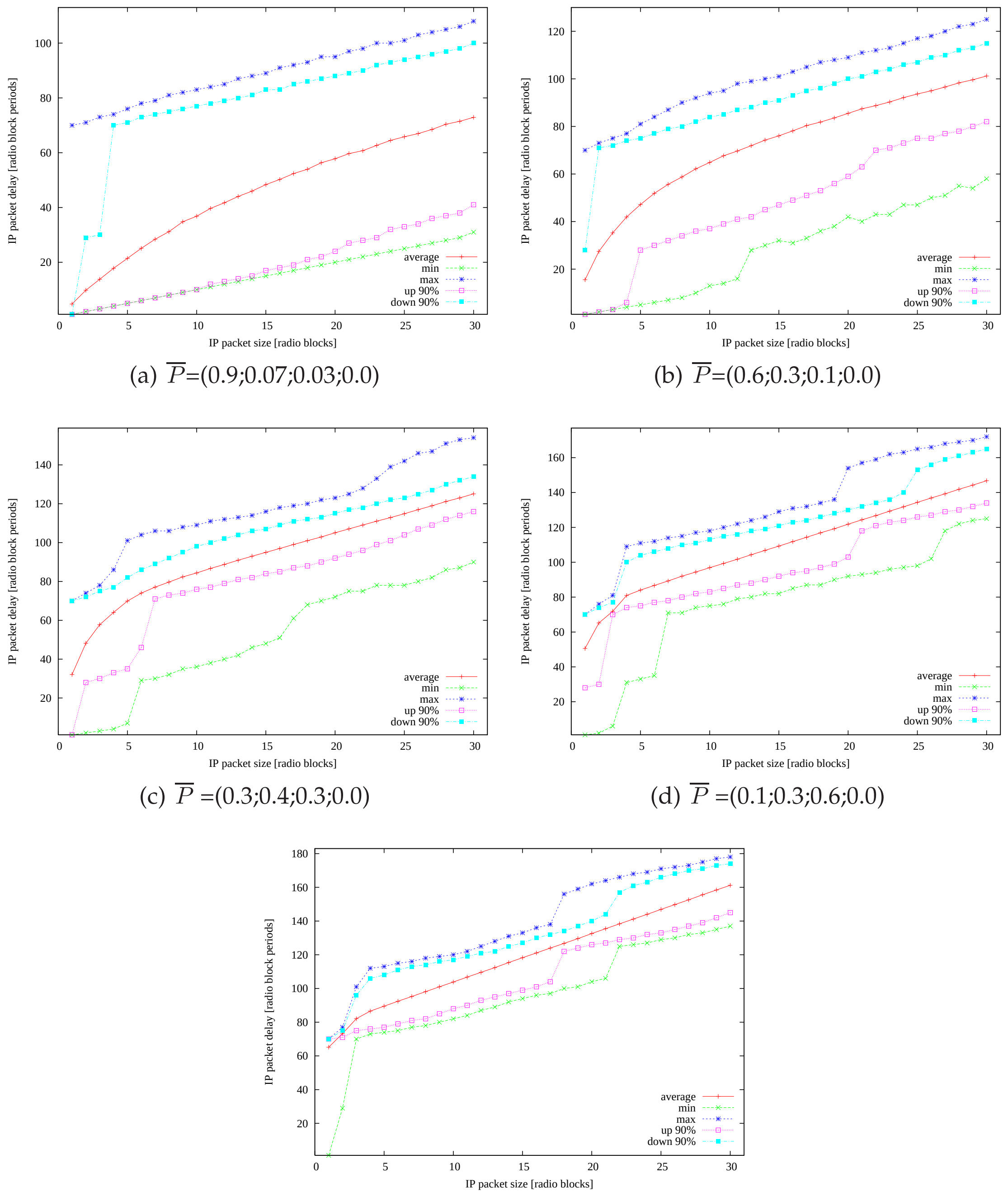
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.14: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure B.15: *Average IP packet delay and min, max, upper 90% and lover 90% bounds - simulation results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

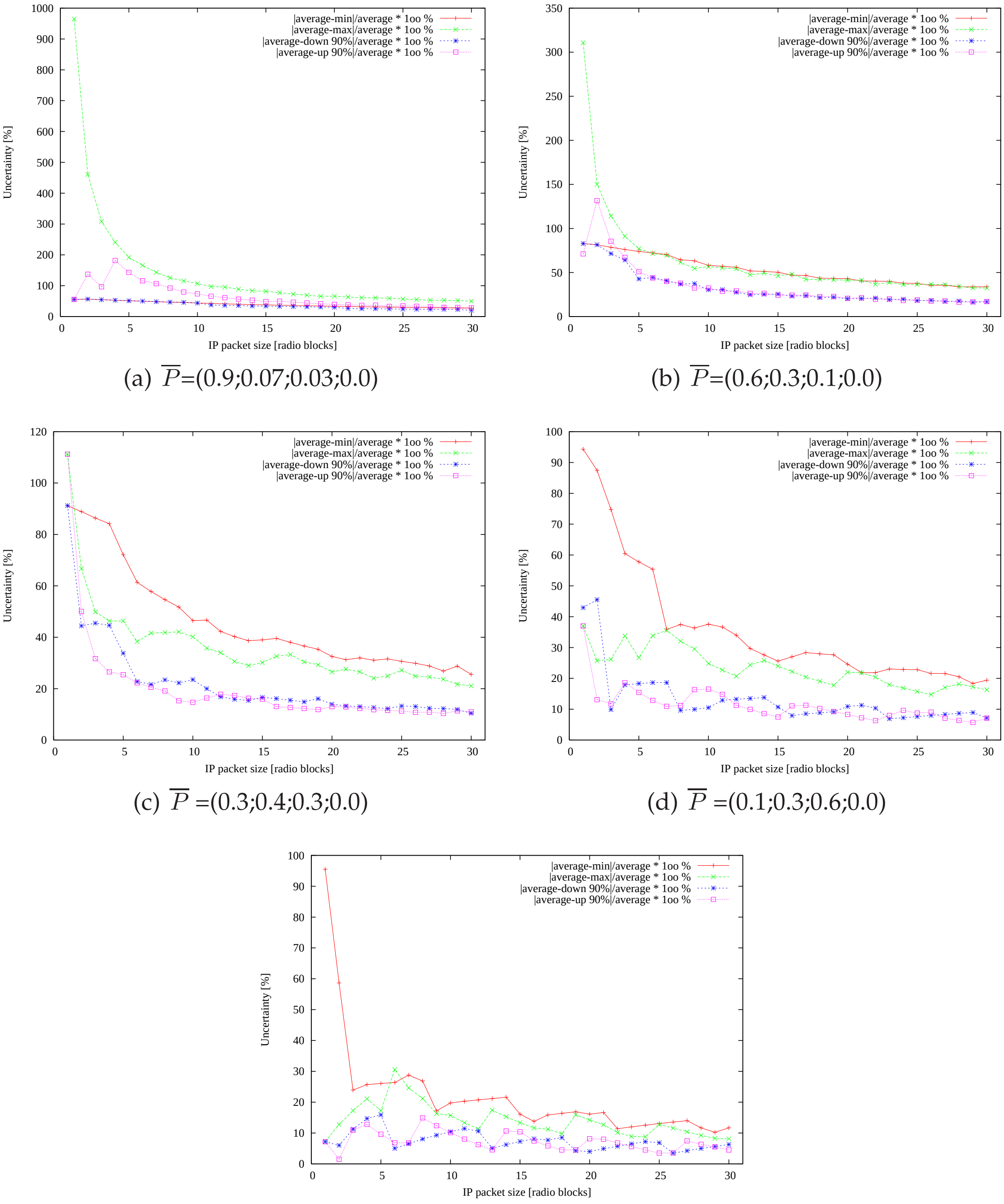
APPENDIX C

Uncertainty of an average IP packet delay characteristic

This appendix presents plots of the uncertainty as a function of IP packet size. The uncertainty is calculated as the percentage difference between the average value and maximum, minimum, upper and lower 90% bounds as shown in equation C.1.

$$\begin{aligned} Uncertainty_{maximum}(k) &= \tfrac{|average(k)-maximum(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{minimum}(k) &= \tfrac{|average(k)-minimum(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{upper90\%}(k) &= \tfrac{|average(k)-upper90\%(k)|}{average(k)} \cdot 100\% \\ Uncertainty_{lower90\%}(k) &= \tfrac{|average(k)-lower90\%(k)|}{average(k)} \cdot 100\% \end{aligned} \tag{C.1}$$

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

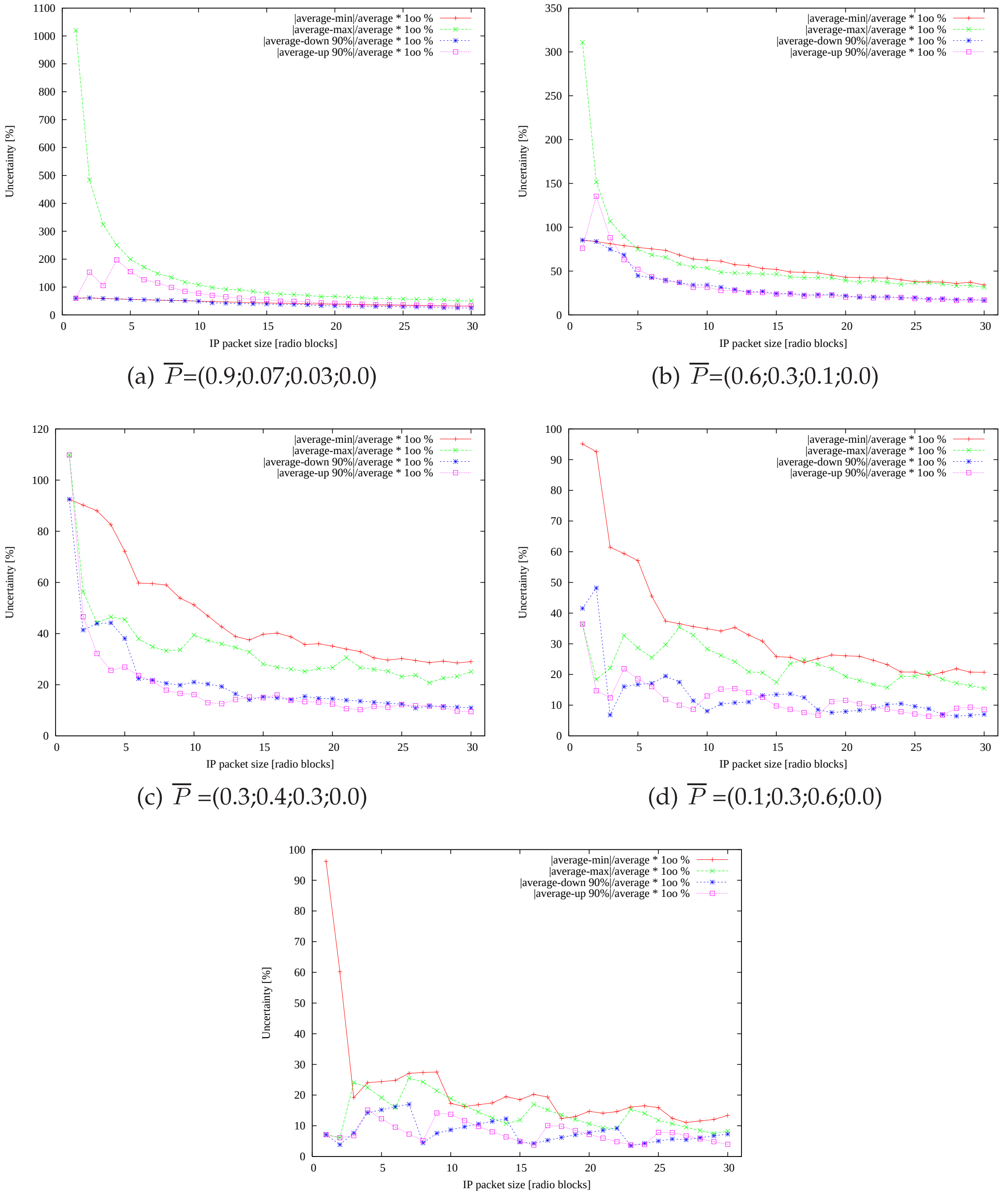
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.1: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

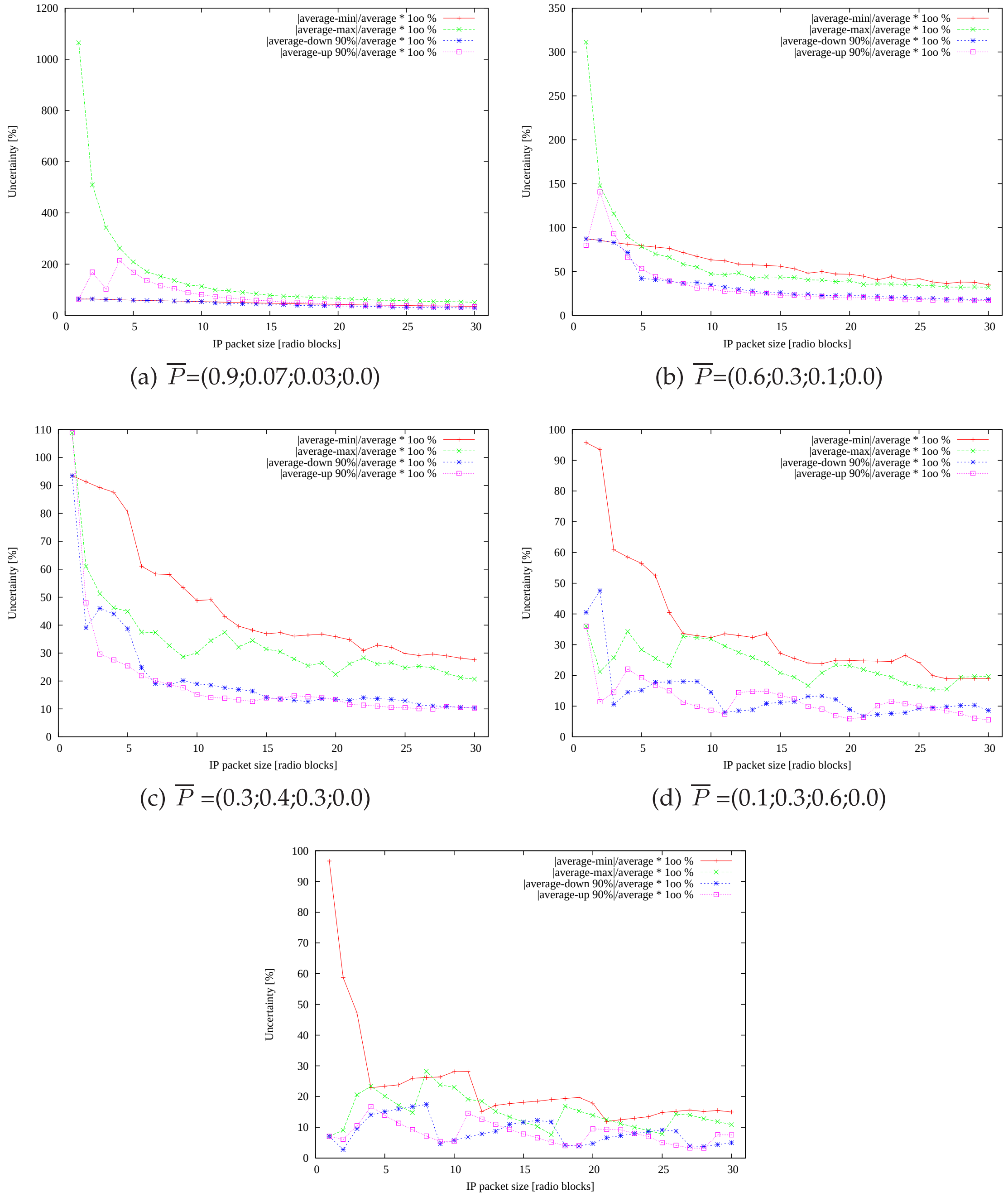
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.2: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

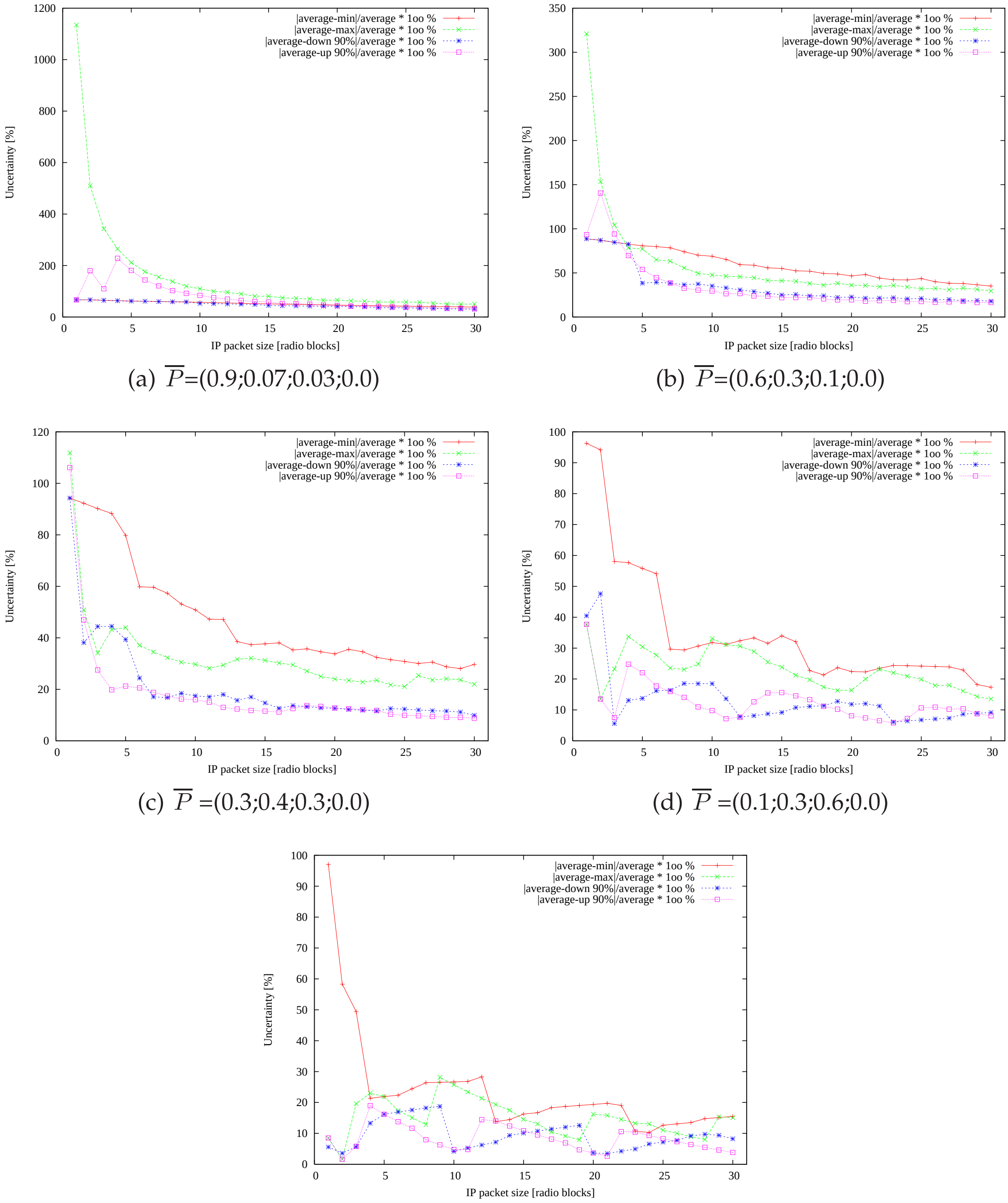
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.3: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

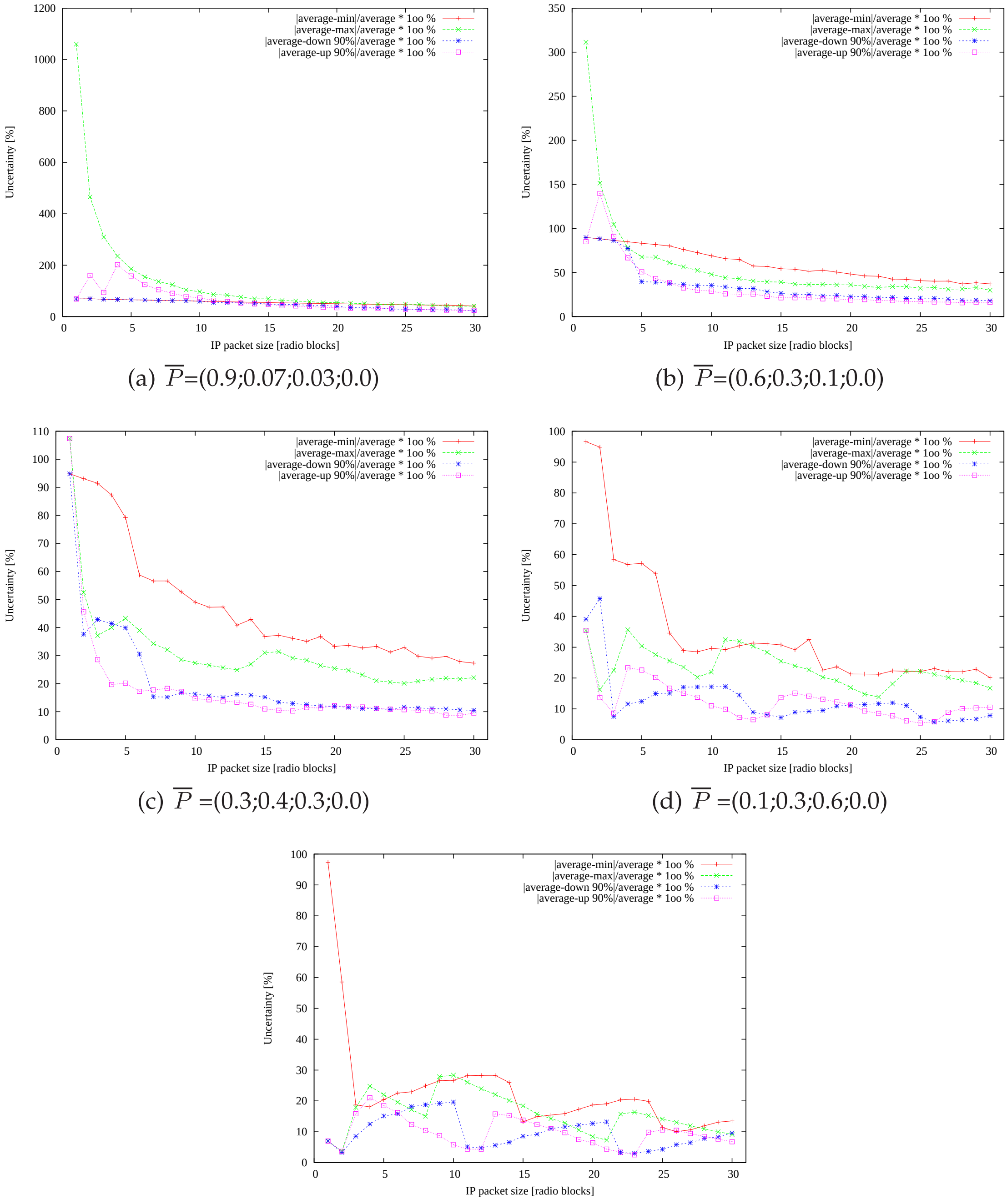
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.4: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

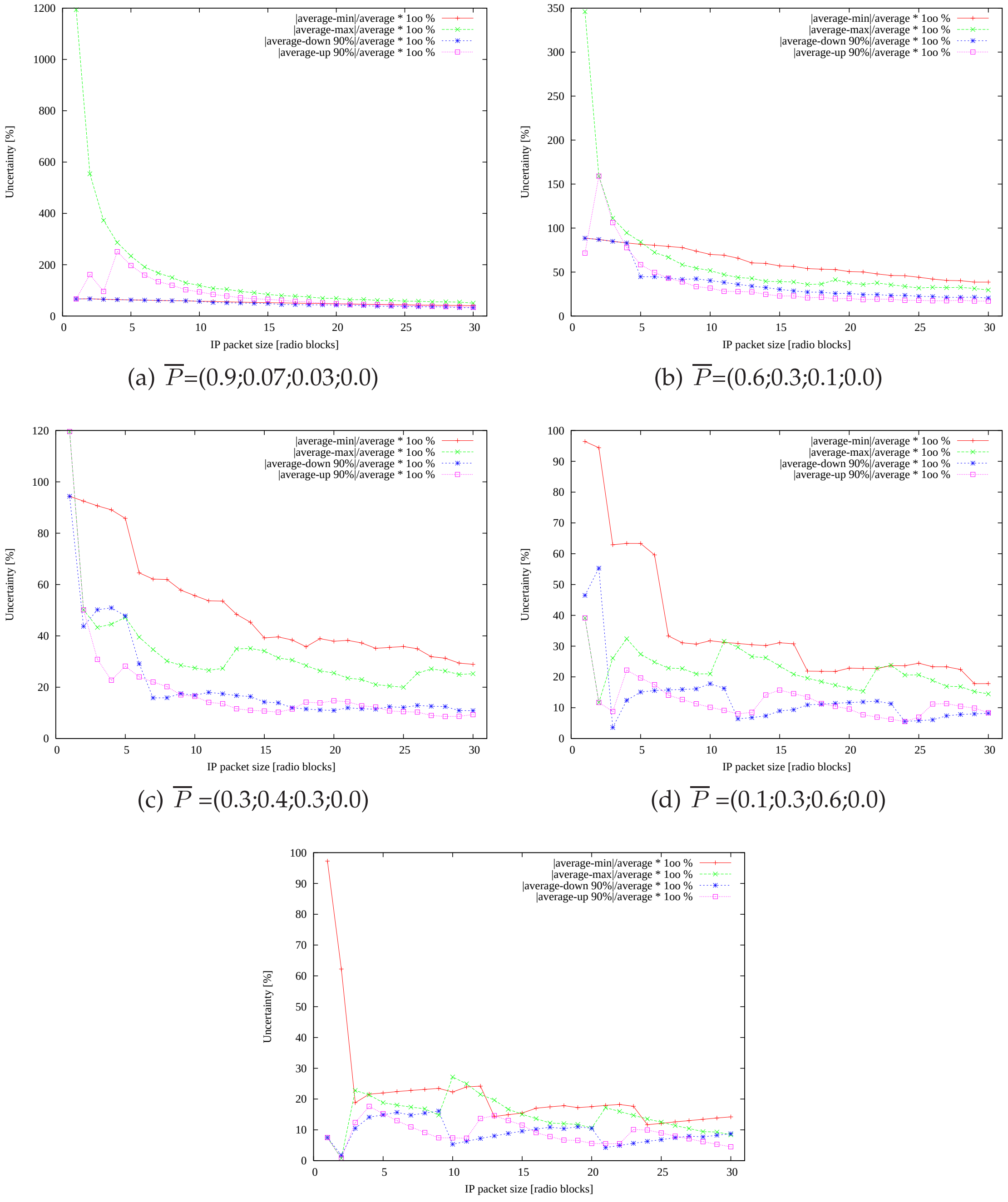
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.5: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

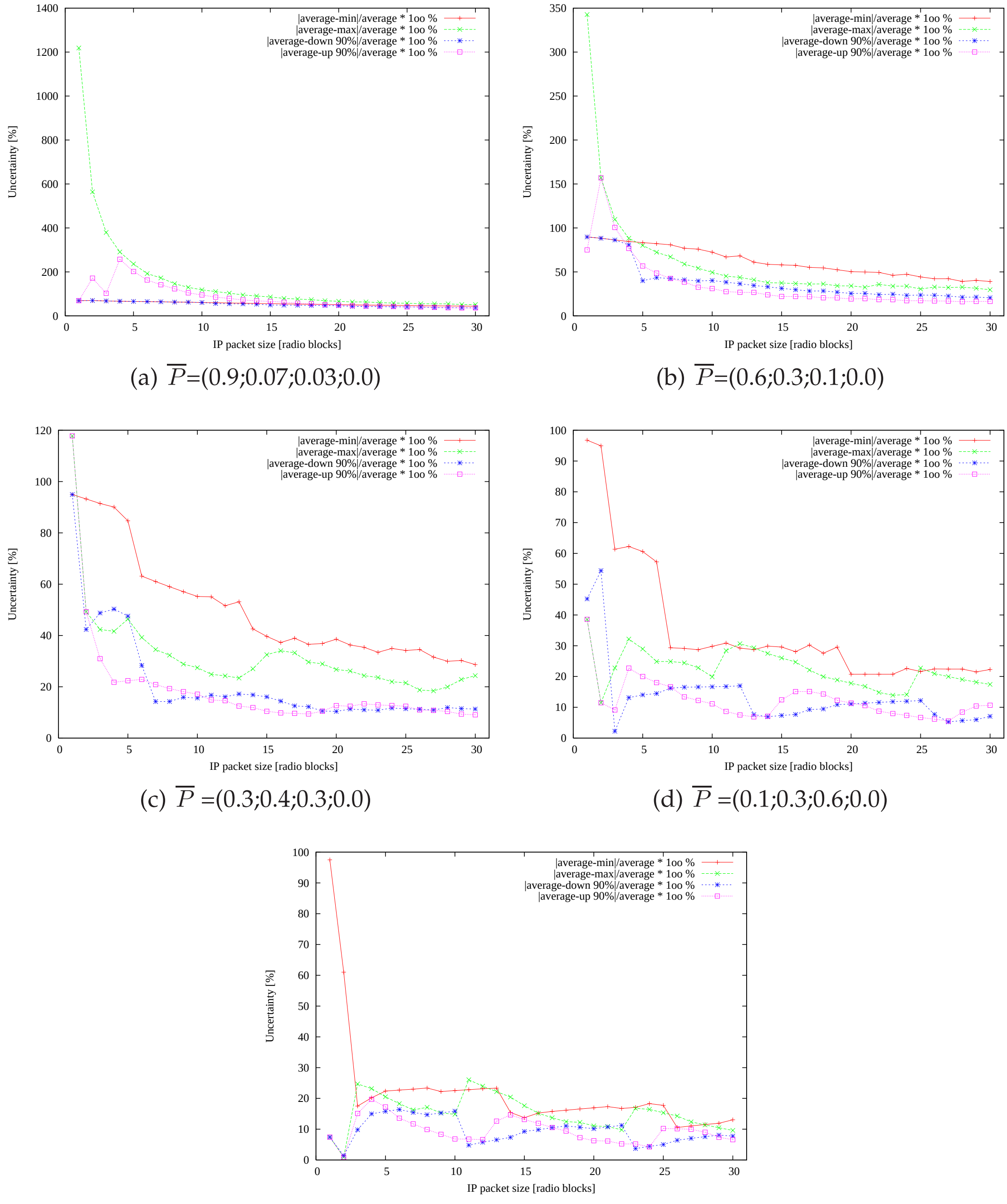
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.6: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

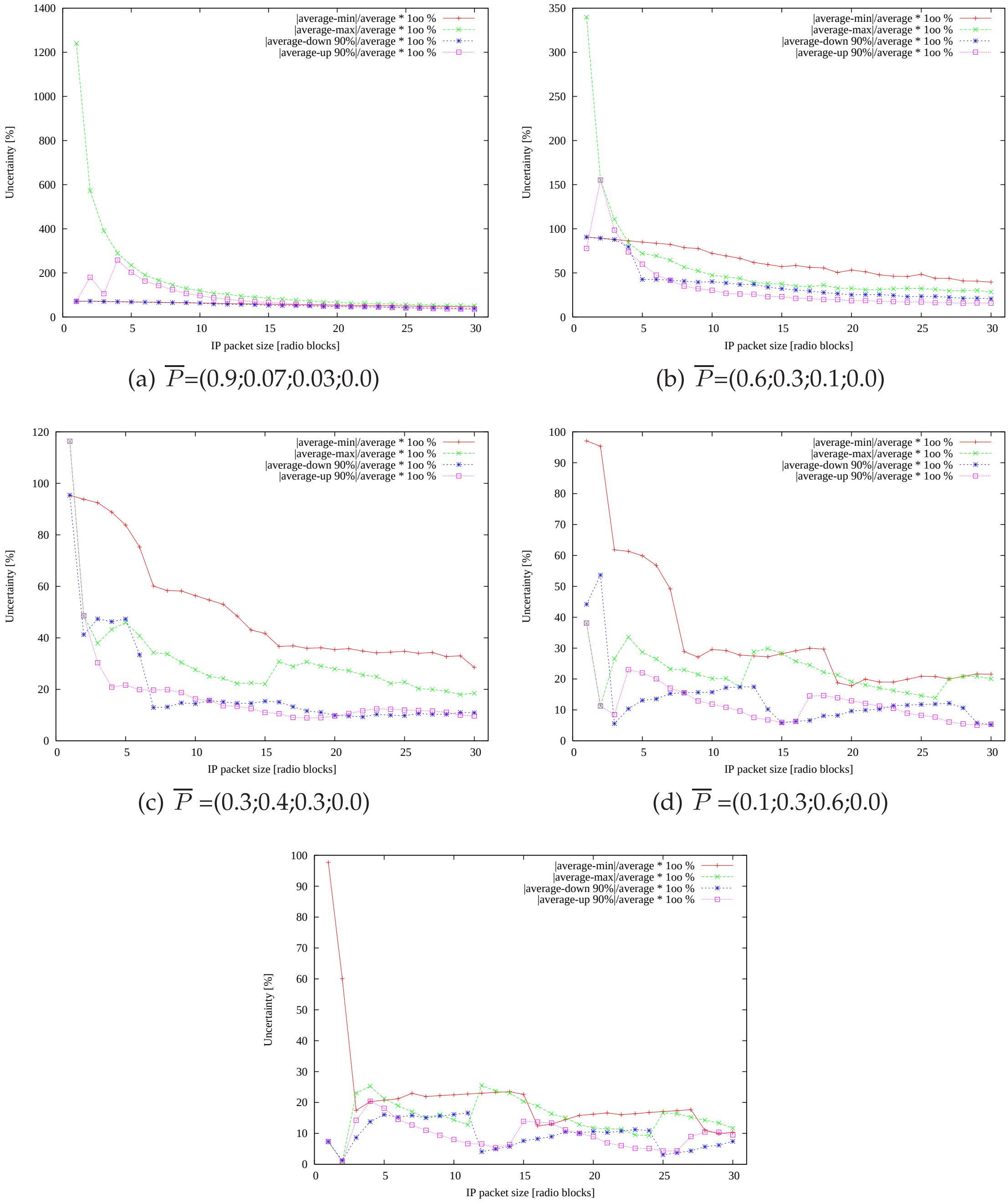
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.7: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

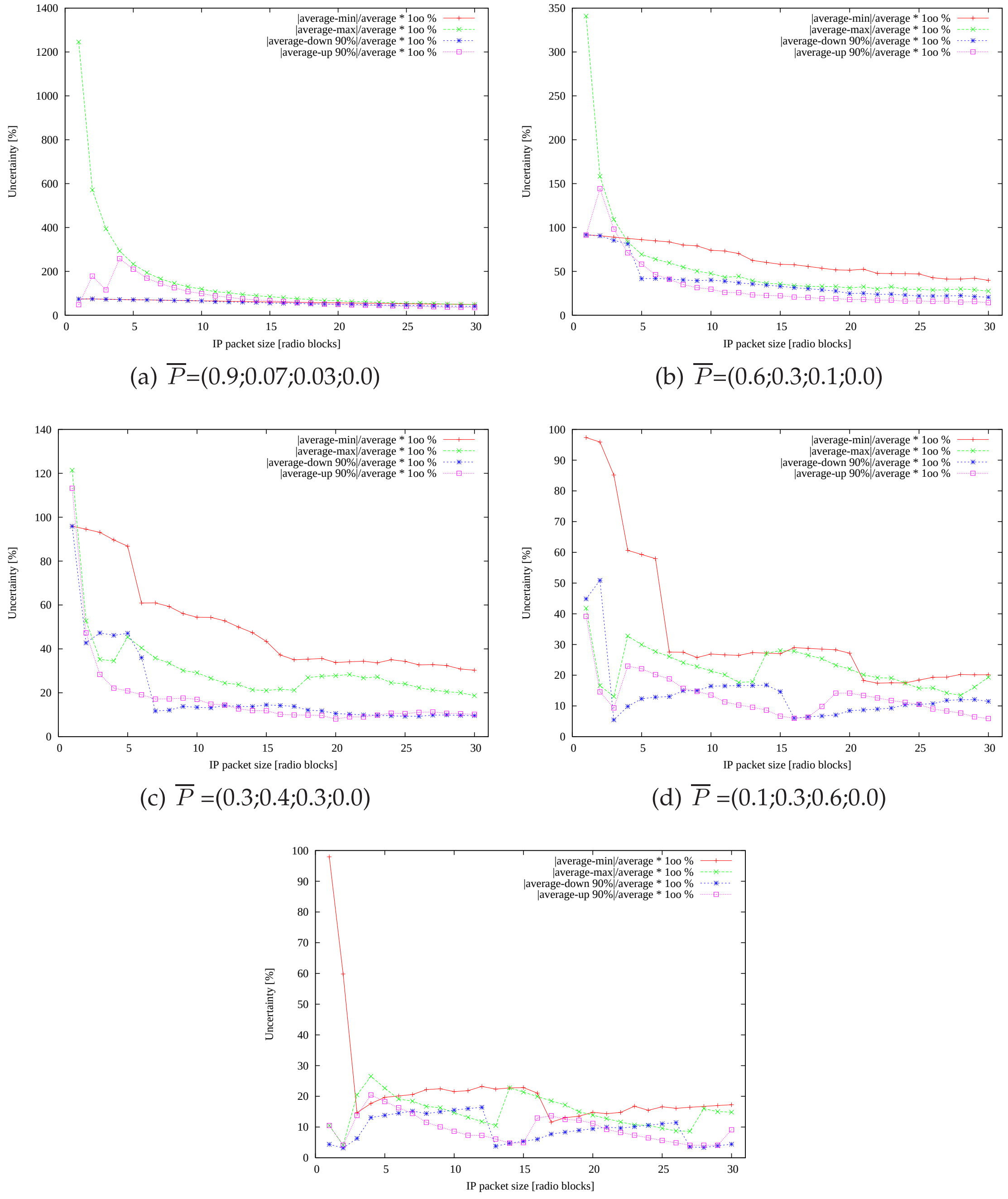
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.8: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

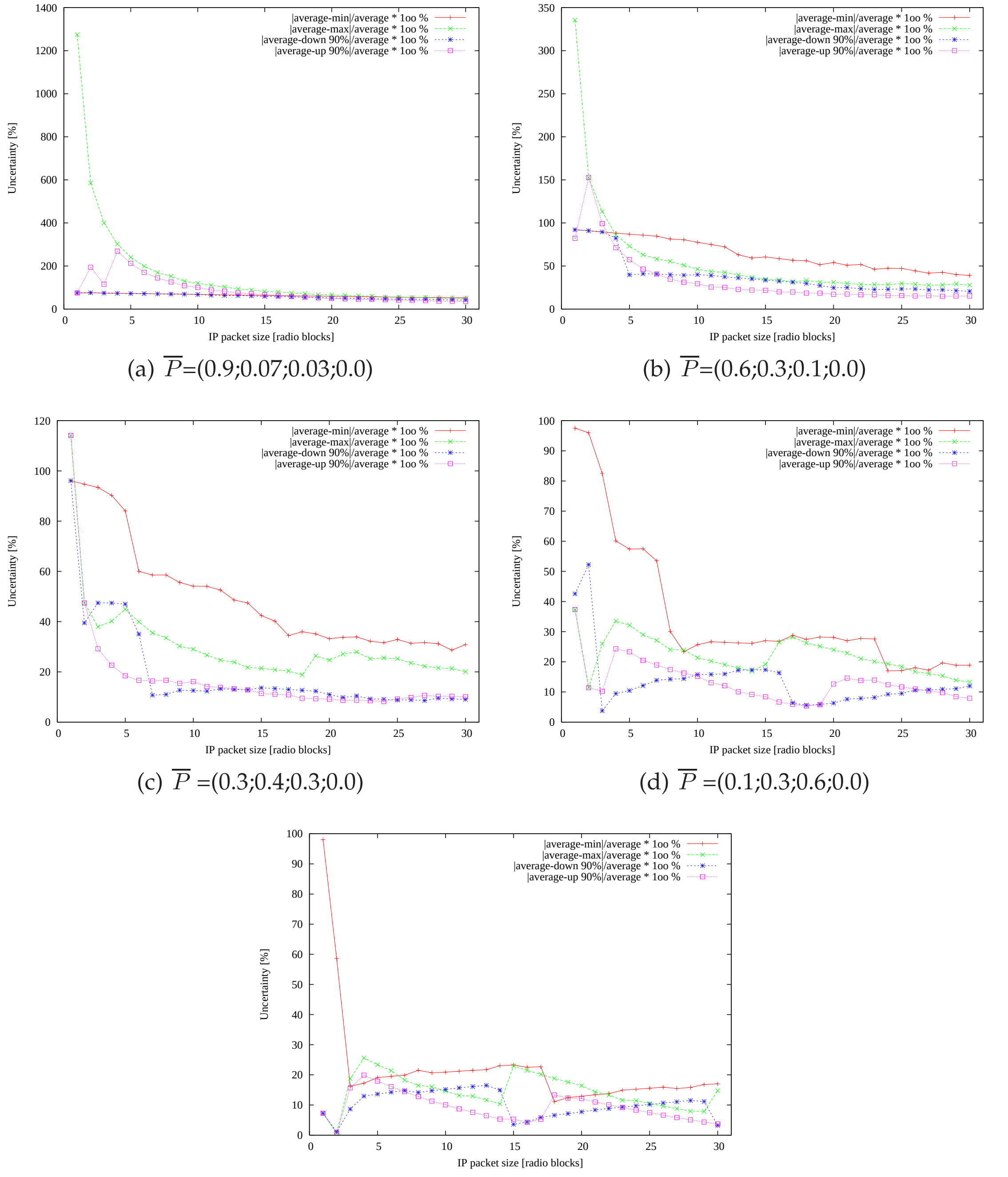
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.9: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

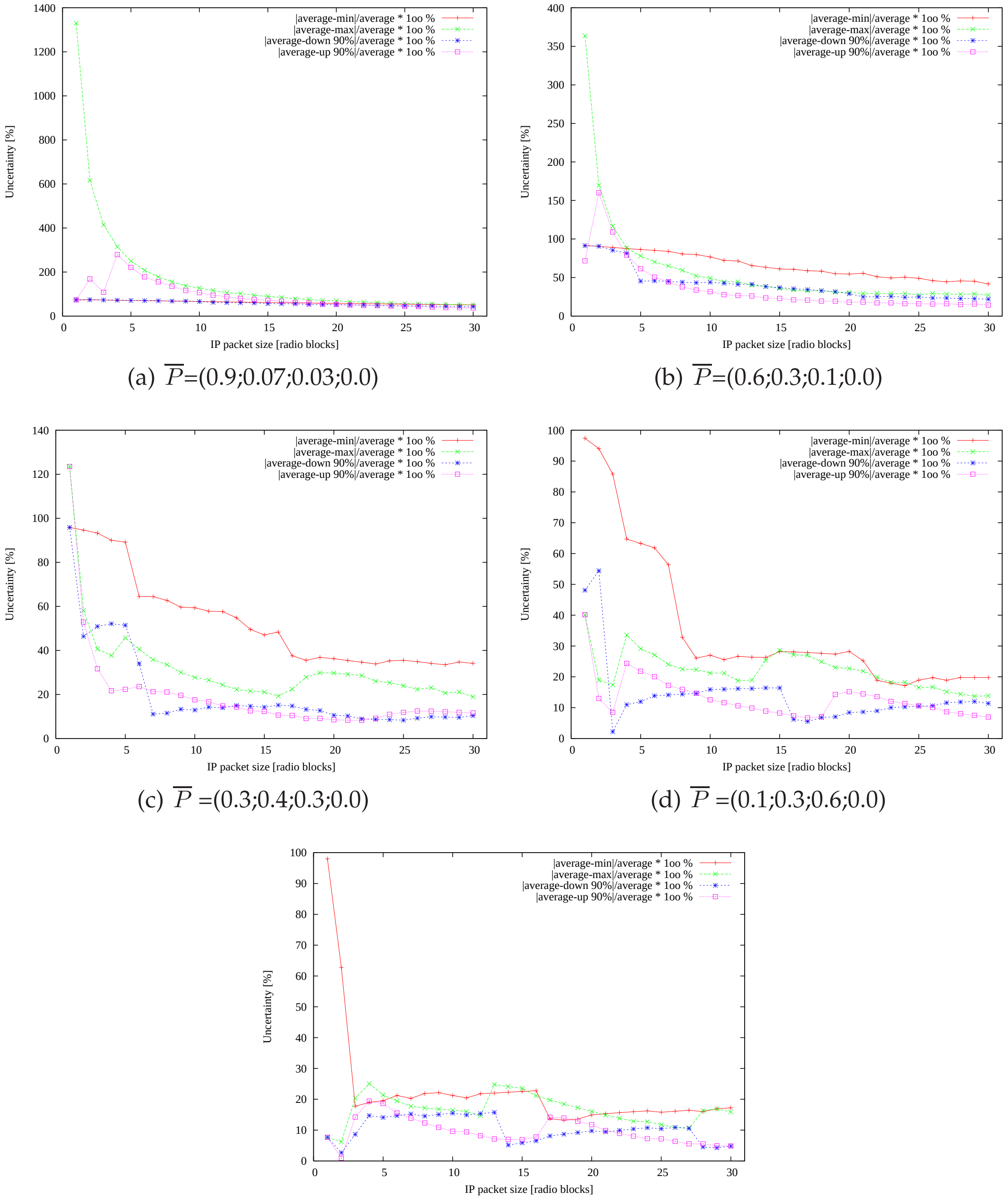
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.10: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

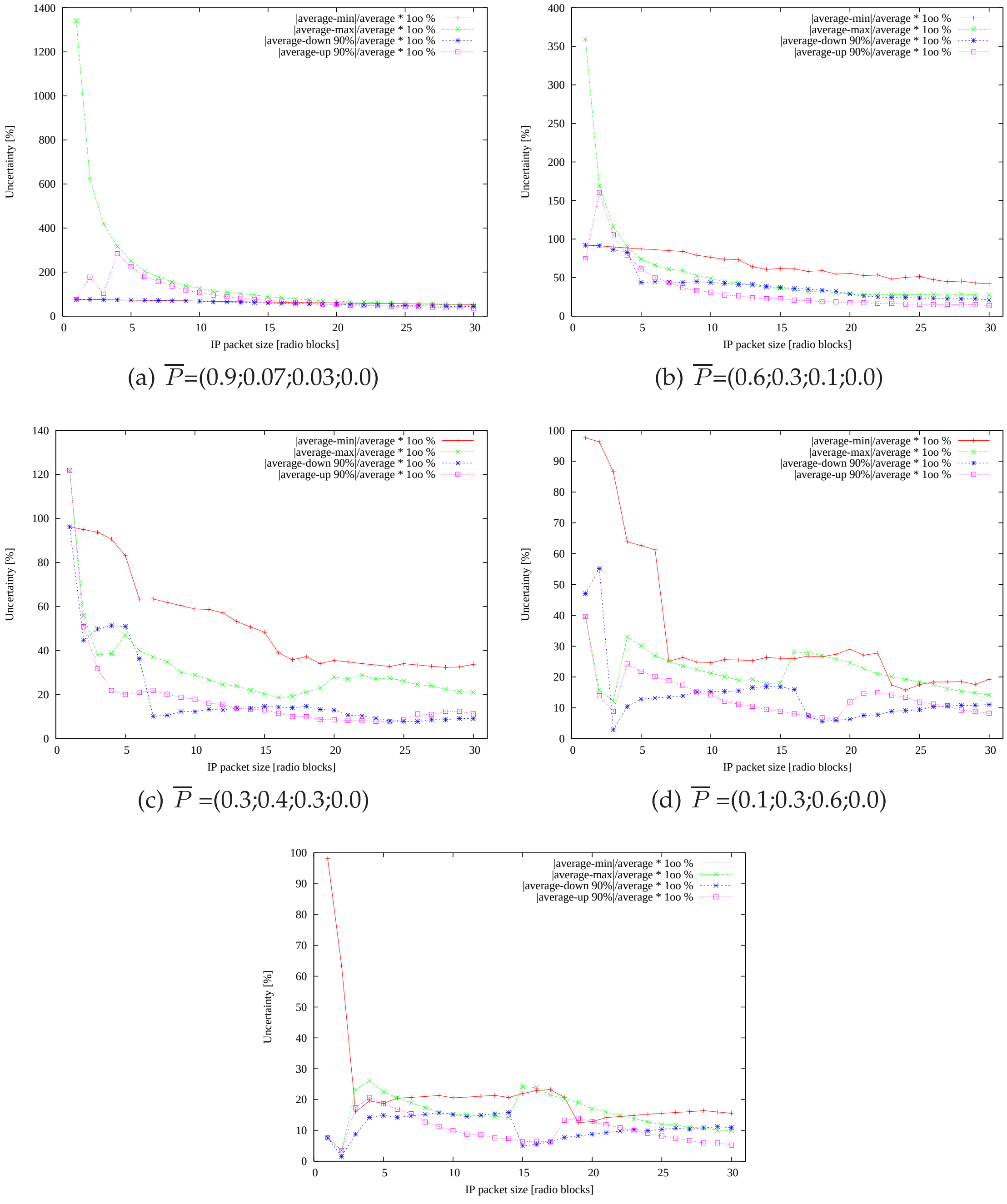
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.11: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

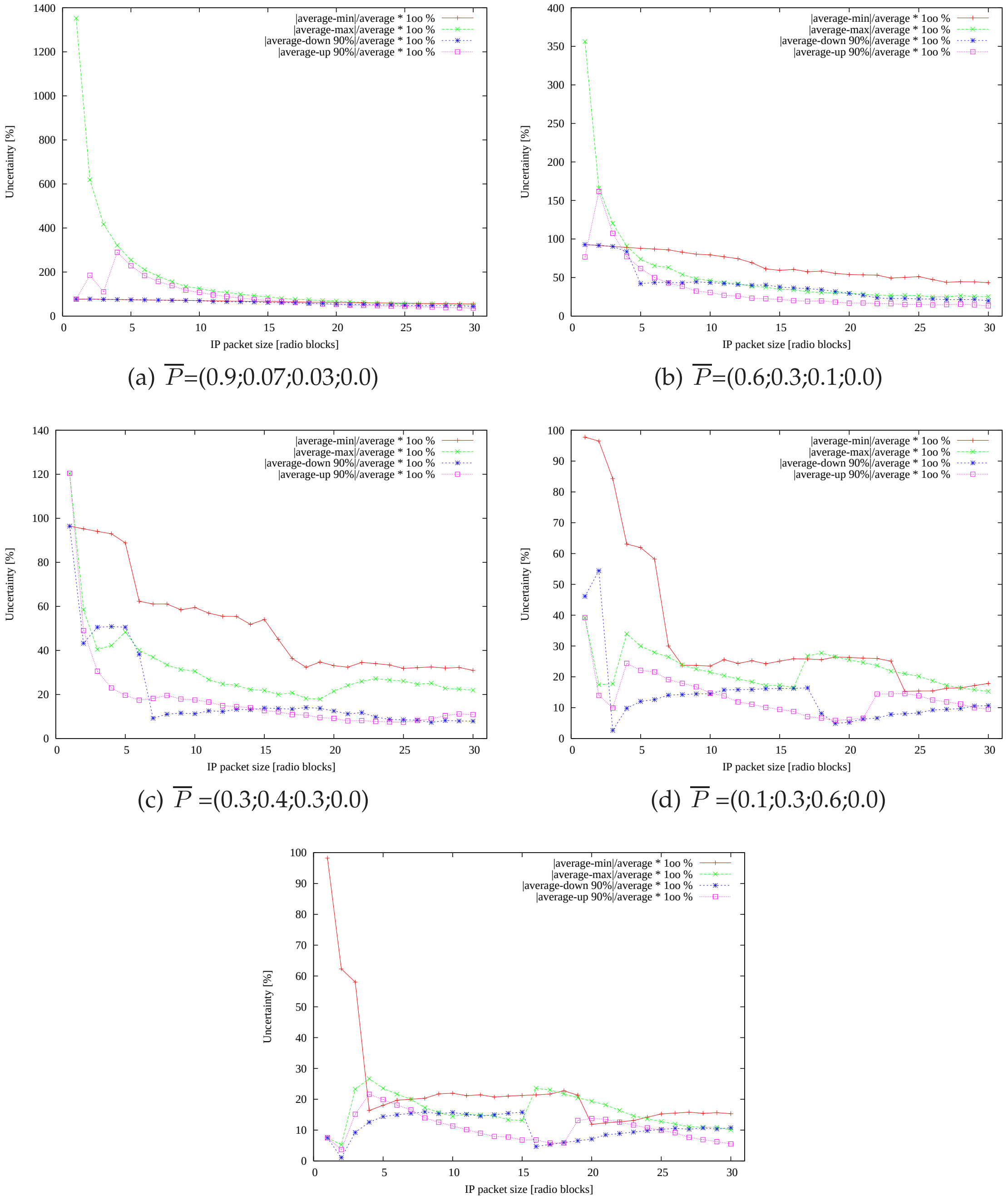
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.12: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

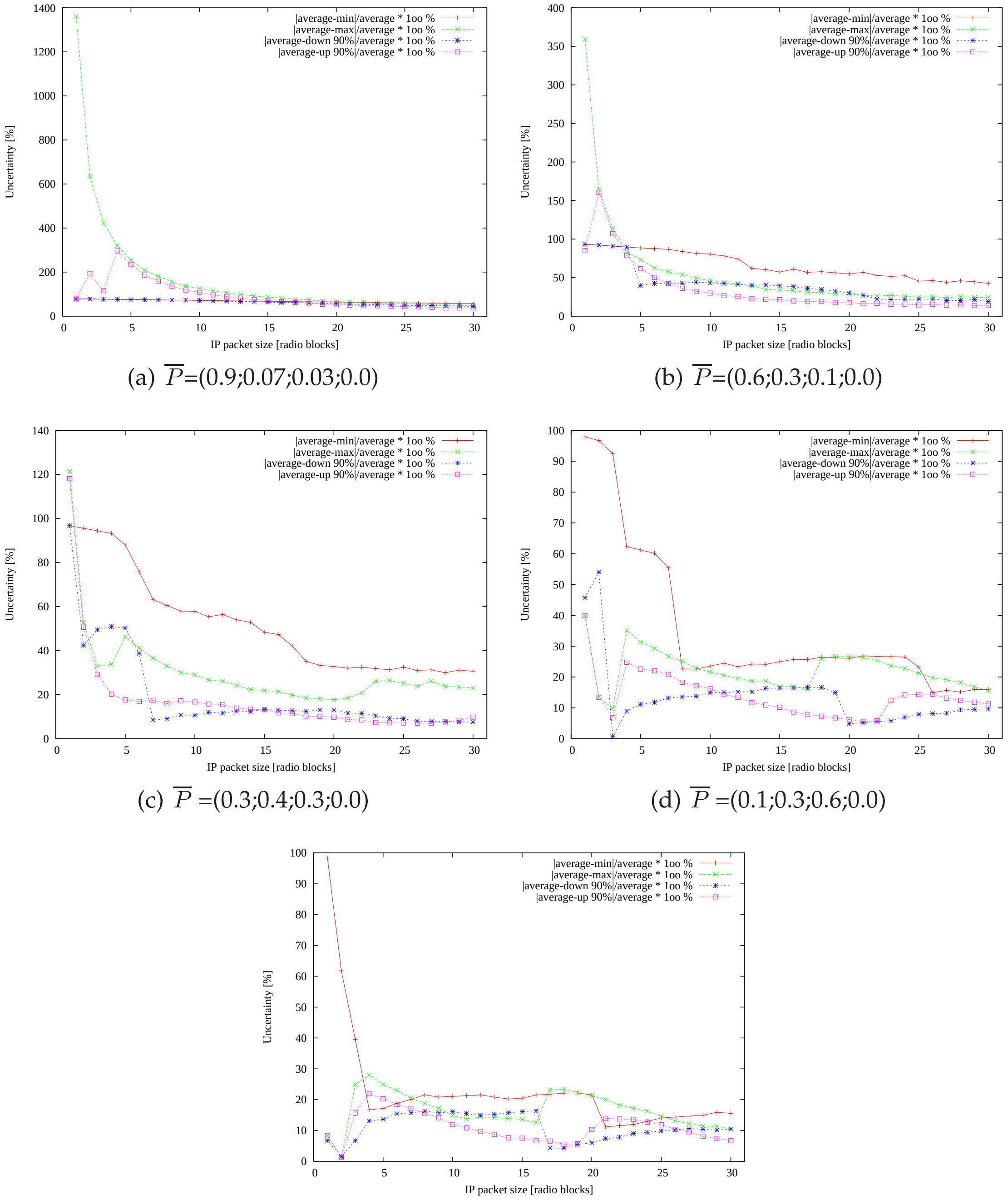
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.13: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

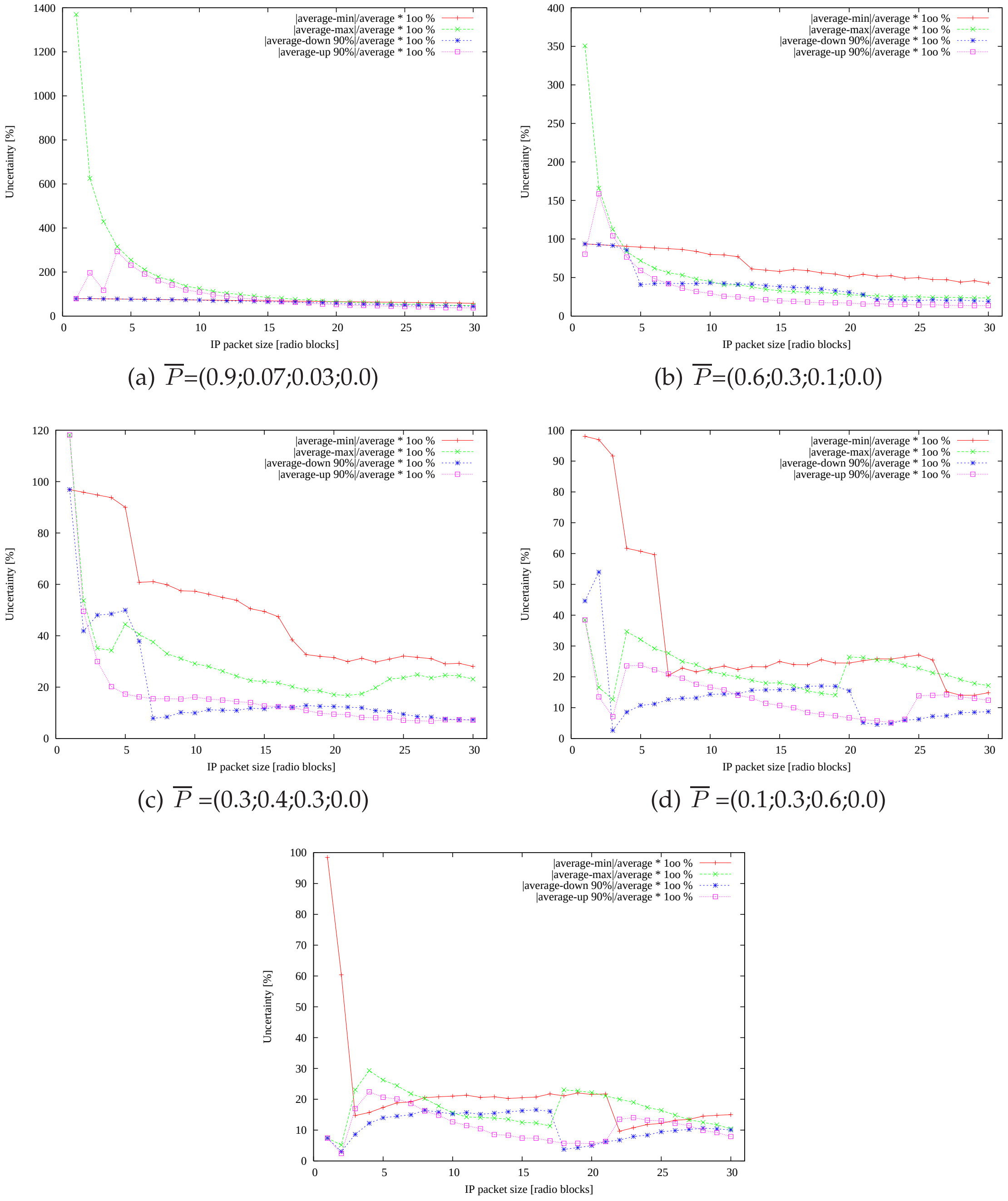
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure C.14: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

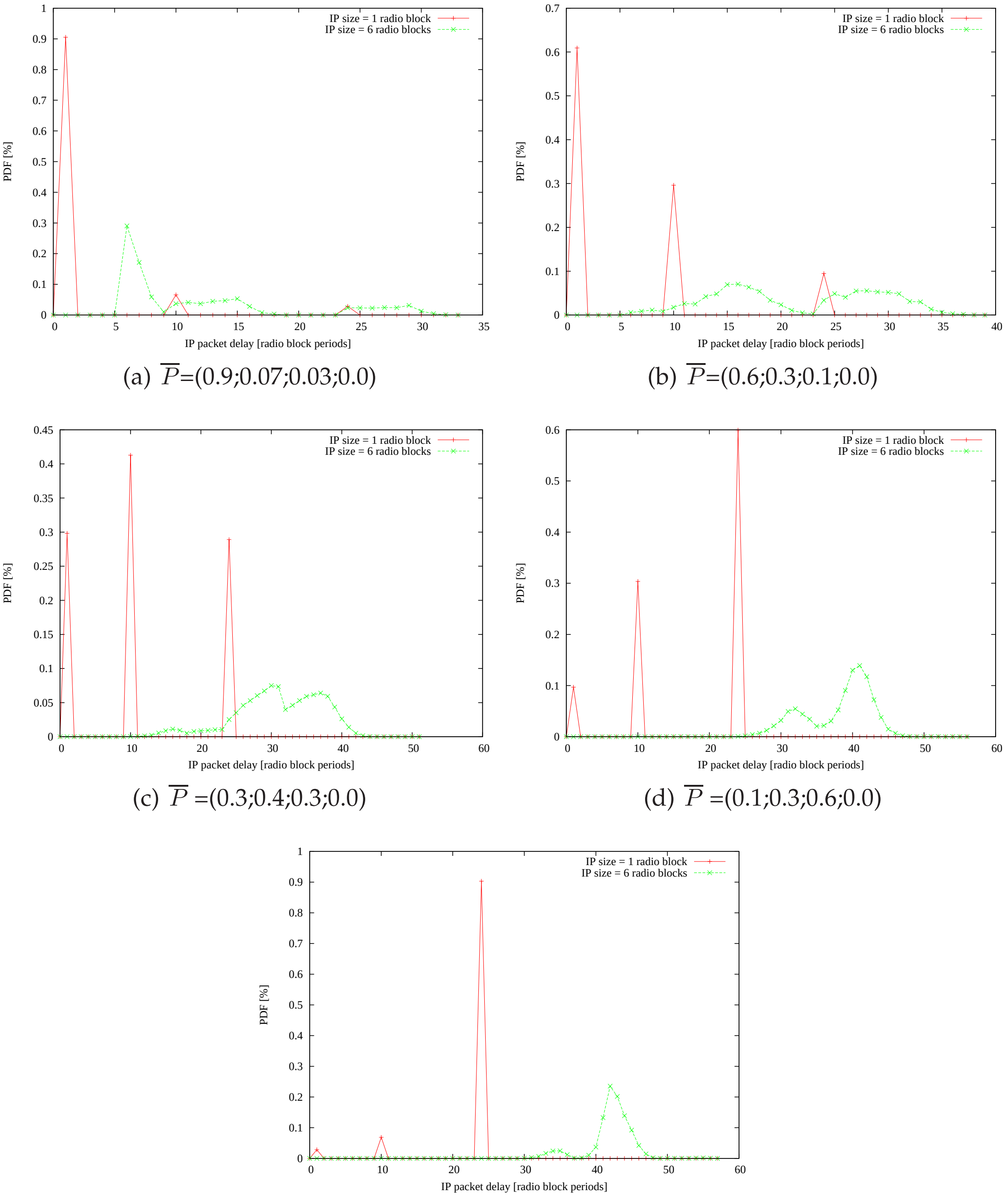
Figure C.15: *Uncertainty of IP packet delay - simulation results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

APPENDIX D

Distribution of IP packet delay for small packets: 1 and 6 radio blocks

This appendix presents graphs that show the distribution of the delay experienced for the transmission of small IP packets. The selected IP packet sizes that represent the small IP packets are one and six radio blocks.

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

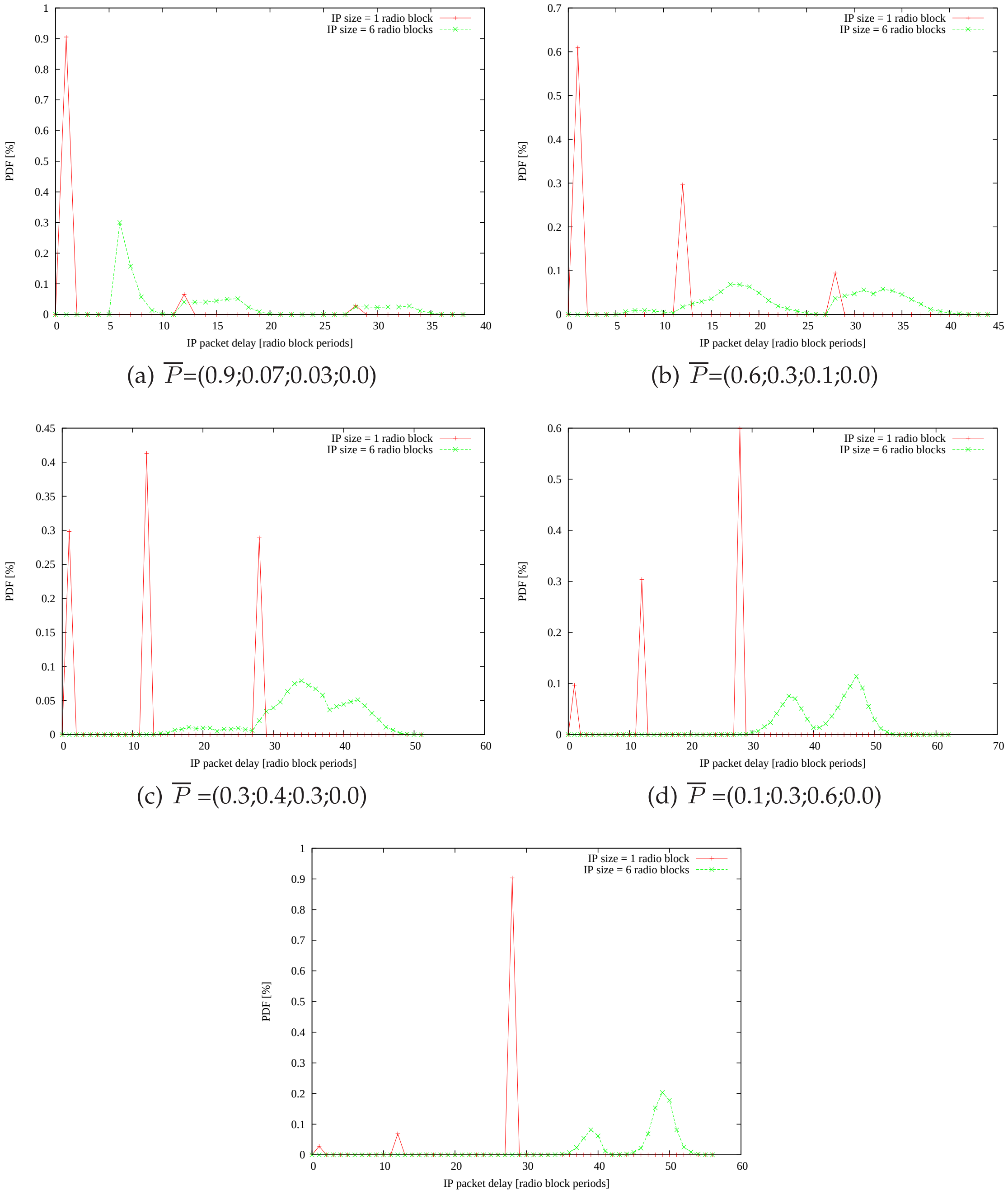
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.1: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

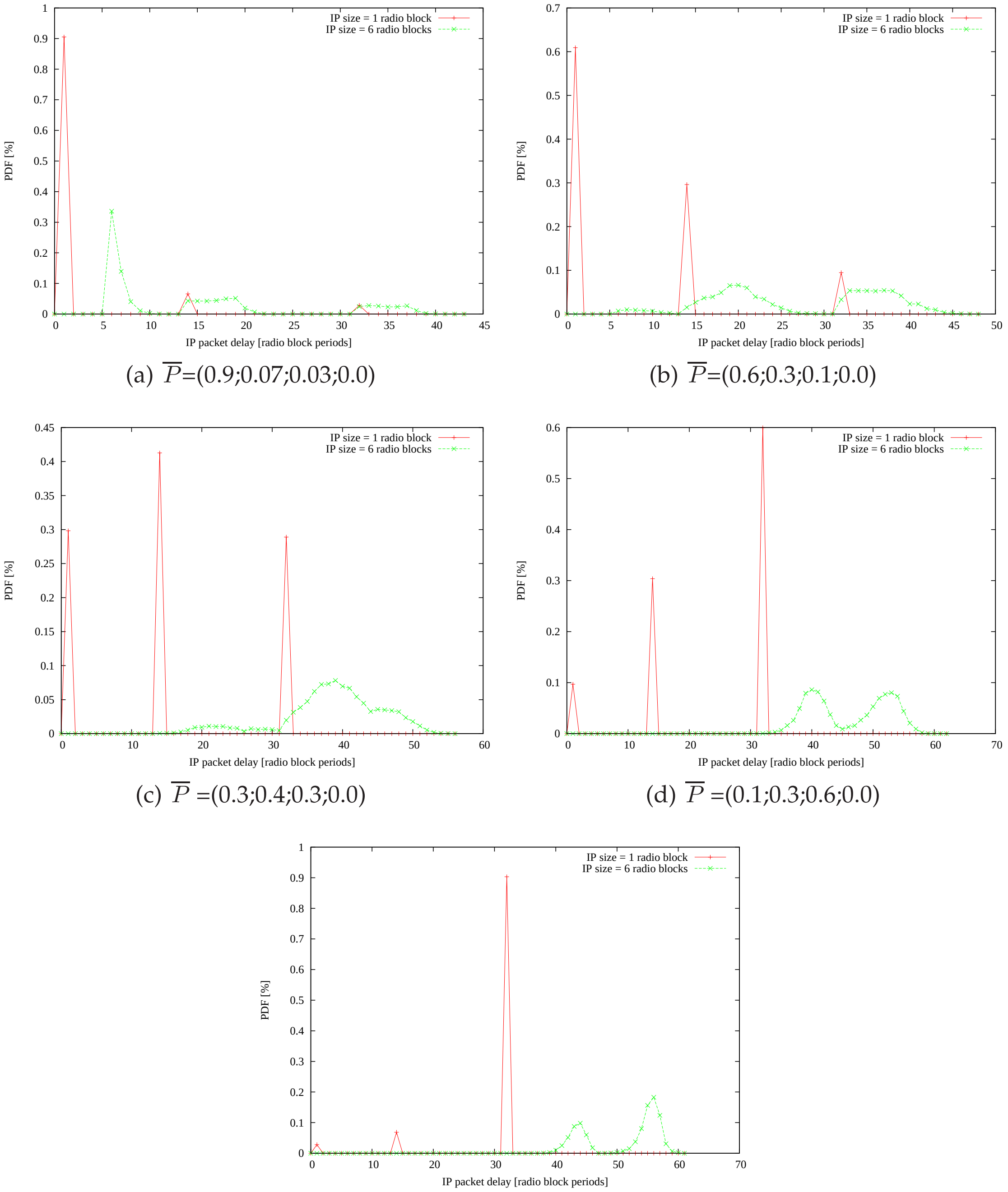
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.2: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

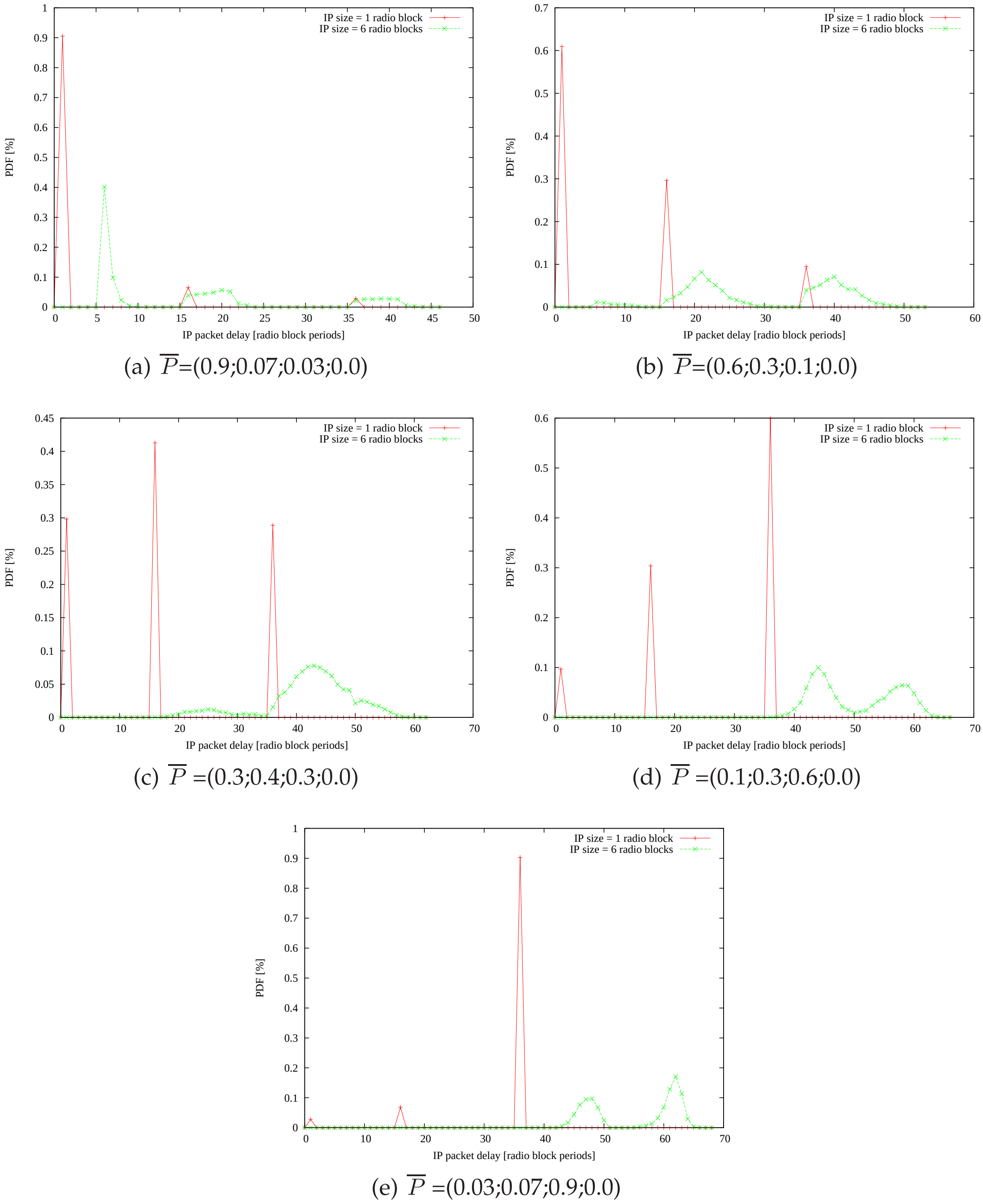
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.3: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

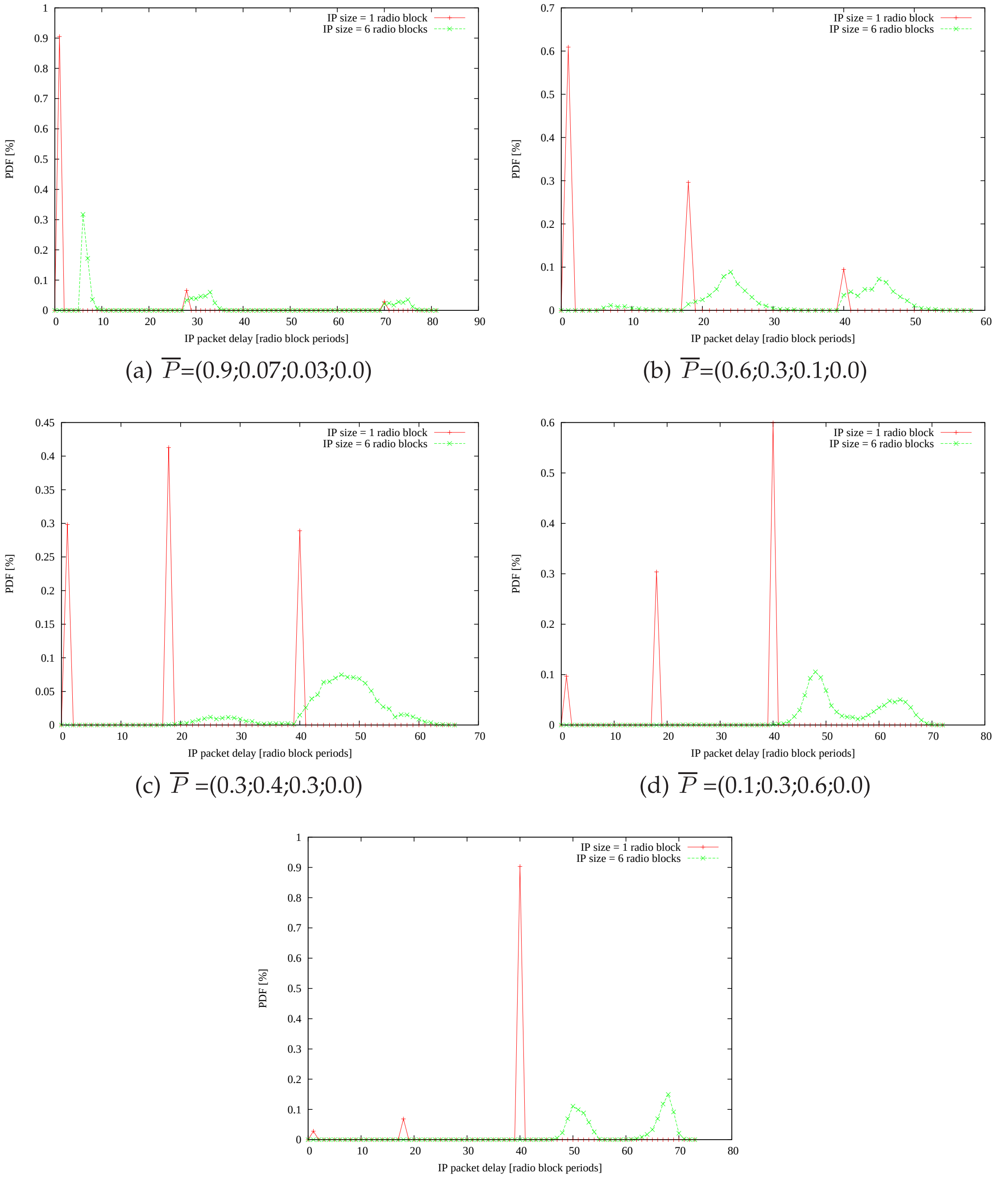
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.4: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

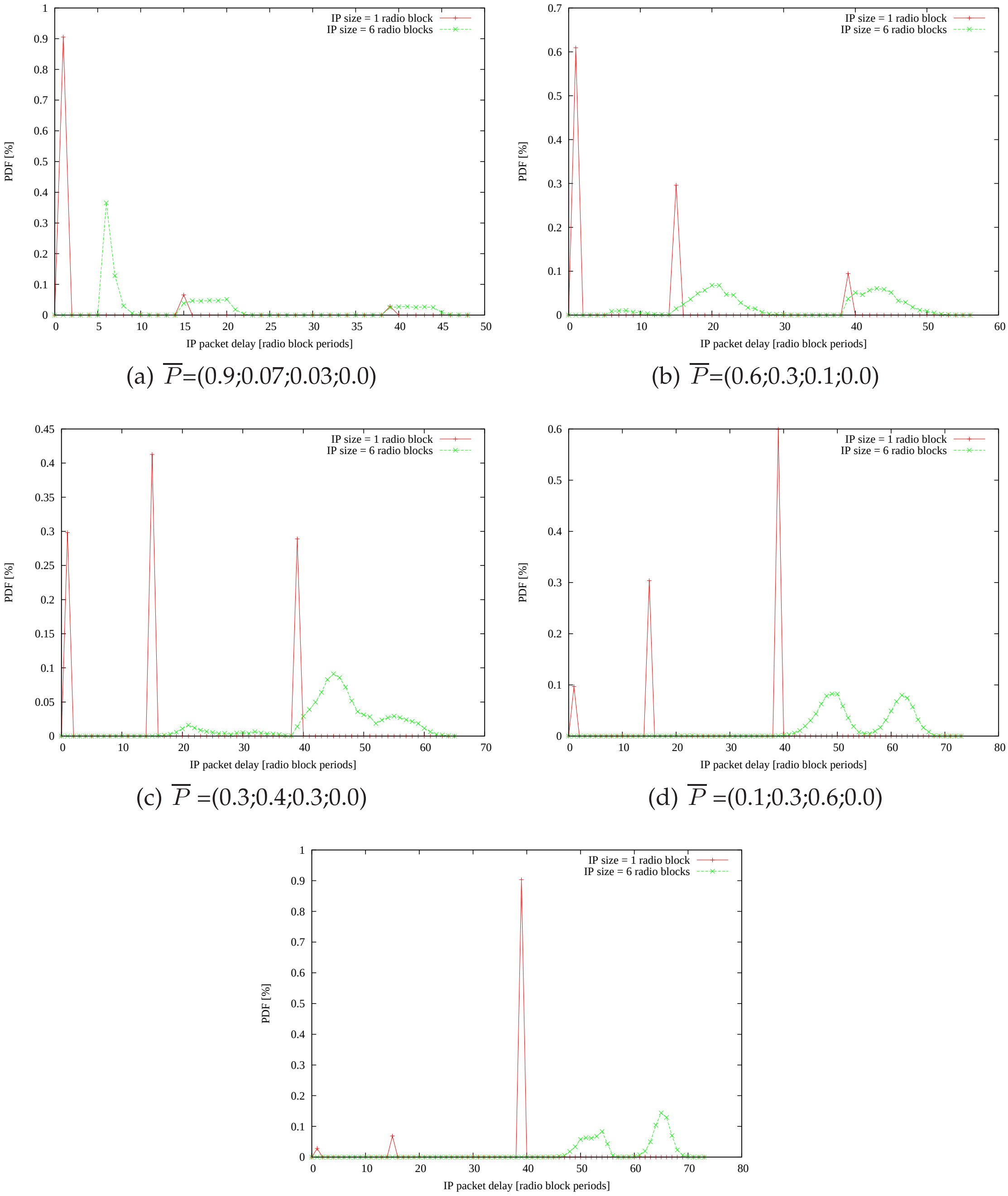
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.5: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

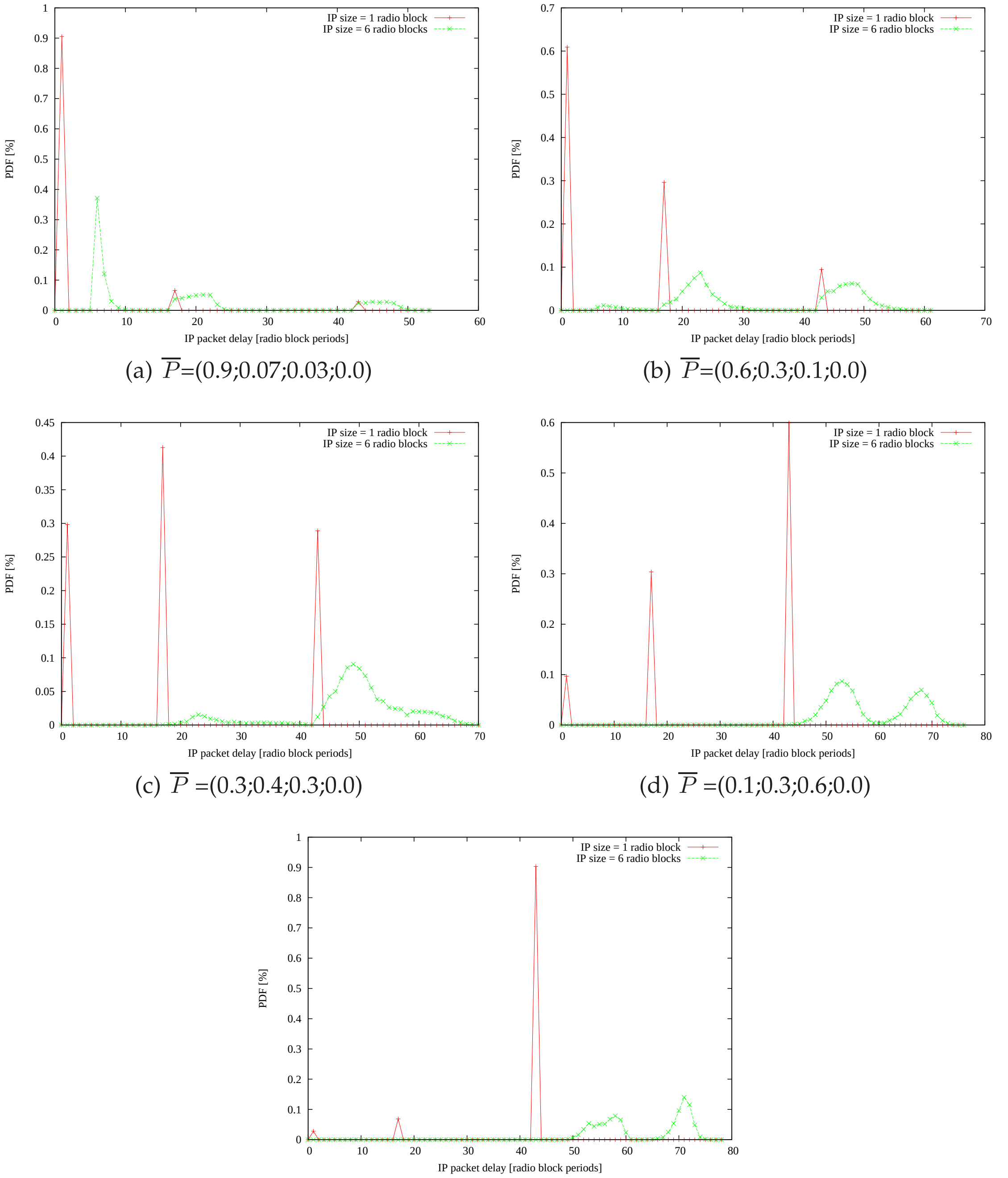
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.6: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

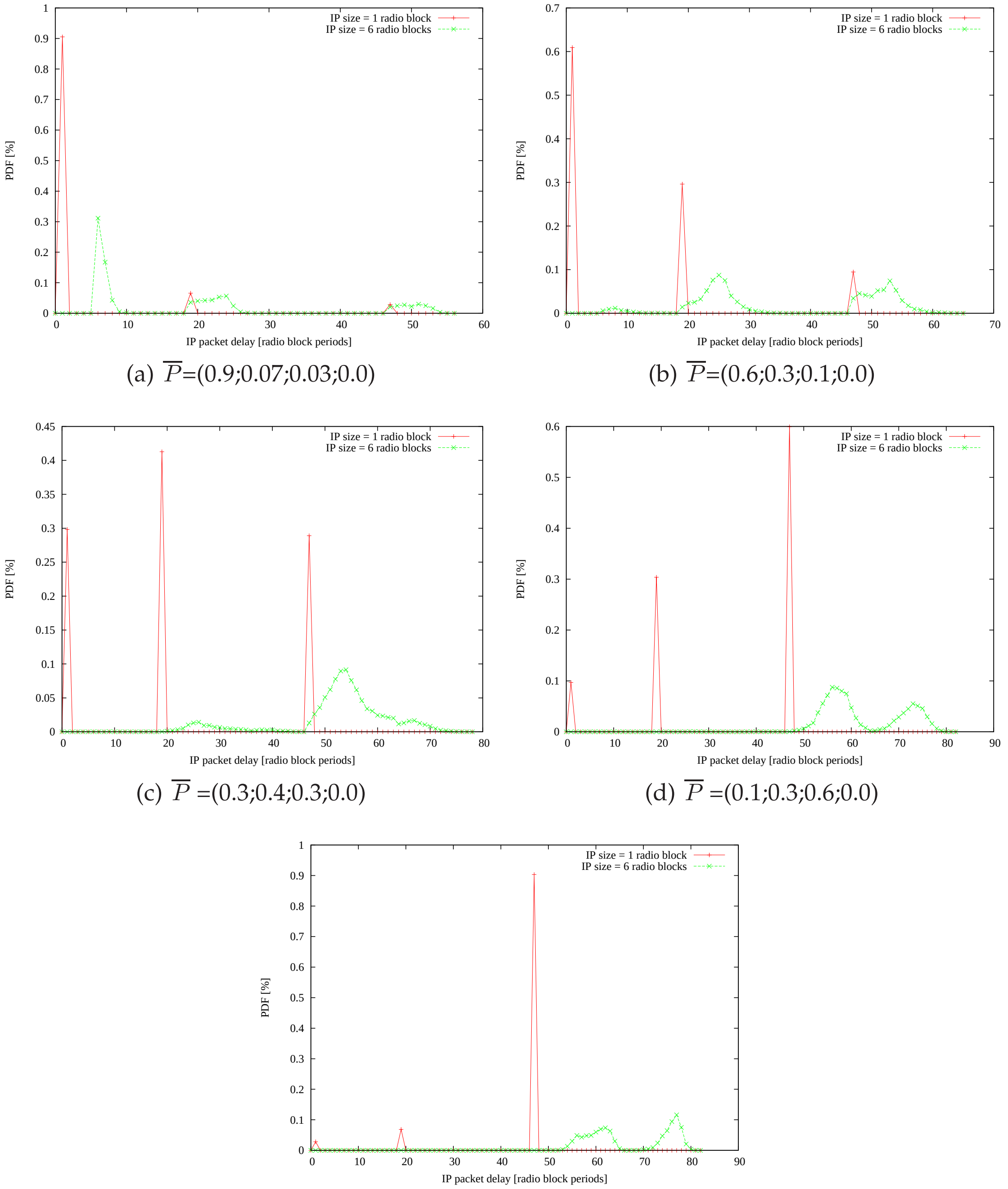
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.7: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

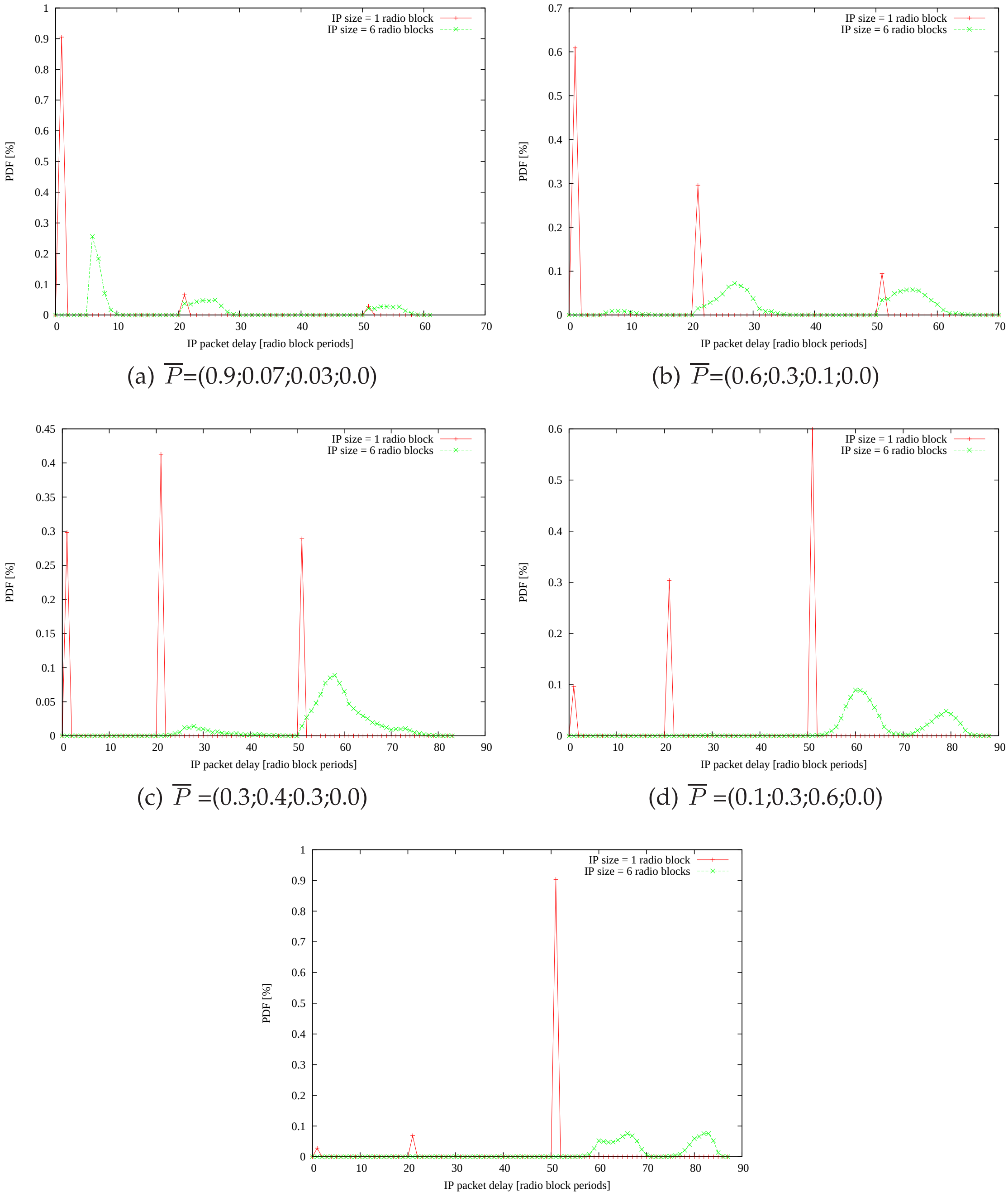
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.8: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

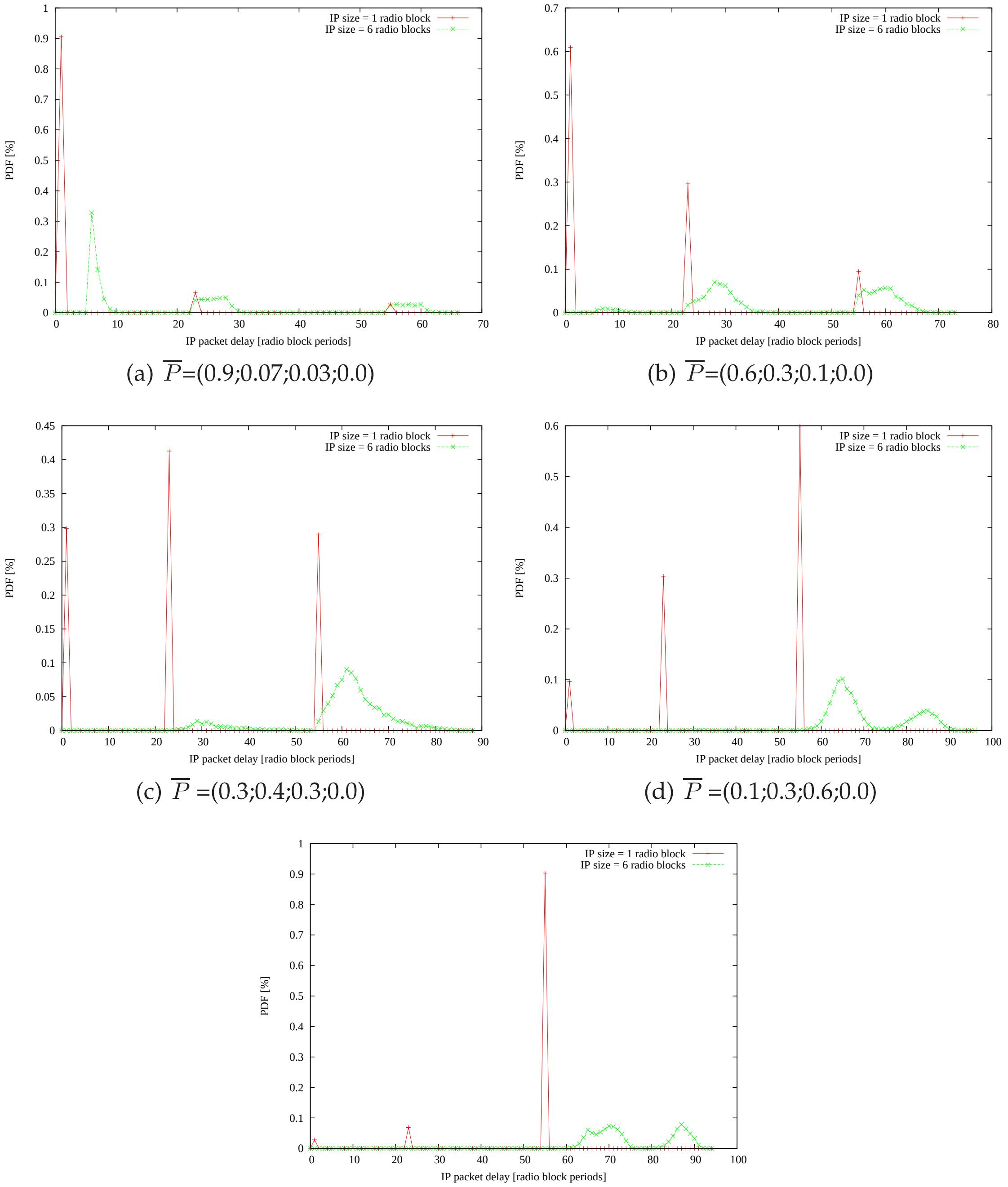
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.9: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

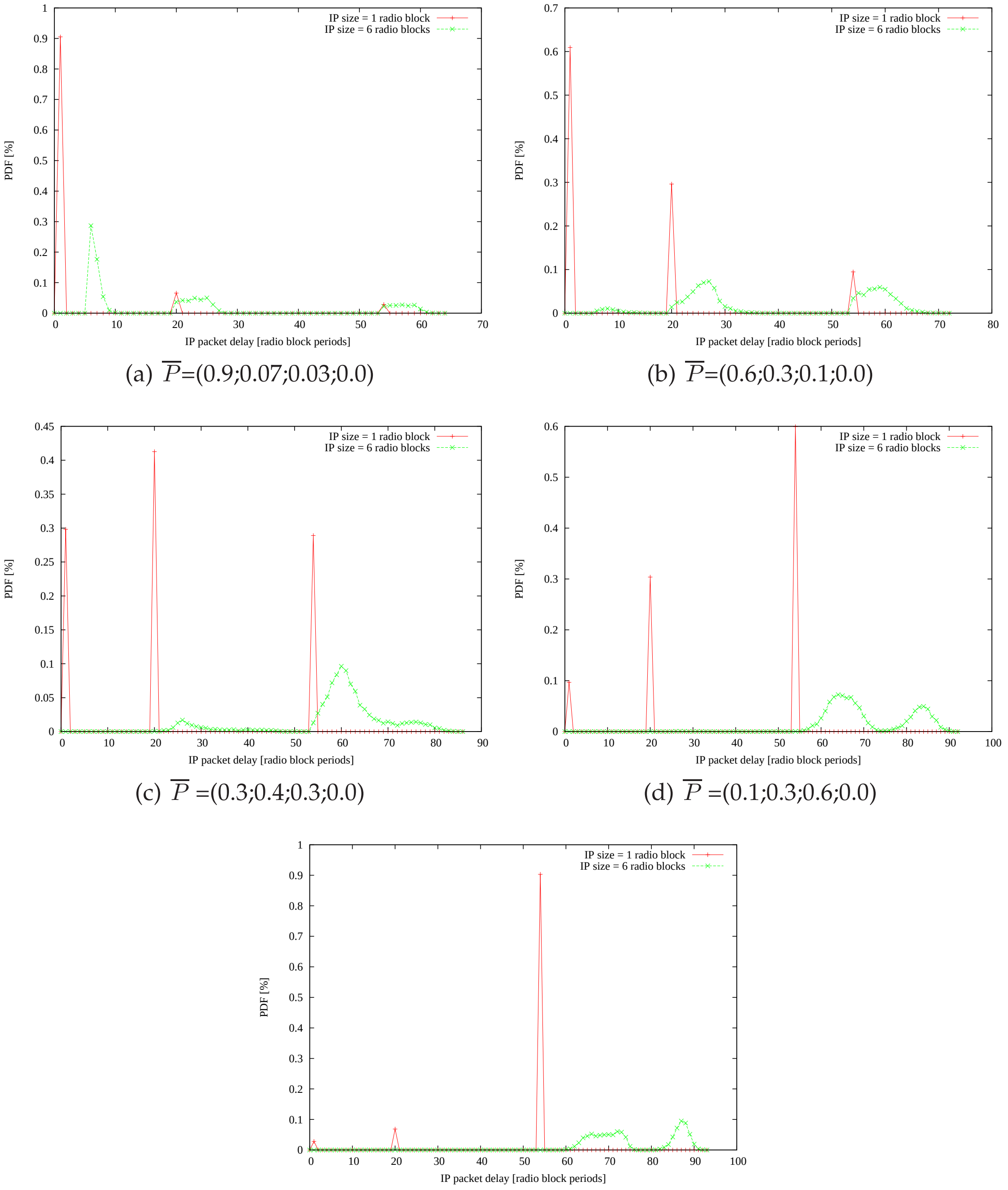
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.10: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

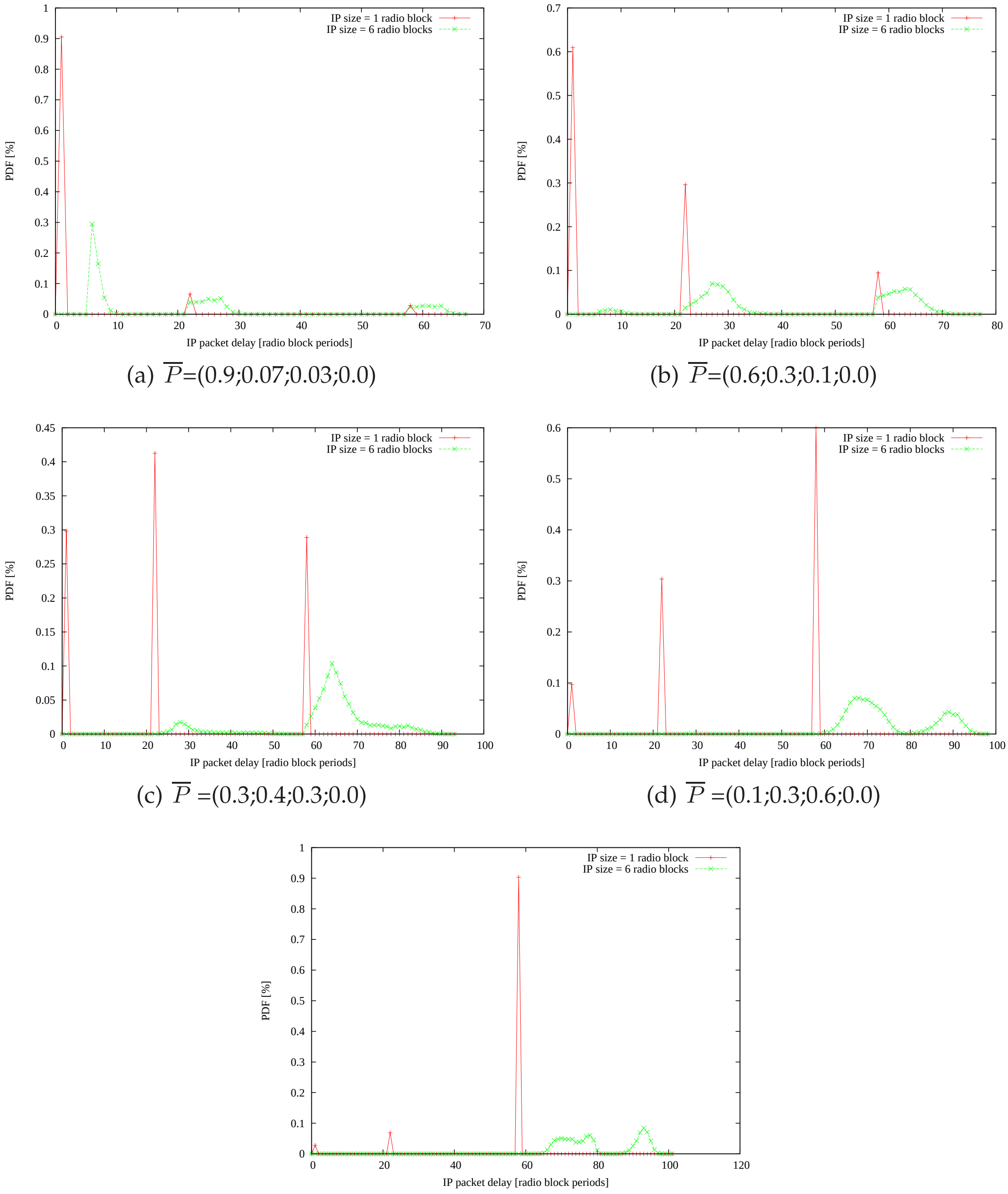
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.11: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

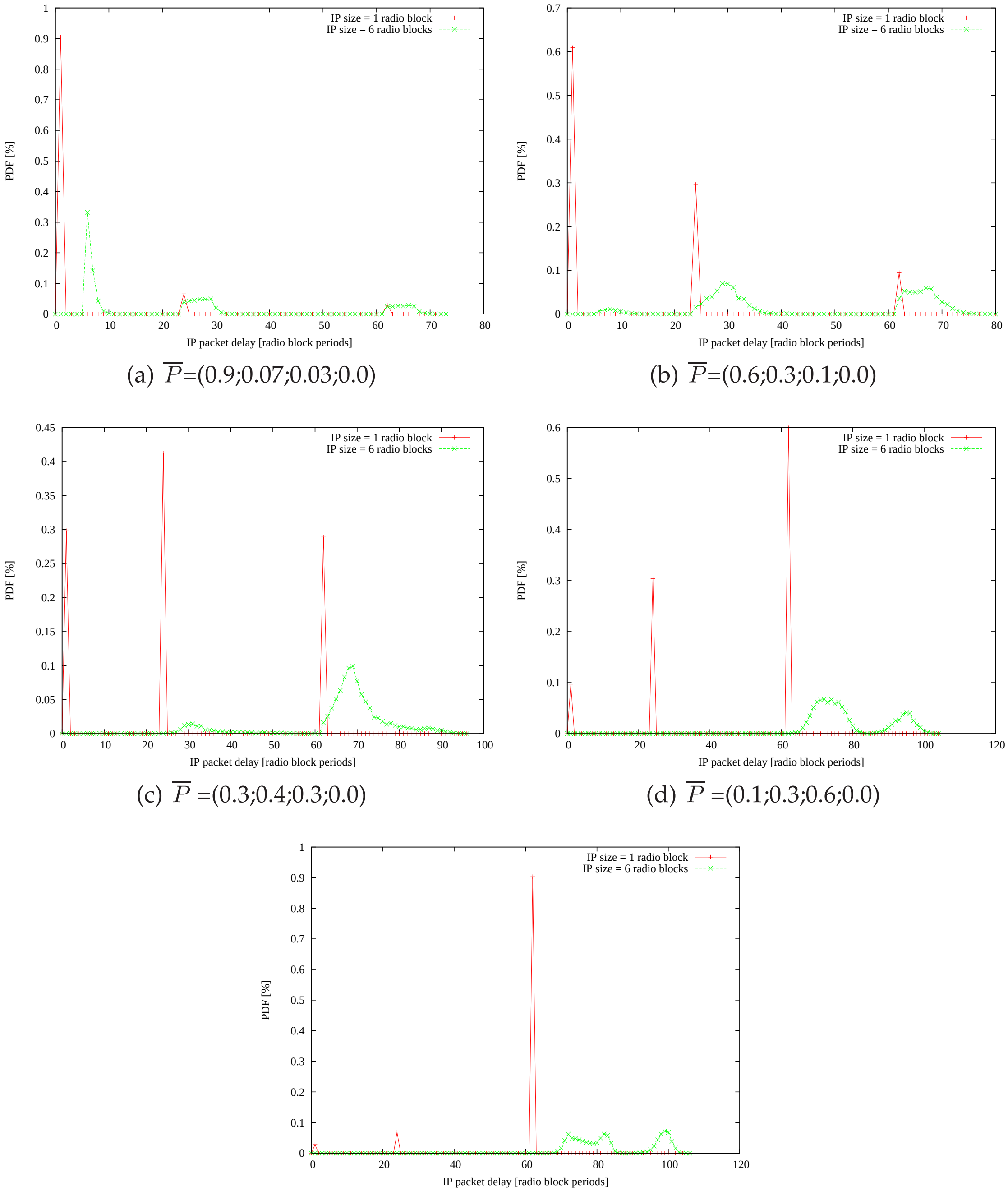
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.12: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

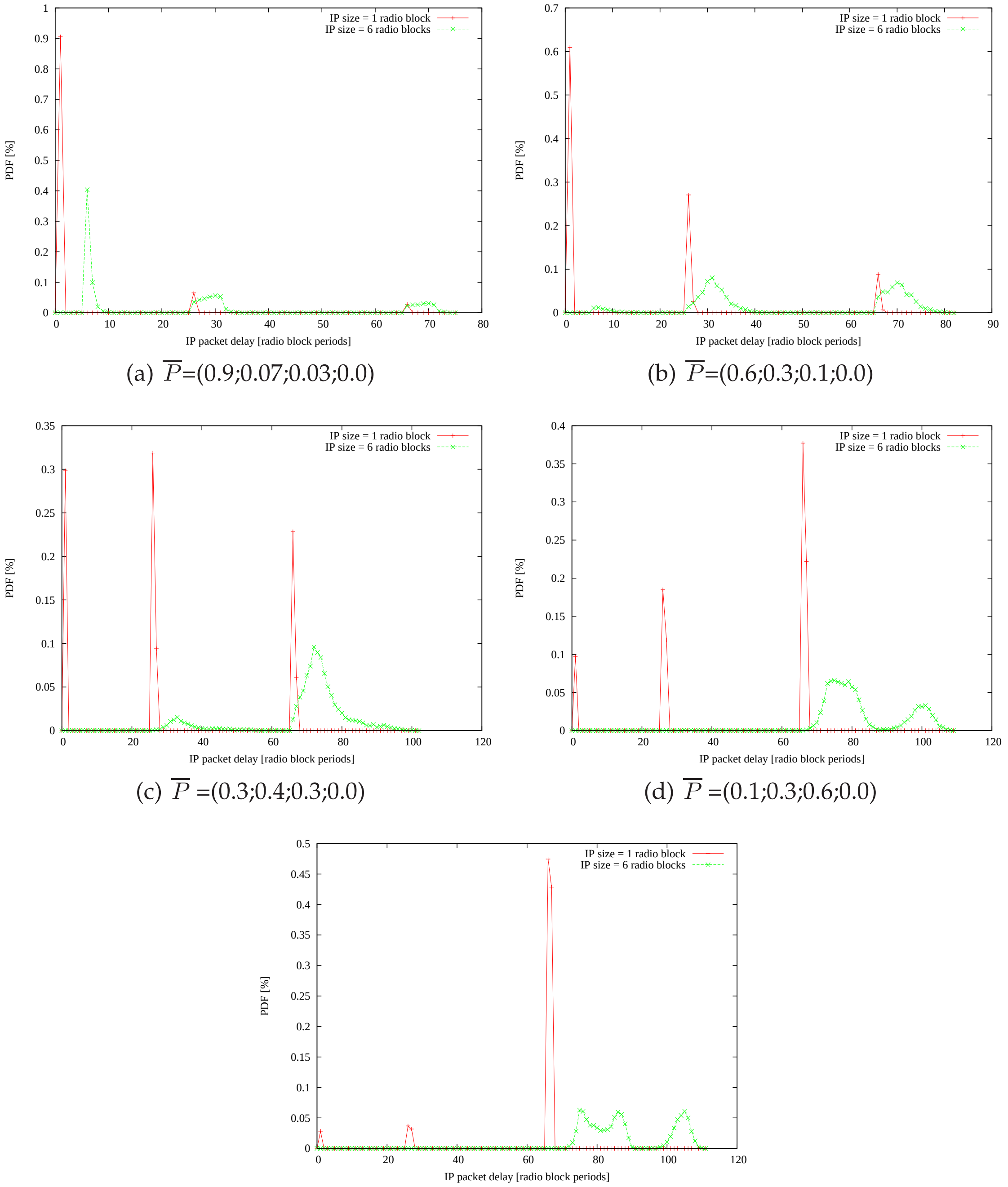
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.13: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

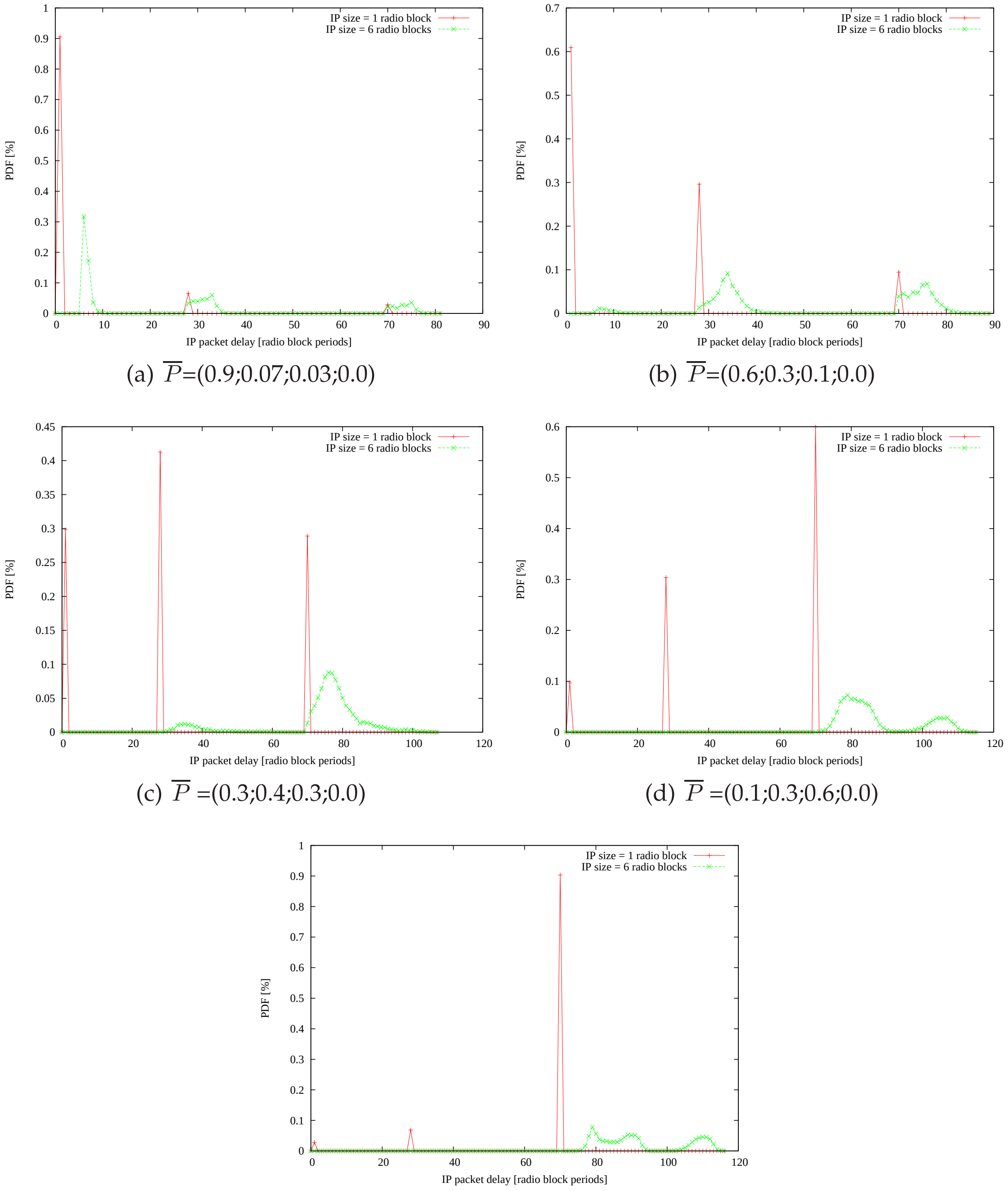
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure D.14: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

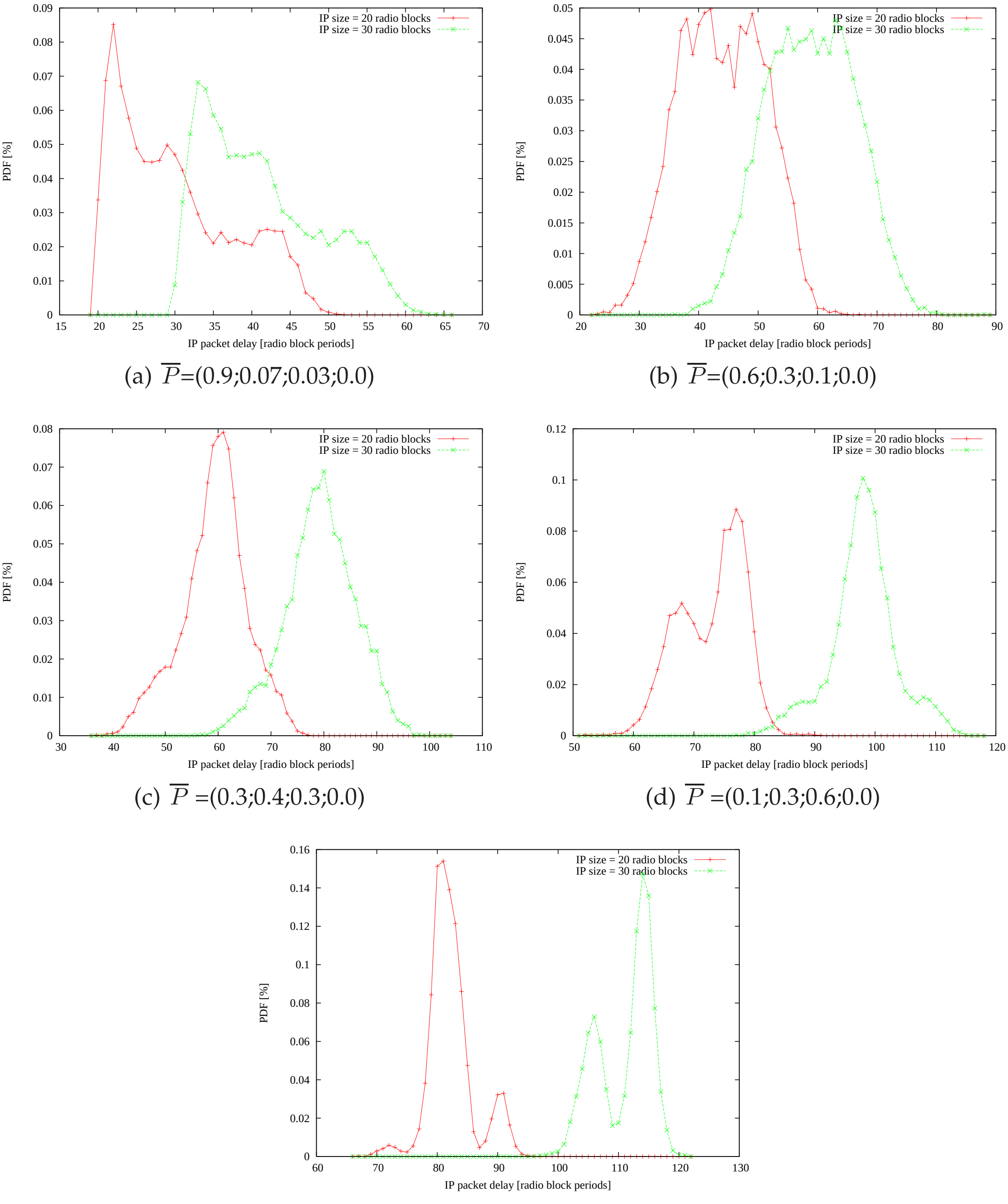
Figure D.15: *Distribution of IP packet delay for two different small packet sizes, one and six radio blocks - simulation results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

APPENDIX E

Distribution of IP packet delay for large packets: 20 and 30 radio blocks

This appendix presents graphs that show the distribution of the delay experienced for the transmission of large IP packets. The selected IP packet sizes that represent the large IP packets are twenty and thirty radio blocks.

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

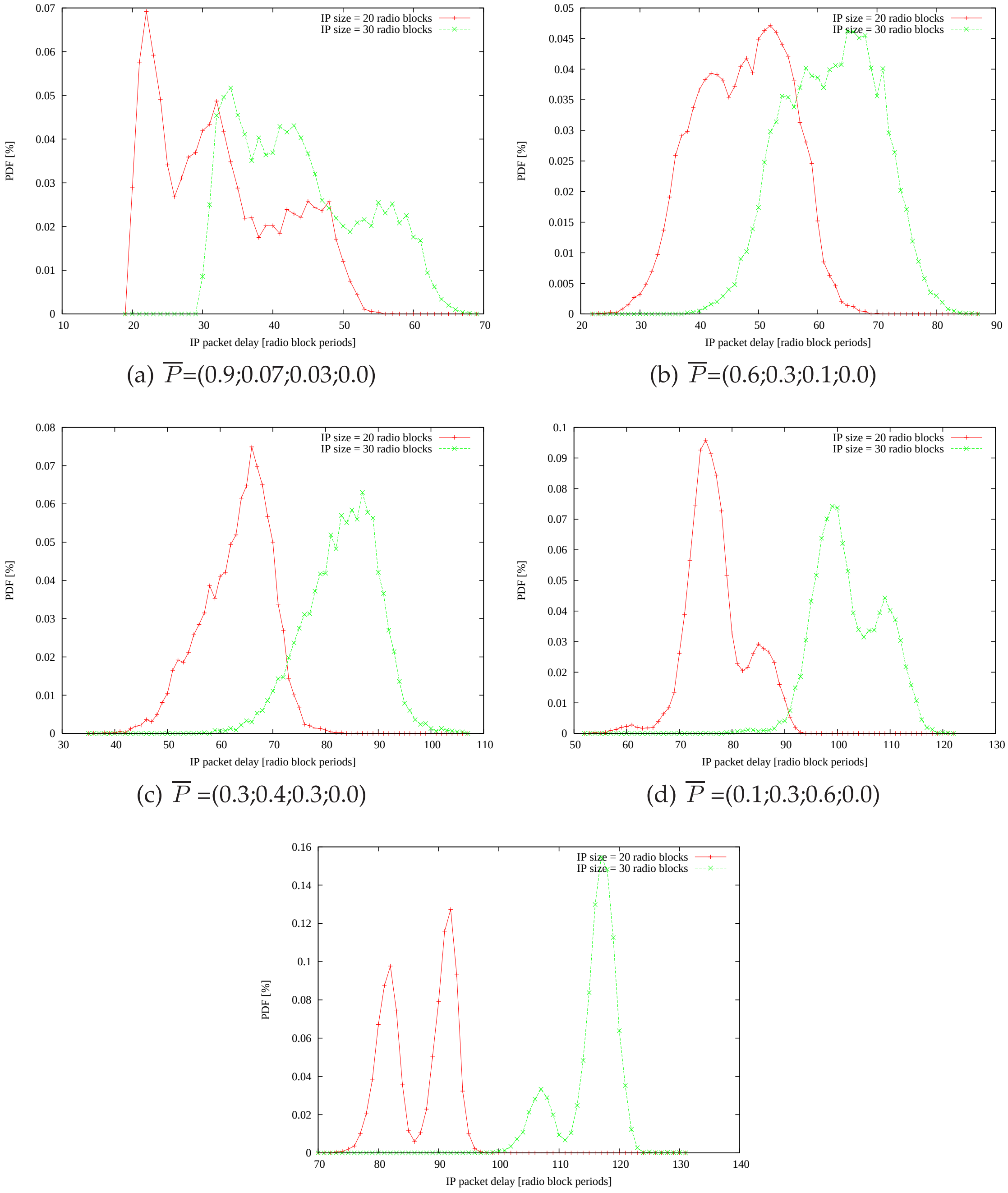
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.1: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

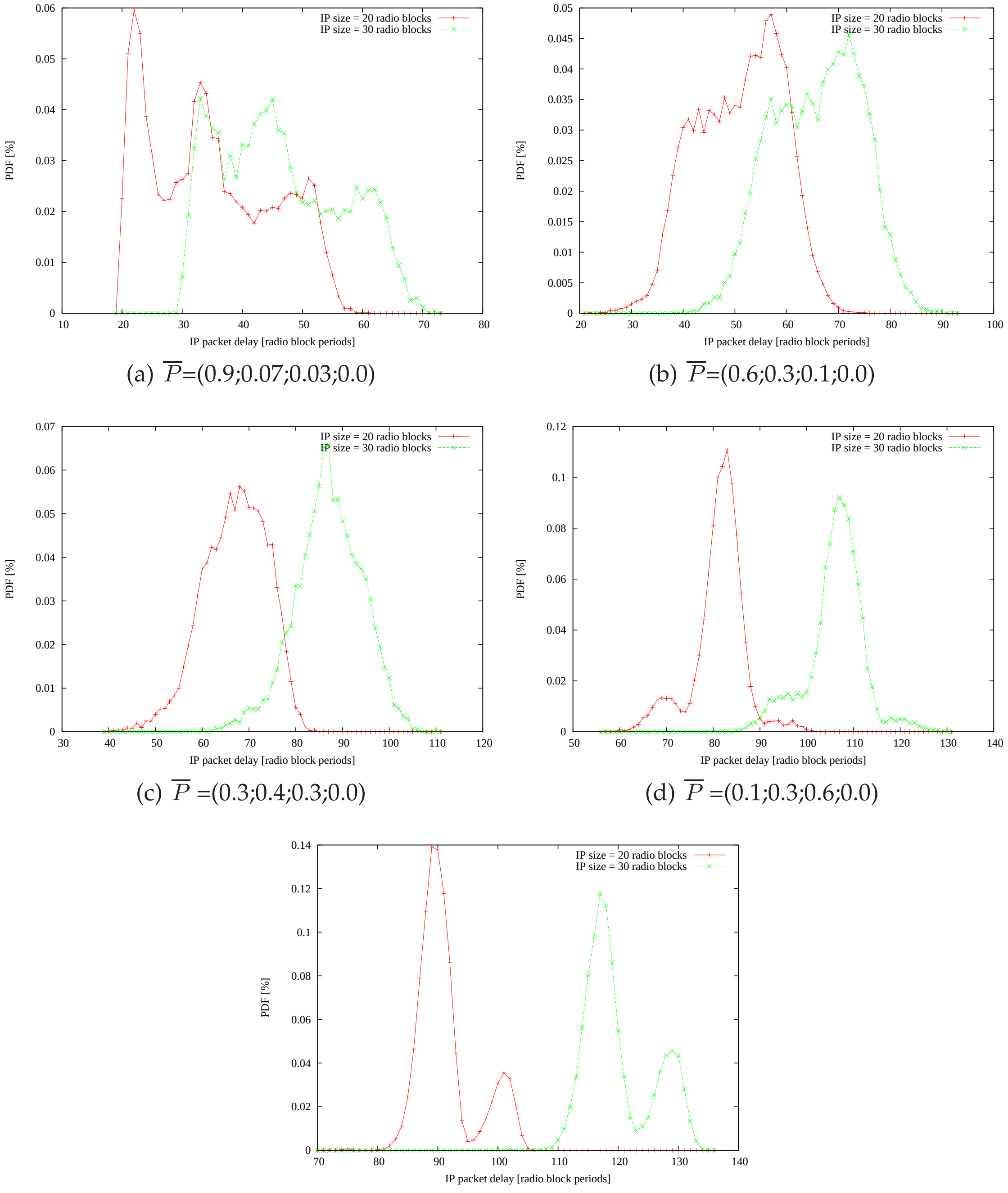
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.2: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

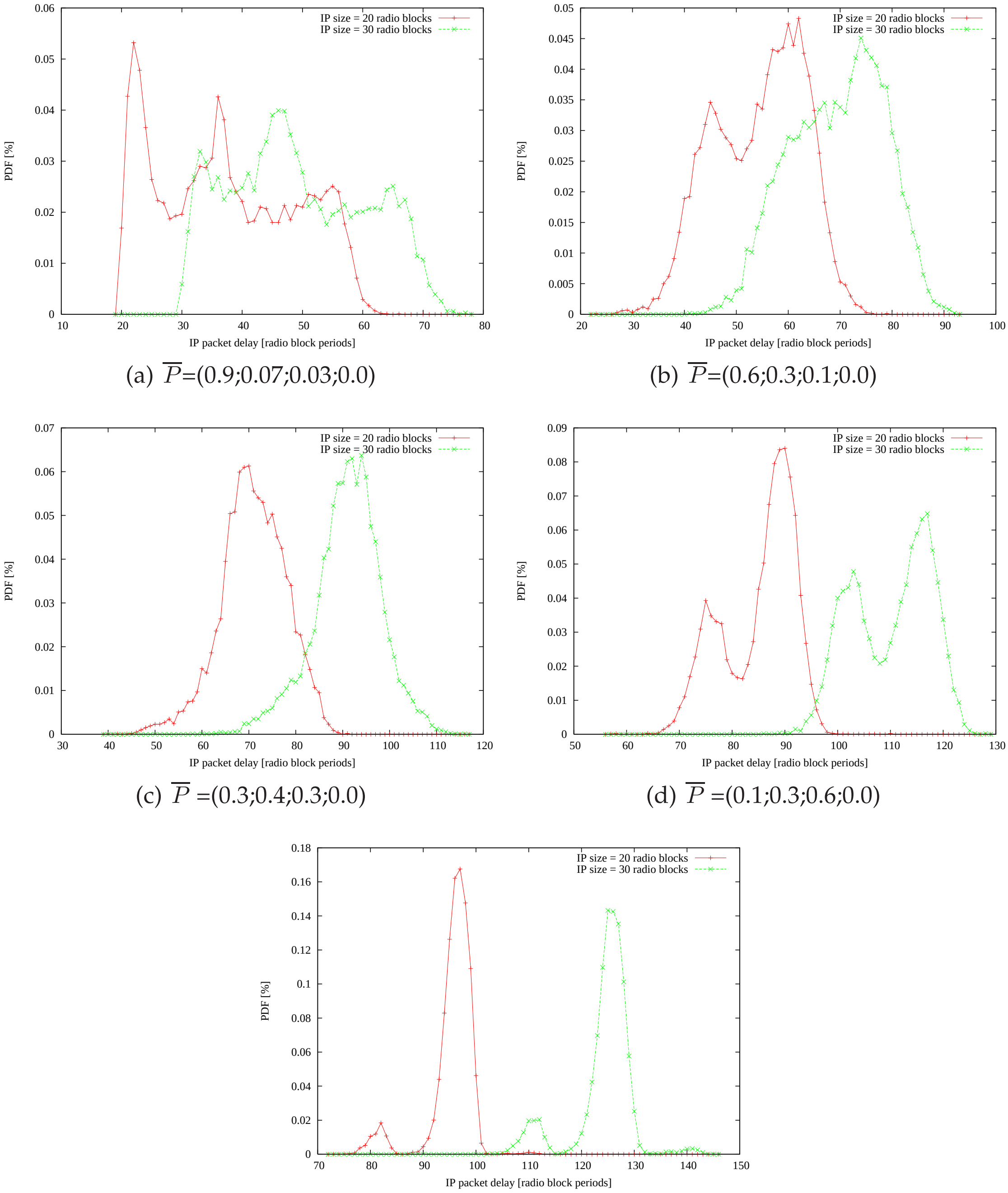
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.3: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

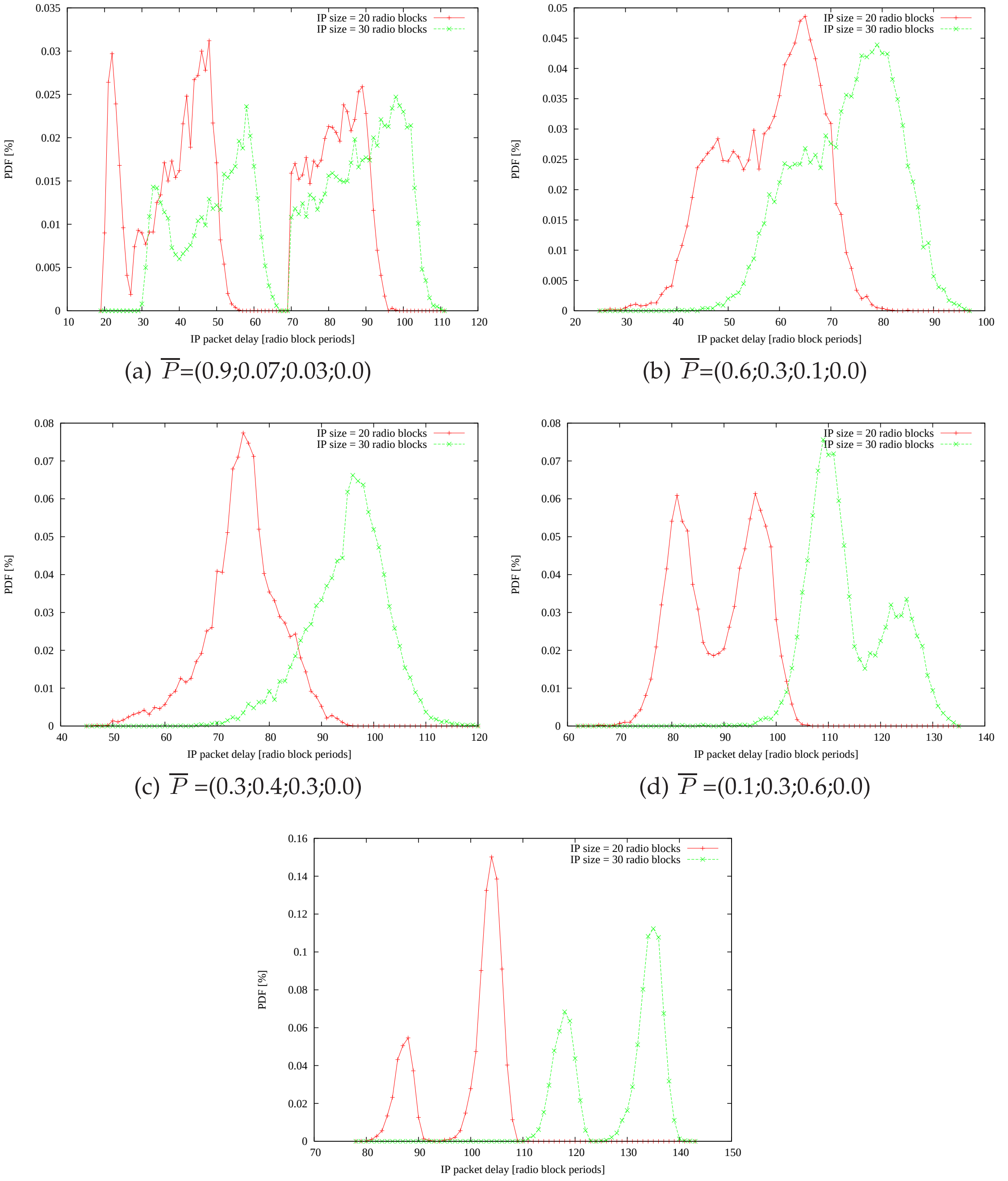
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.4: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

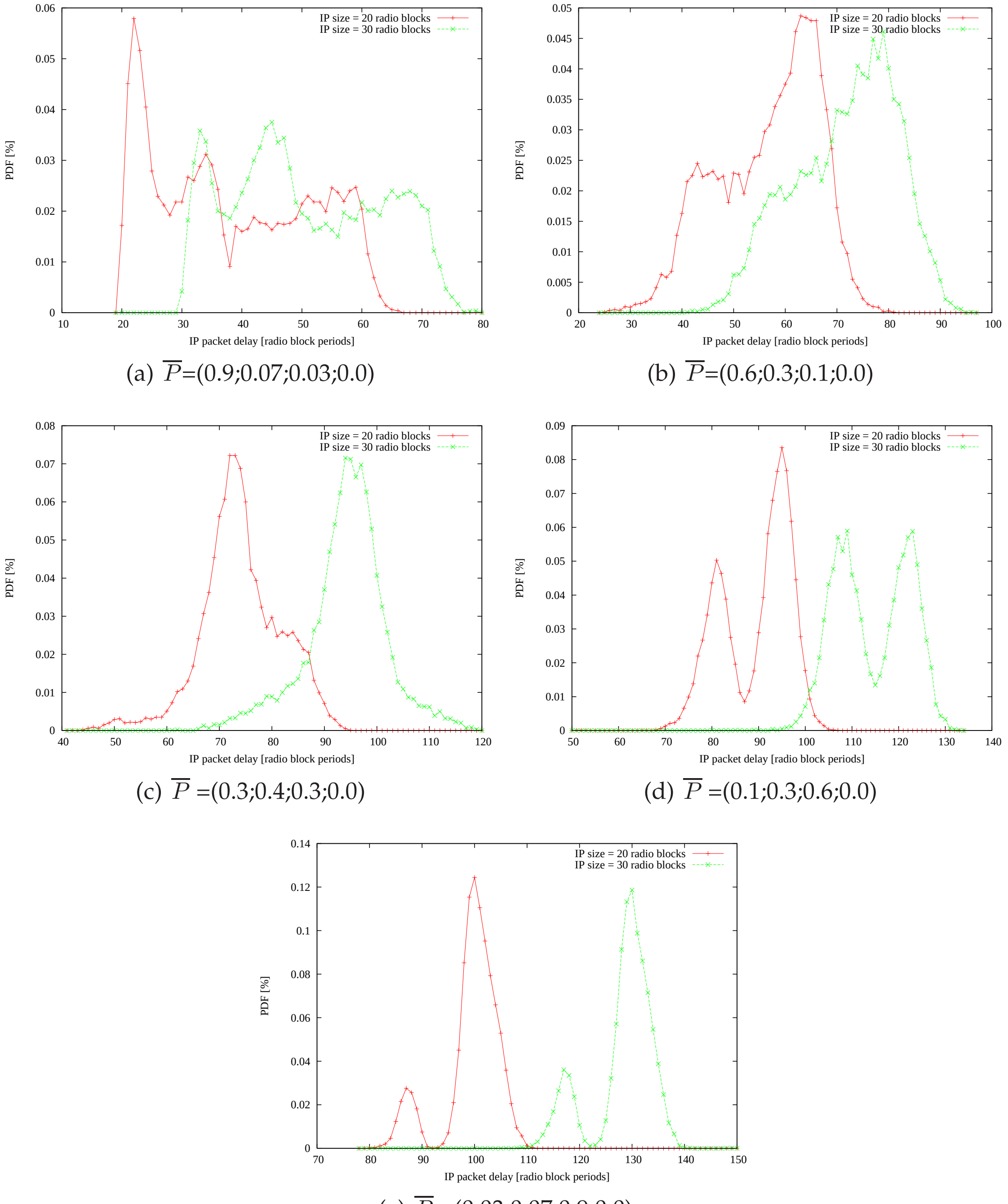
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.5: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

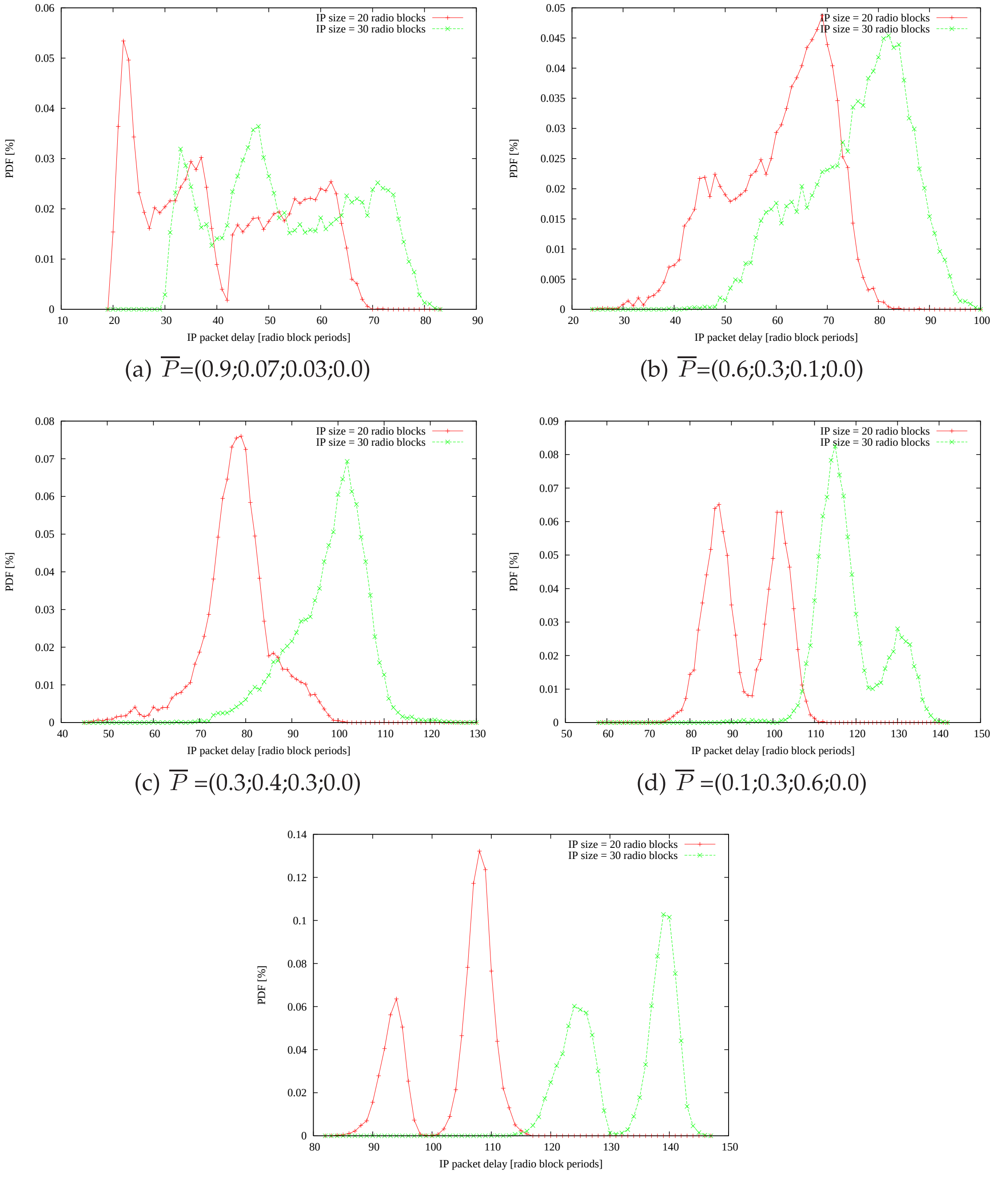
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.6: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

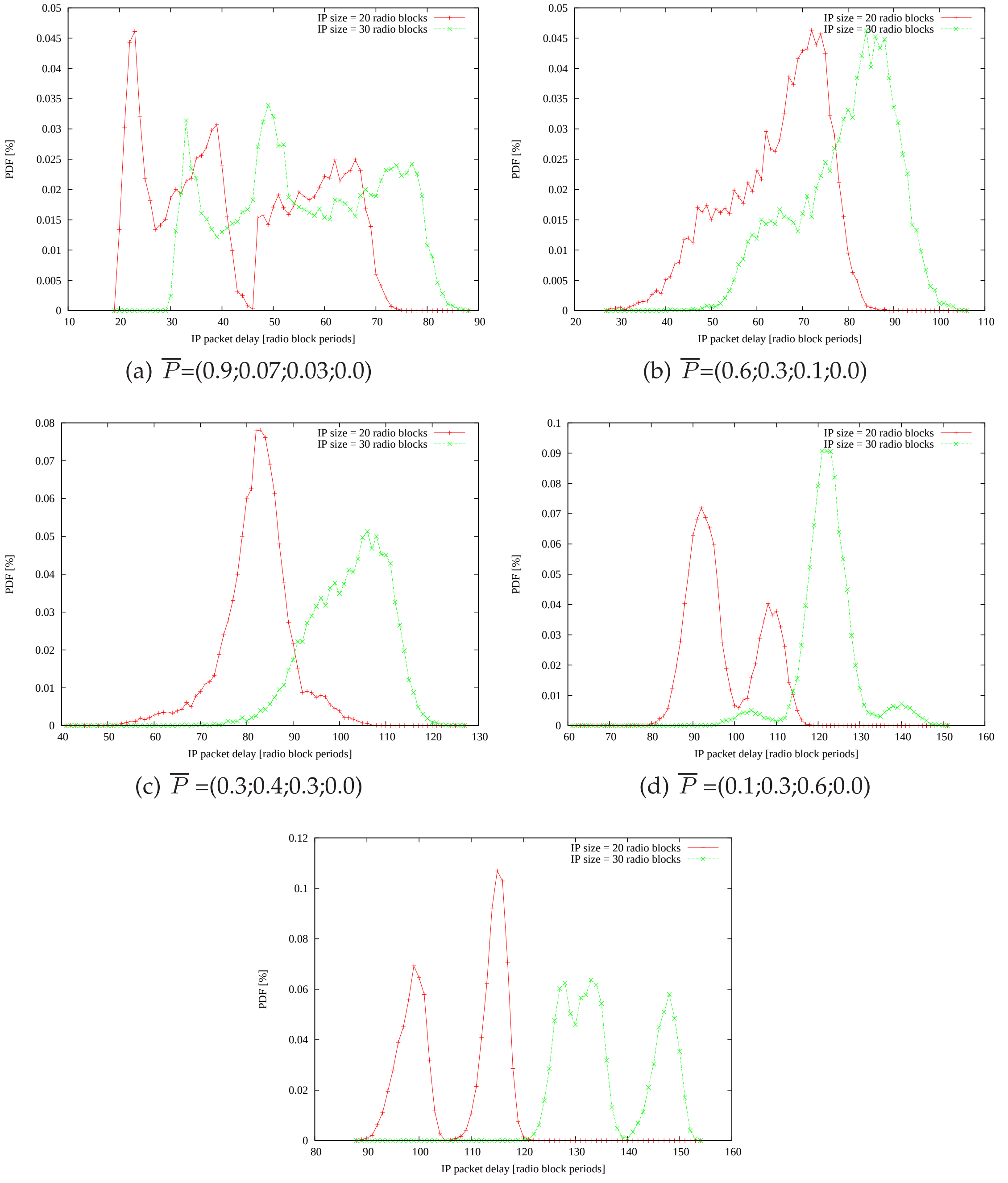
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.7: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

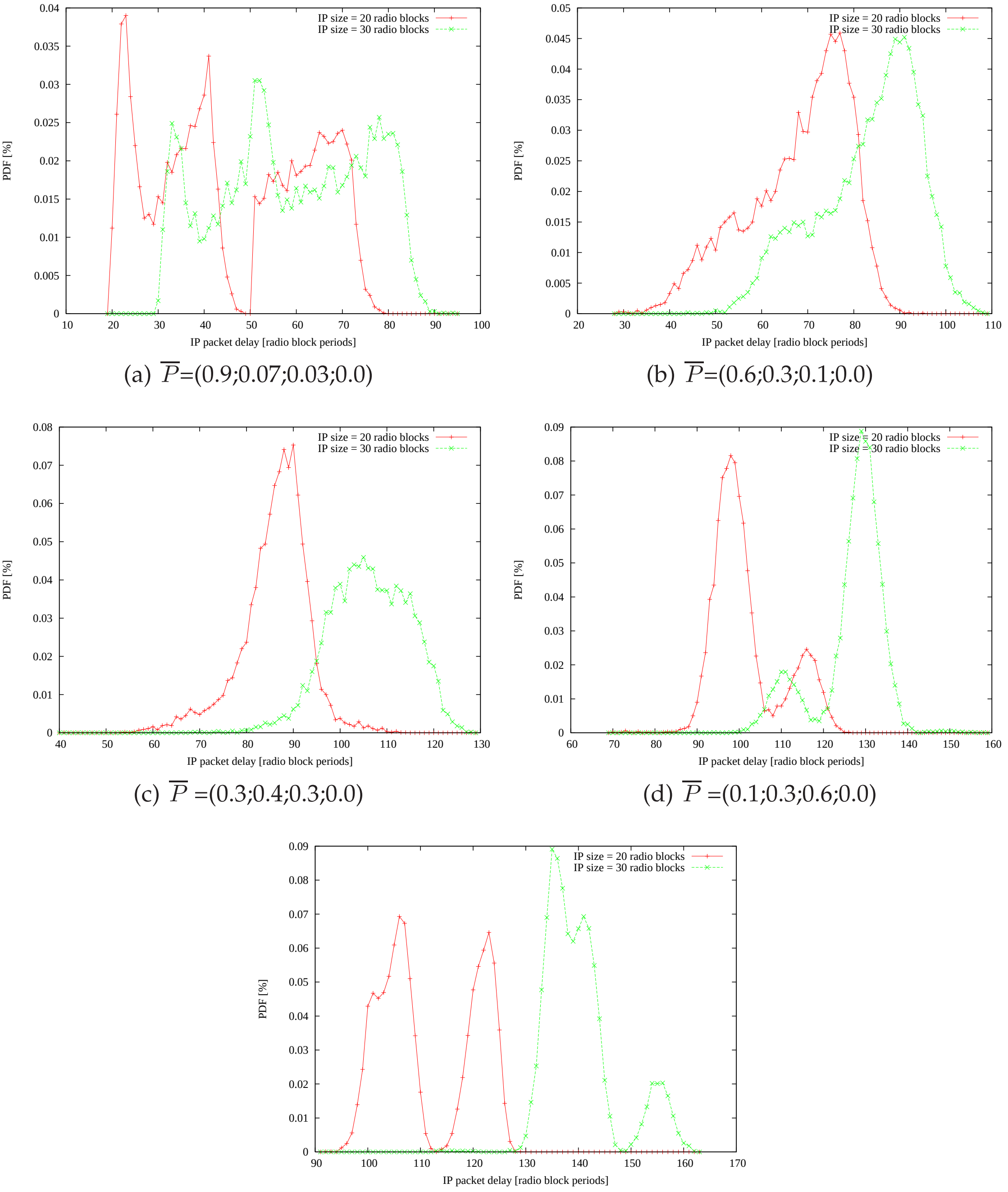
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.8: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

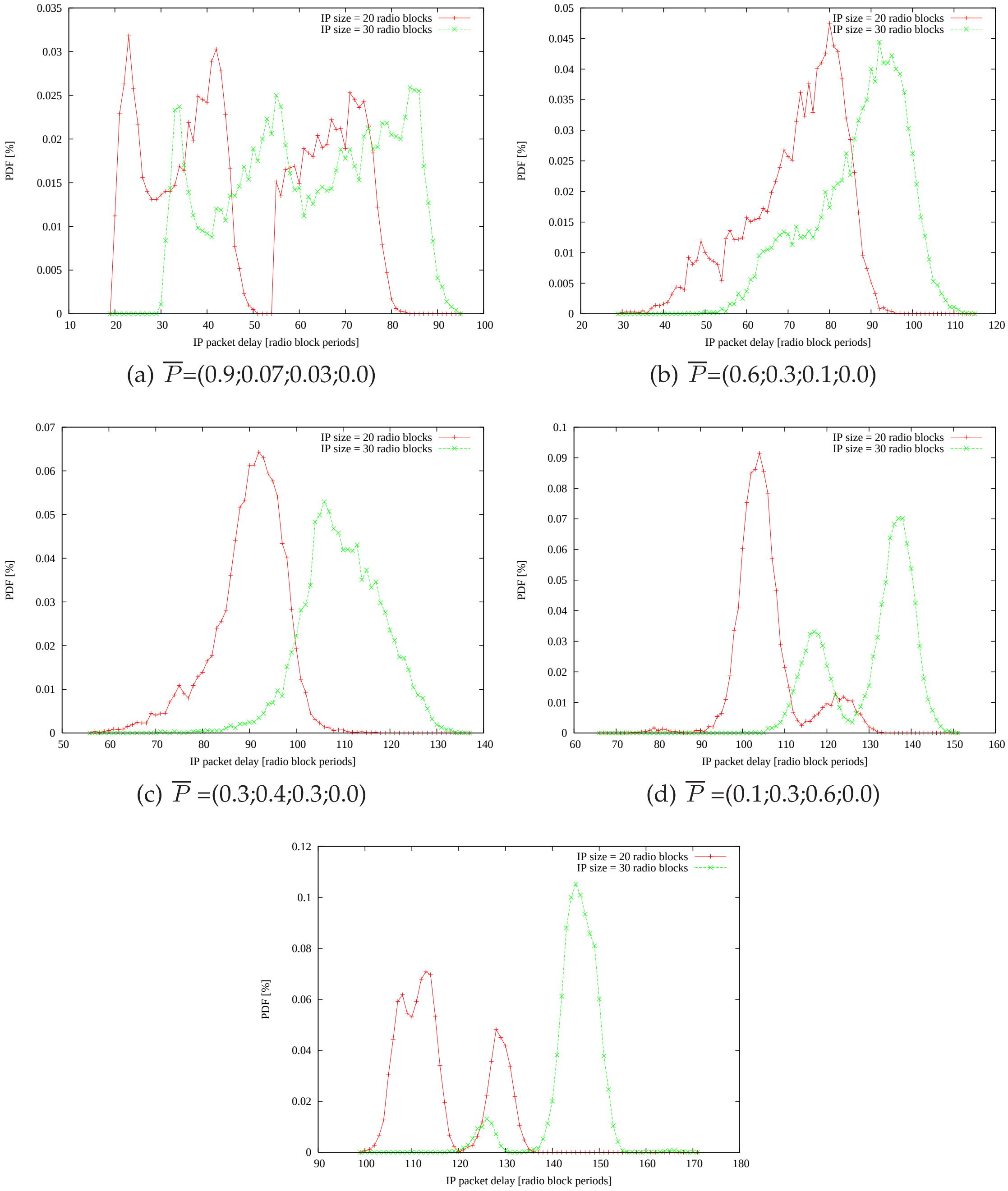
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.9: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

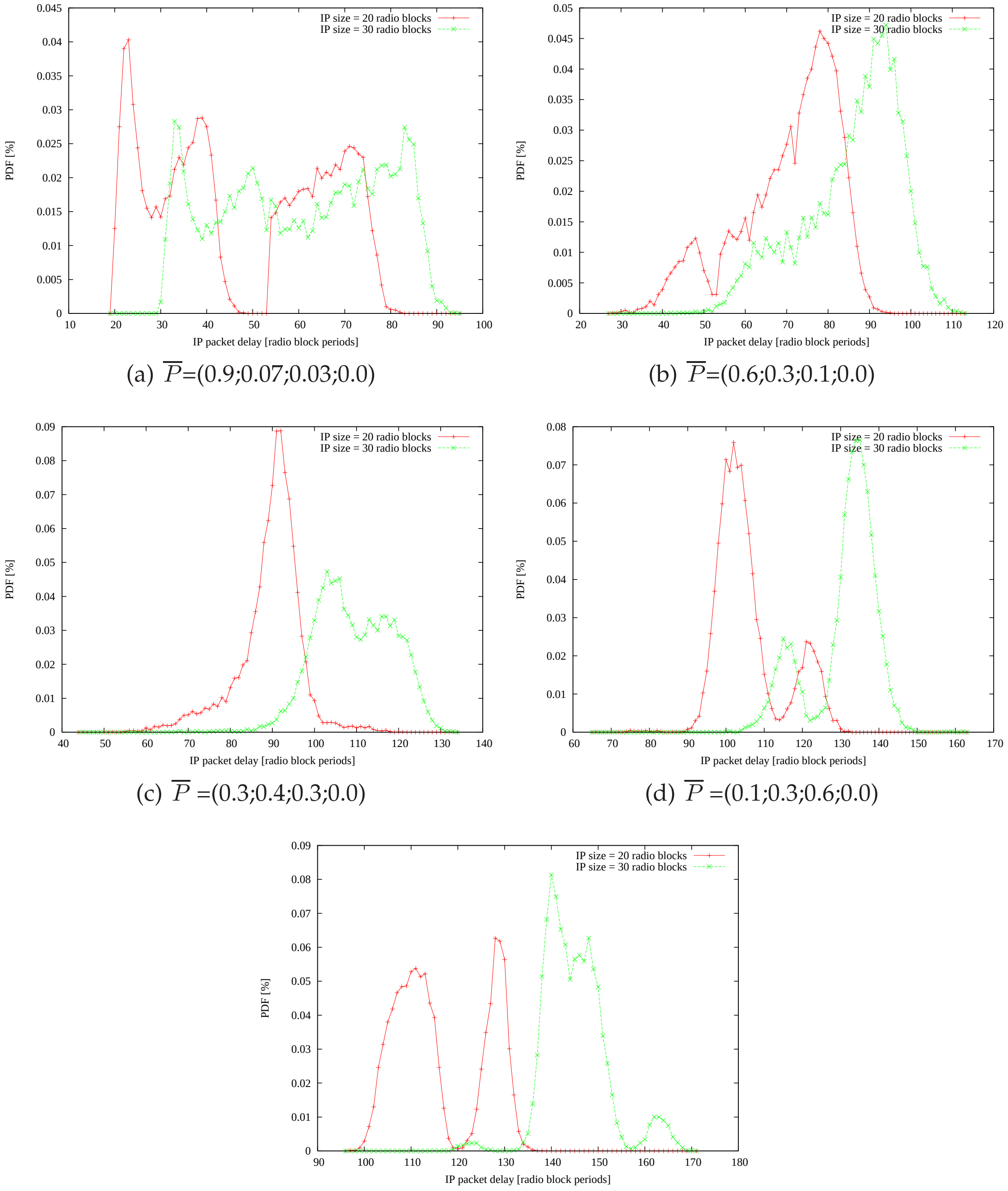
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.10: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

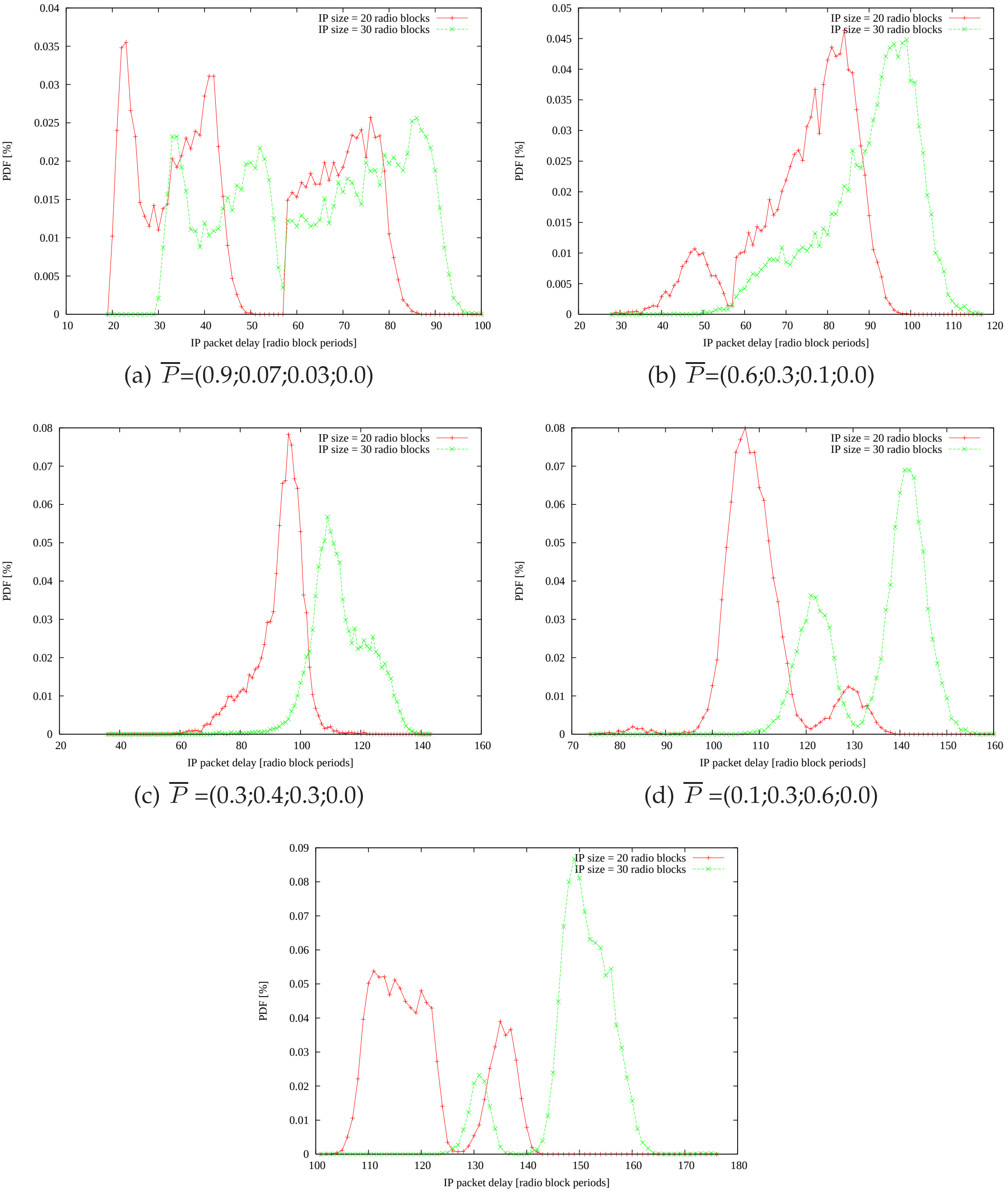
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.11: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

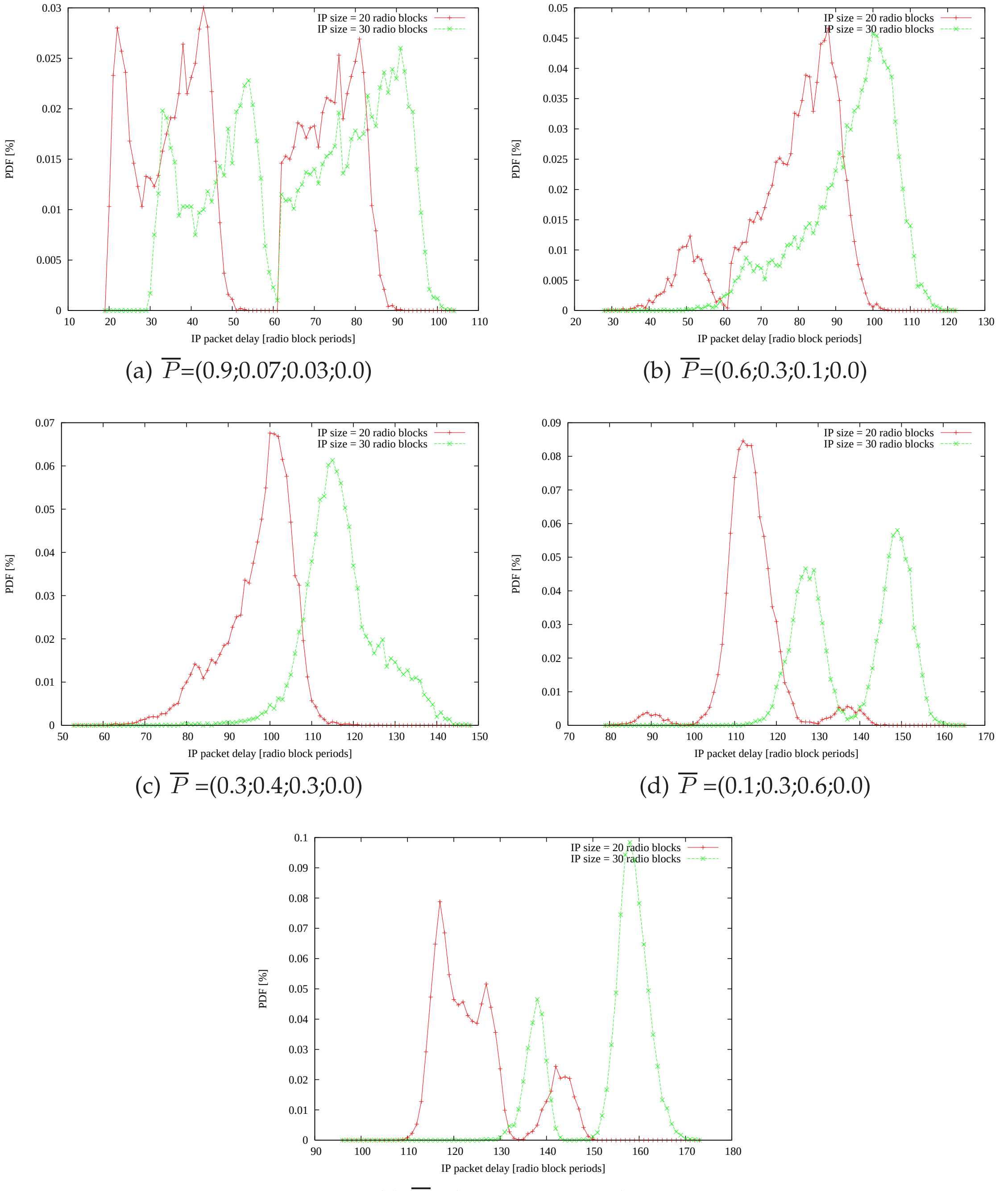
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.12: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

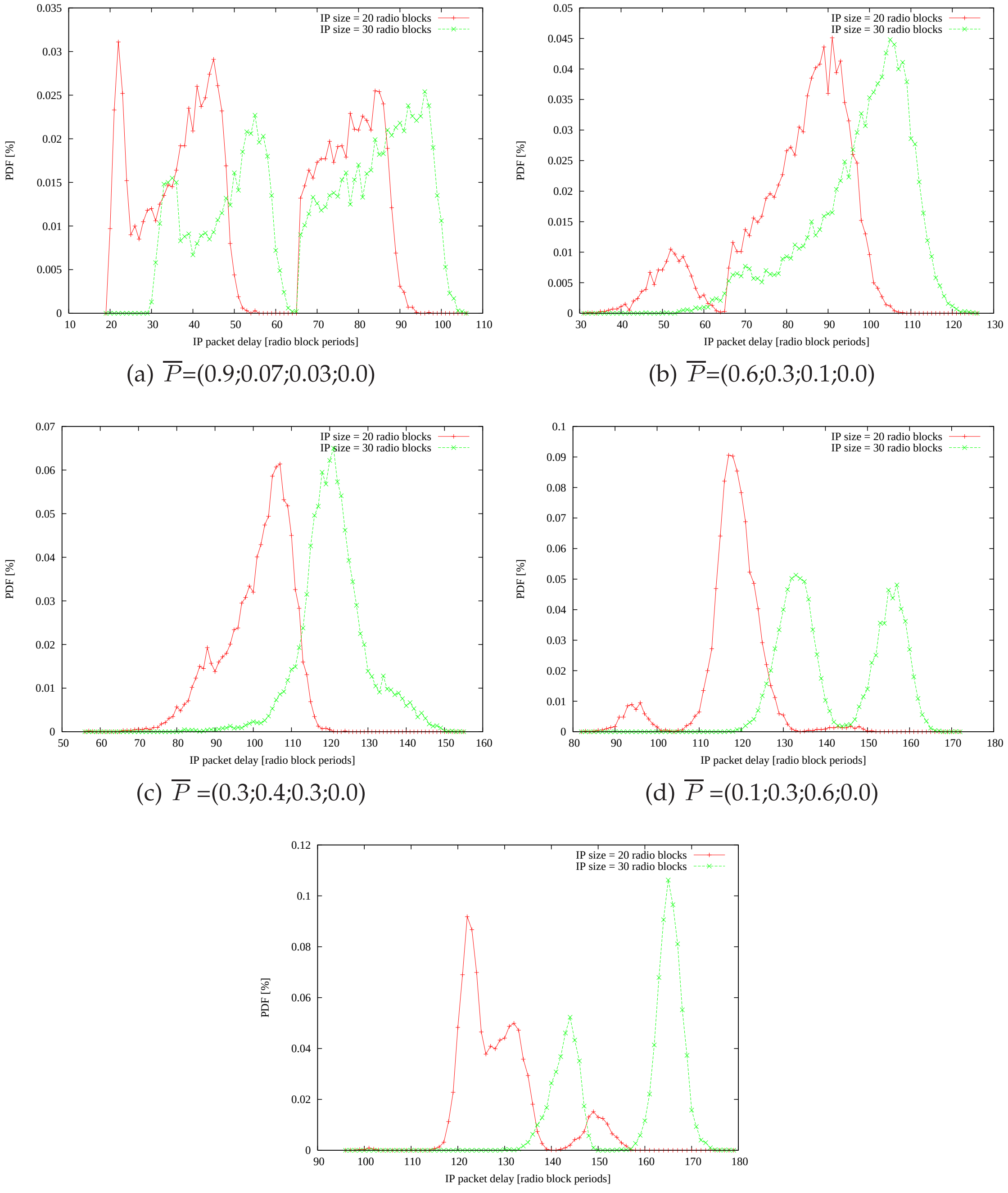
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.13: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$*.*



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

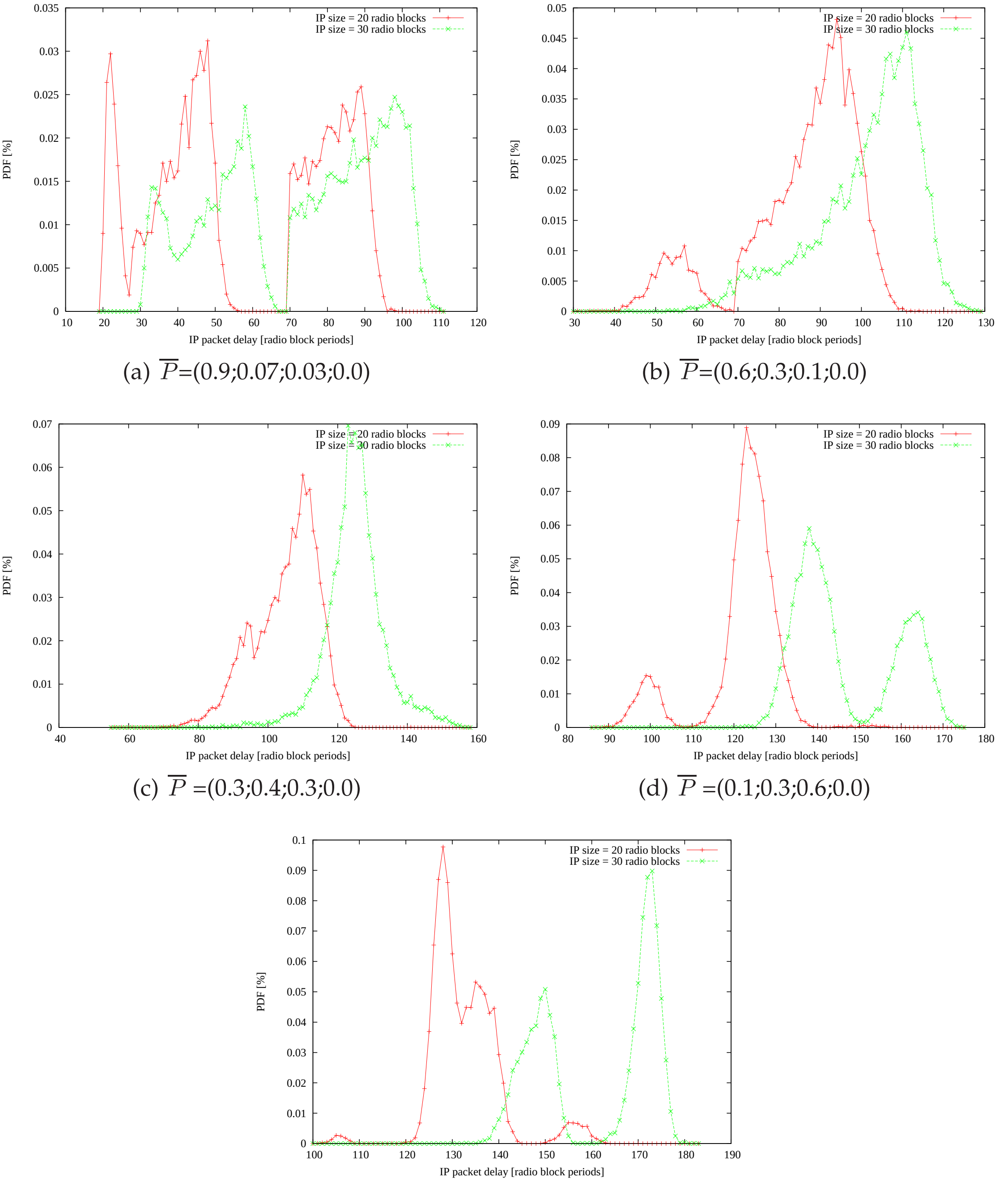
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.14: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure E.15: *Distribution of IP packet delay for two different large packet sizes, twenty and thirty radio blocks - simulation results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

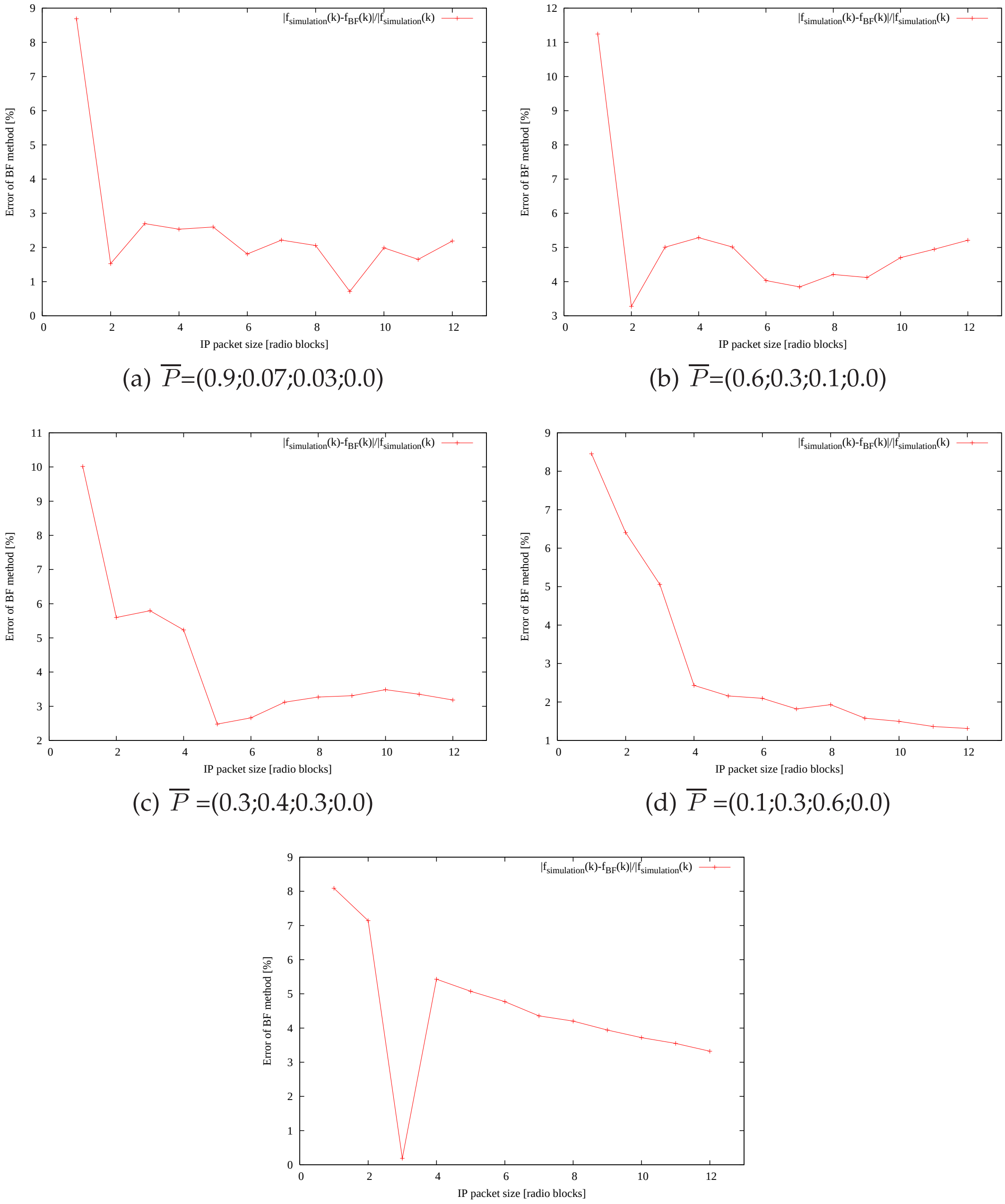
APPENDIX F

Error of the Brute Force method

This appendix presents the error introduced by the BF method for a variety of ARQ delay vectors. The simulation results are compared to relevant results obtained from the BF algorithm. The error is calculated according to the formula below.

$$error_{BF}(k) = \left| \frac{f_{sim}(k) - f_{BF}(k)}{f_{sim}(k)} \right| \cdot 100\% \tag{F.1}$$

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

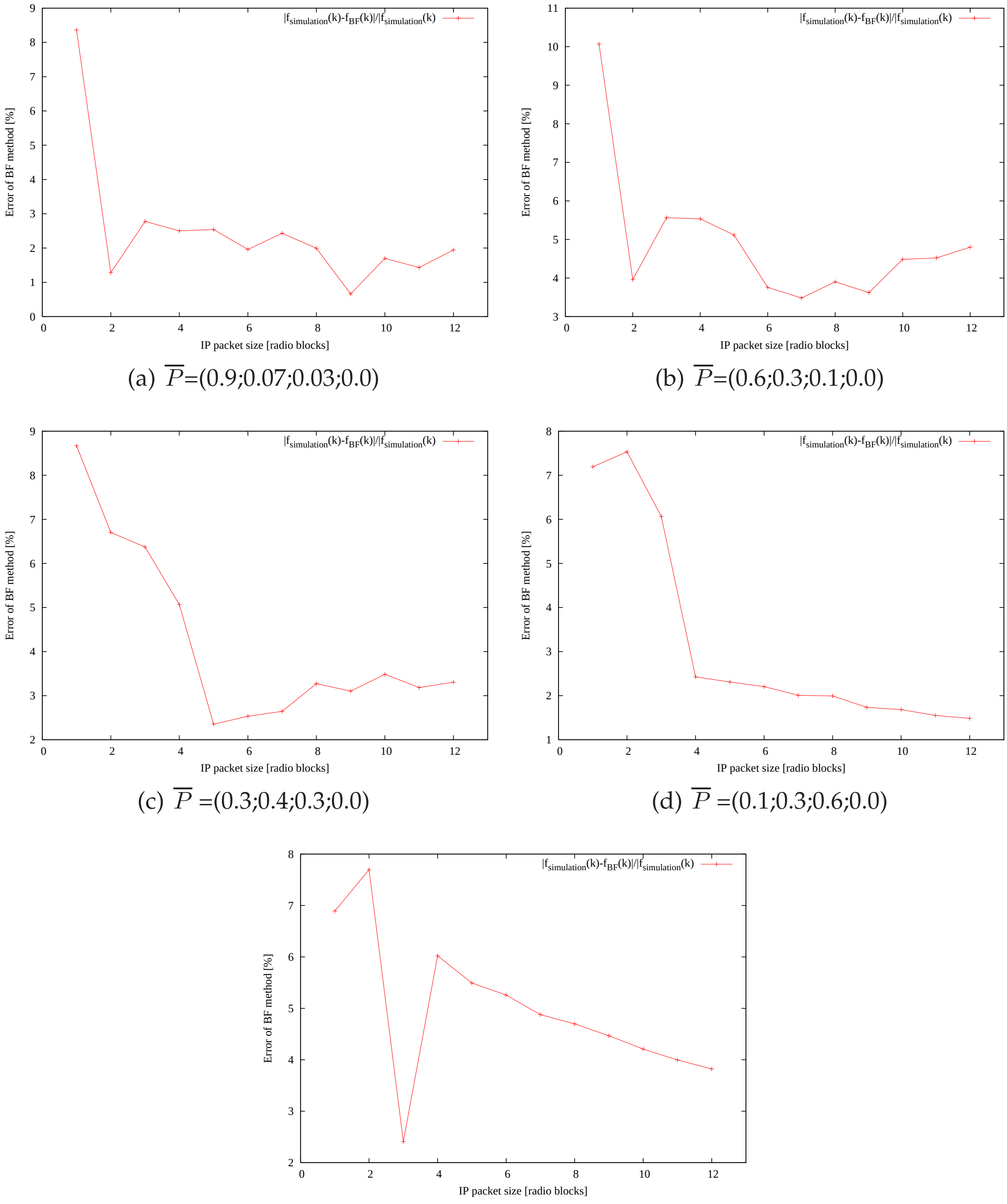
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.1: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

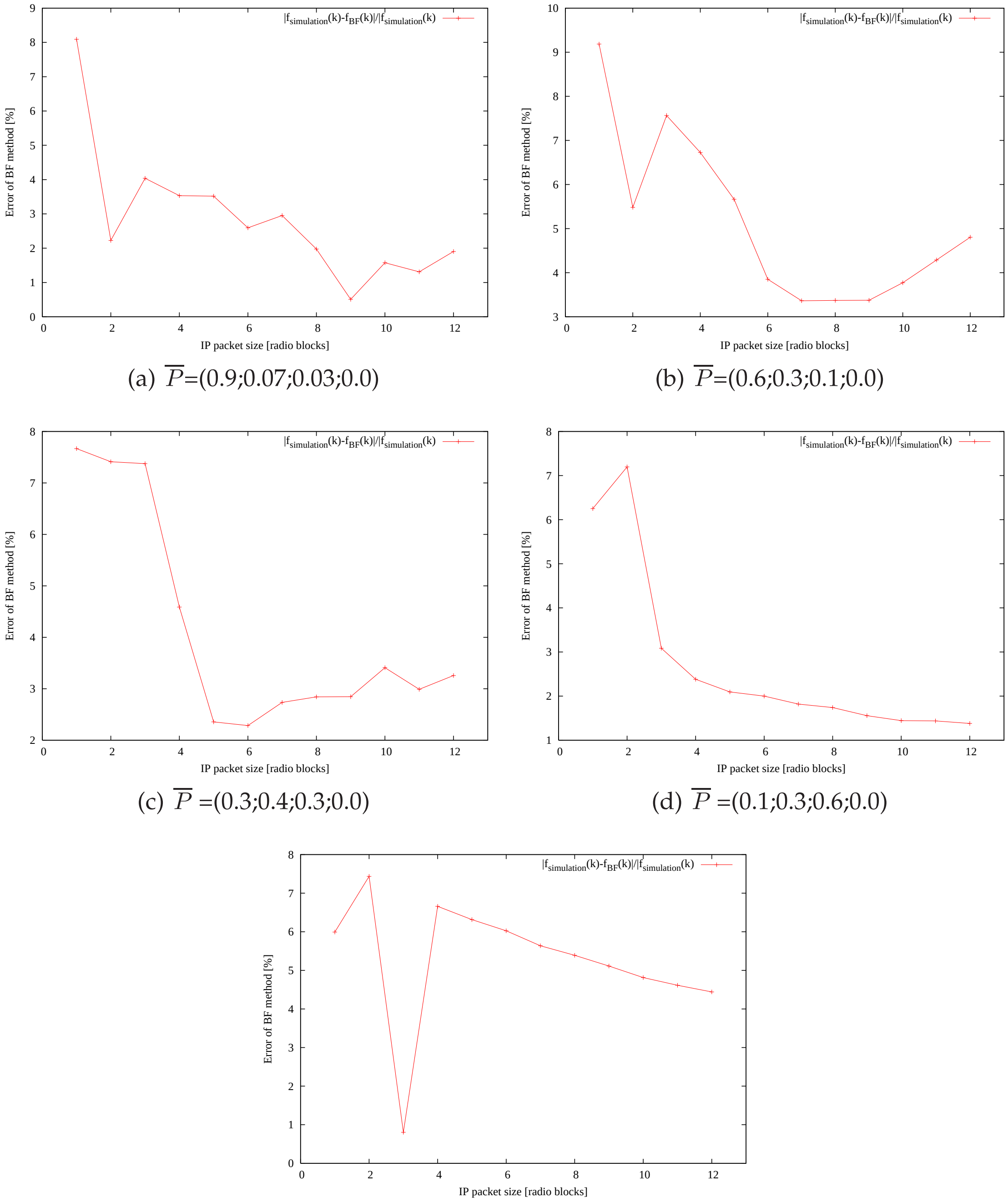
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.2: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

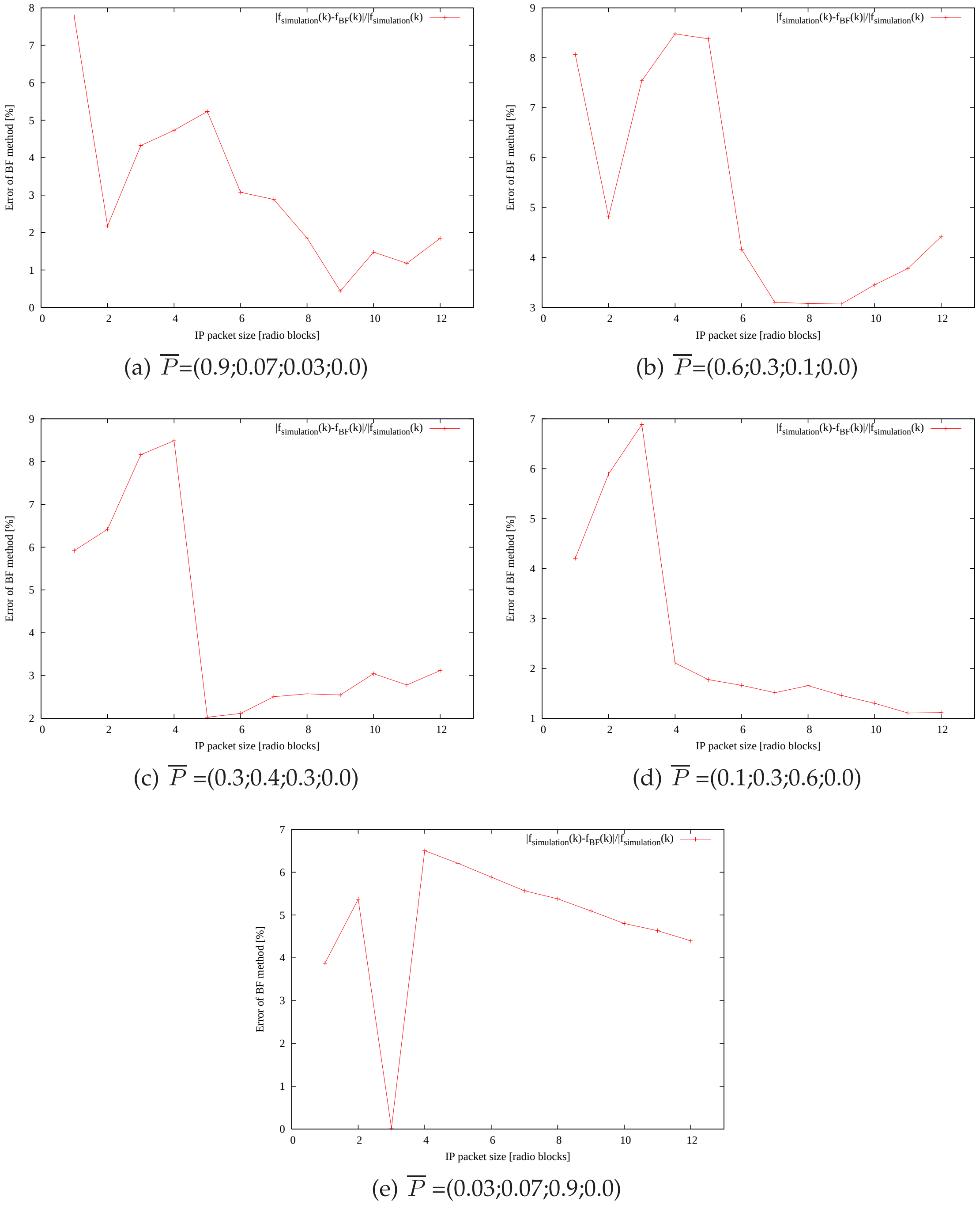
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.3: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

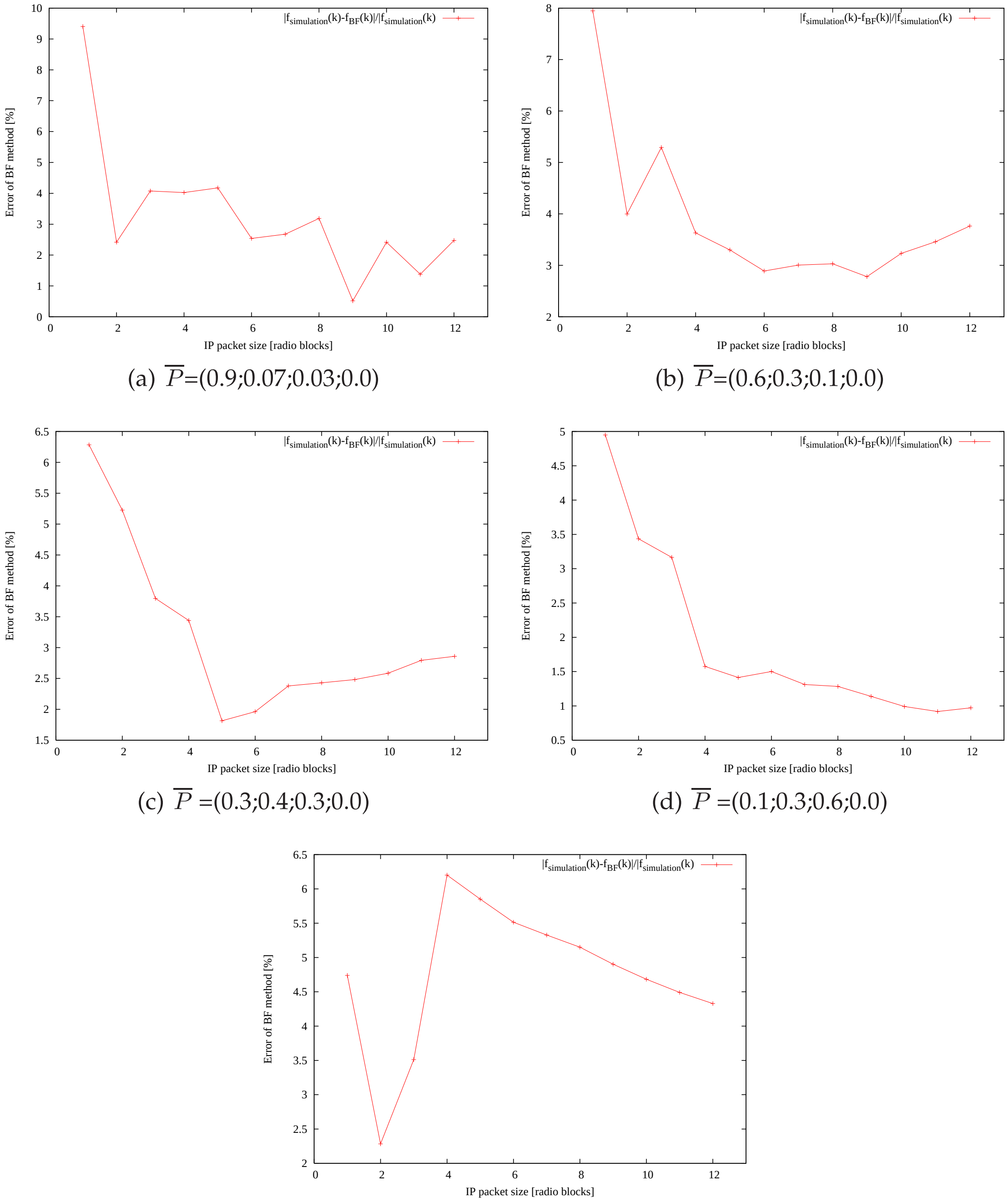
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.4: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

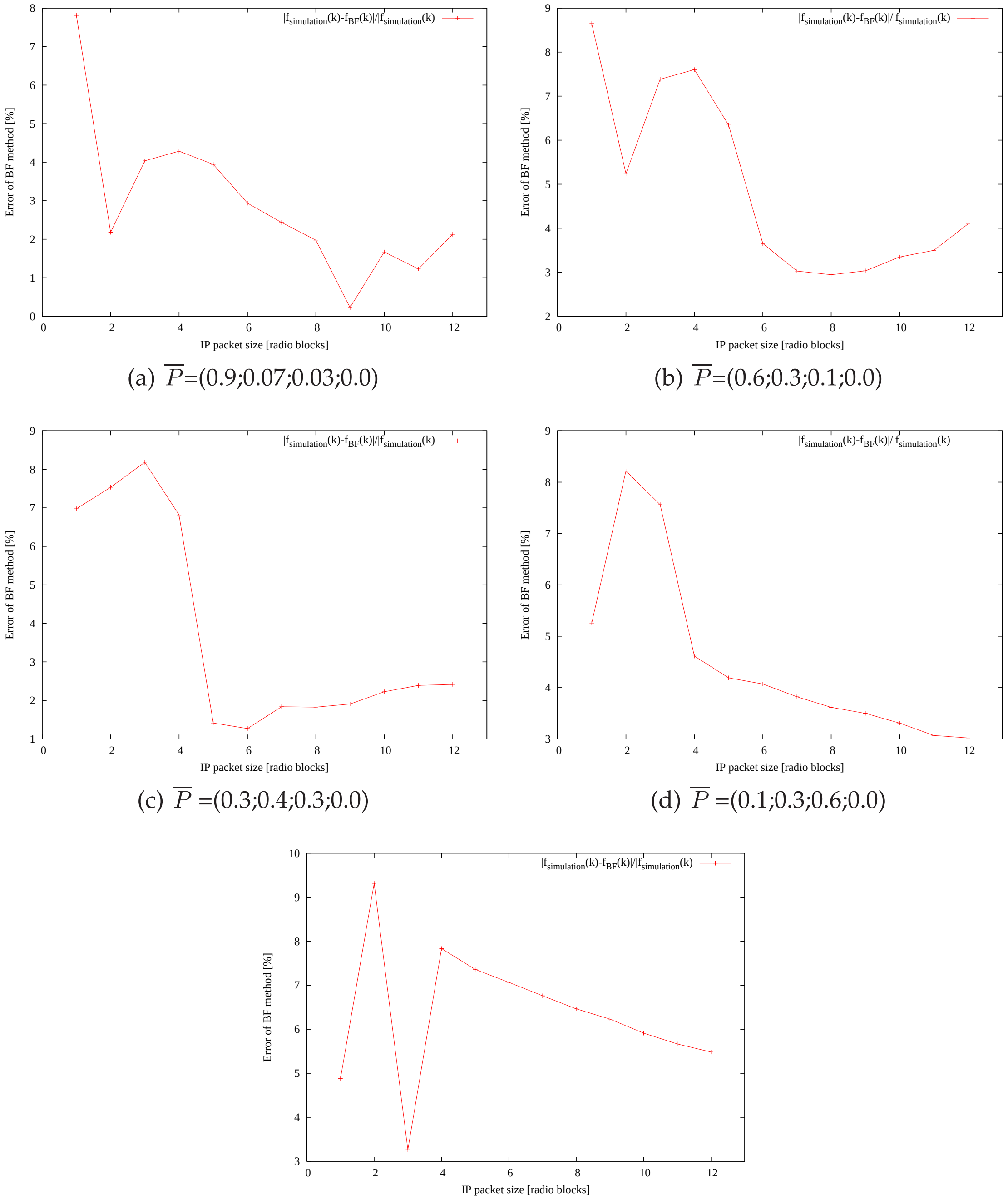
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.5: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

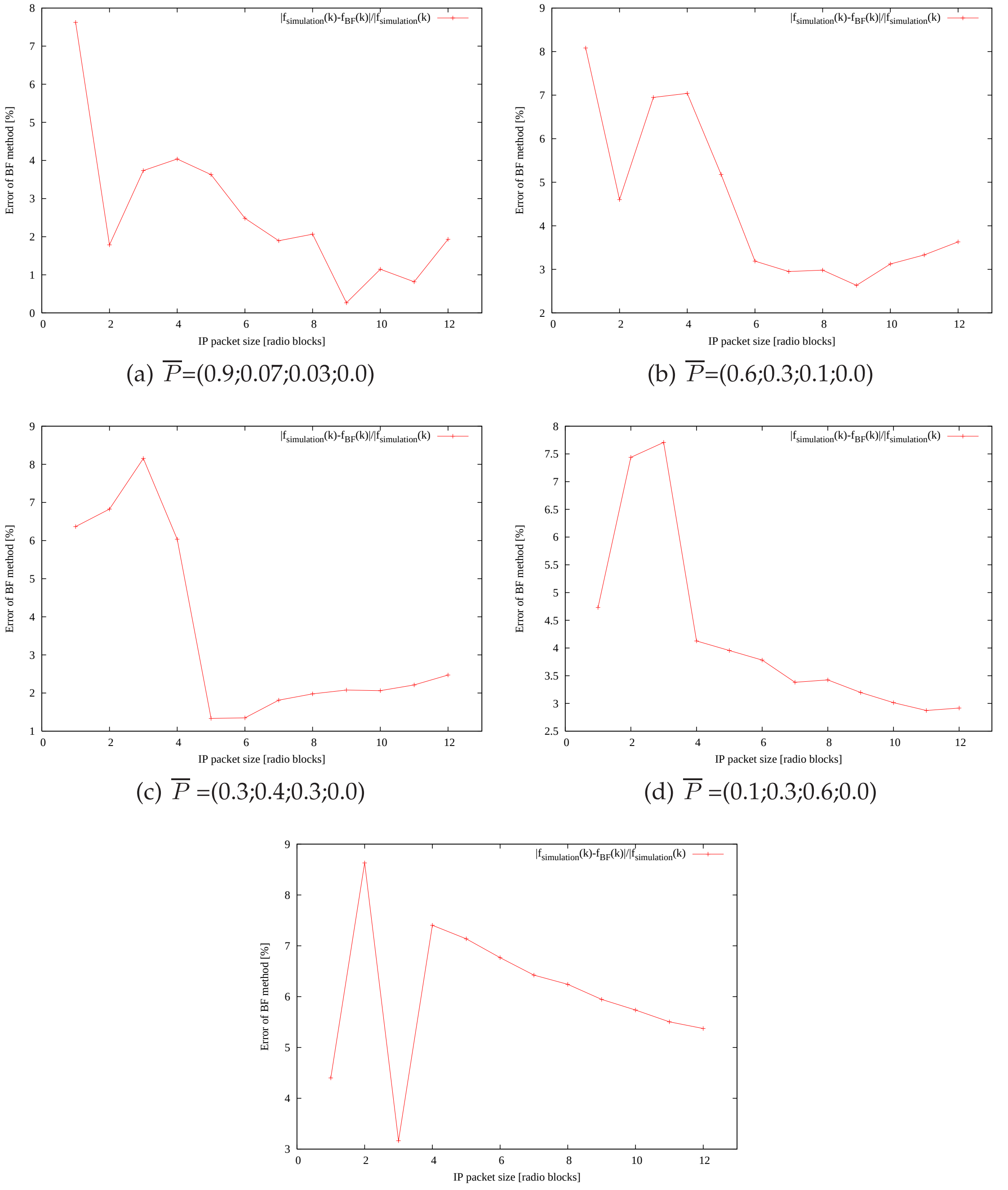
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.6: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

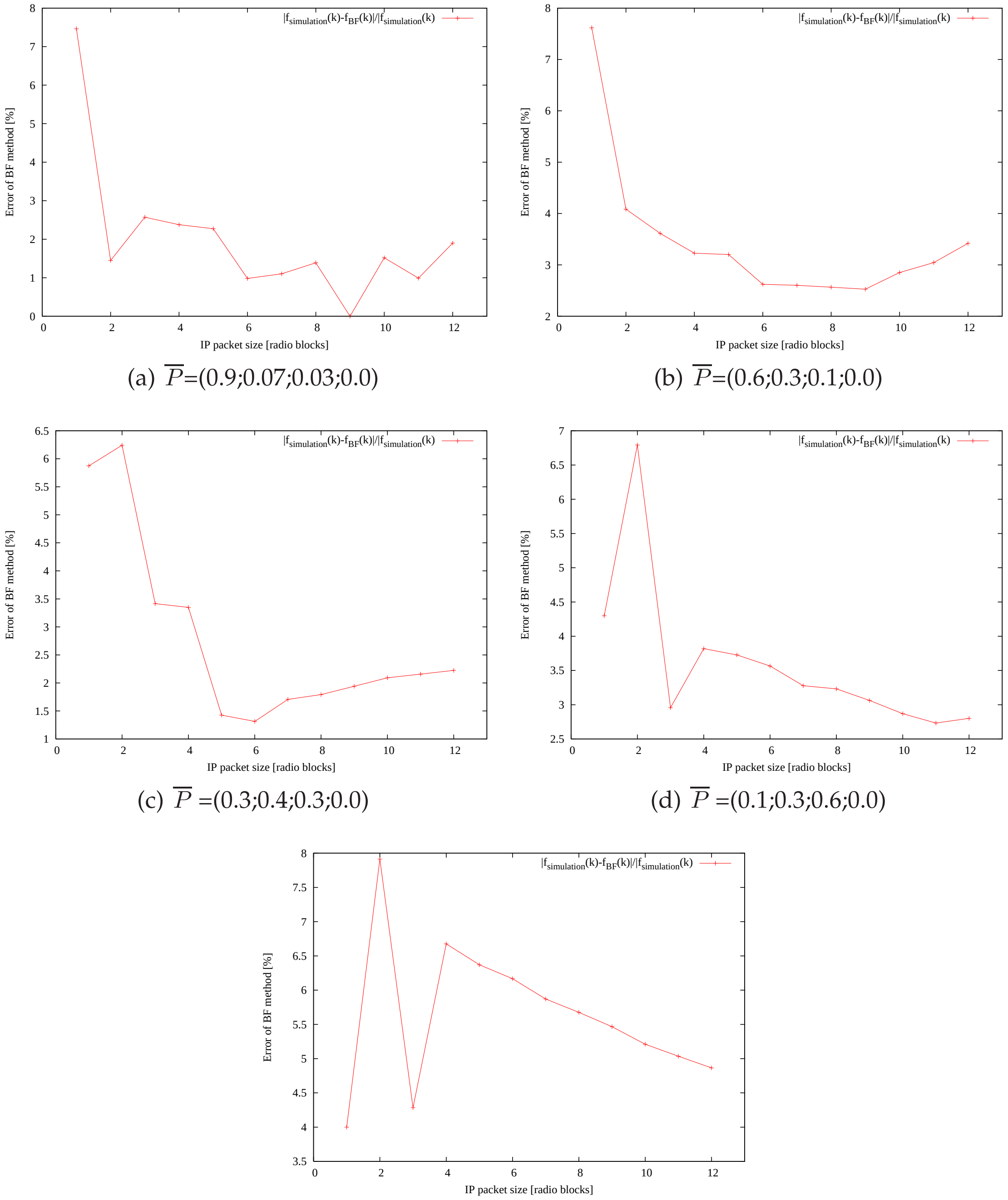
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.7: *Error of Brute Force method - results for* $\overline{\Delta} = [0;15;25;0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

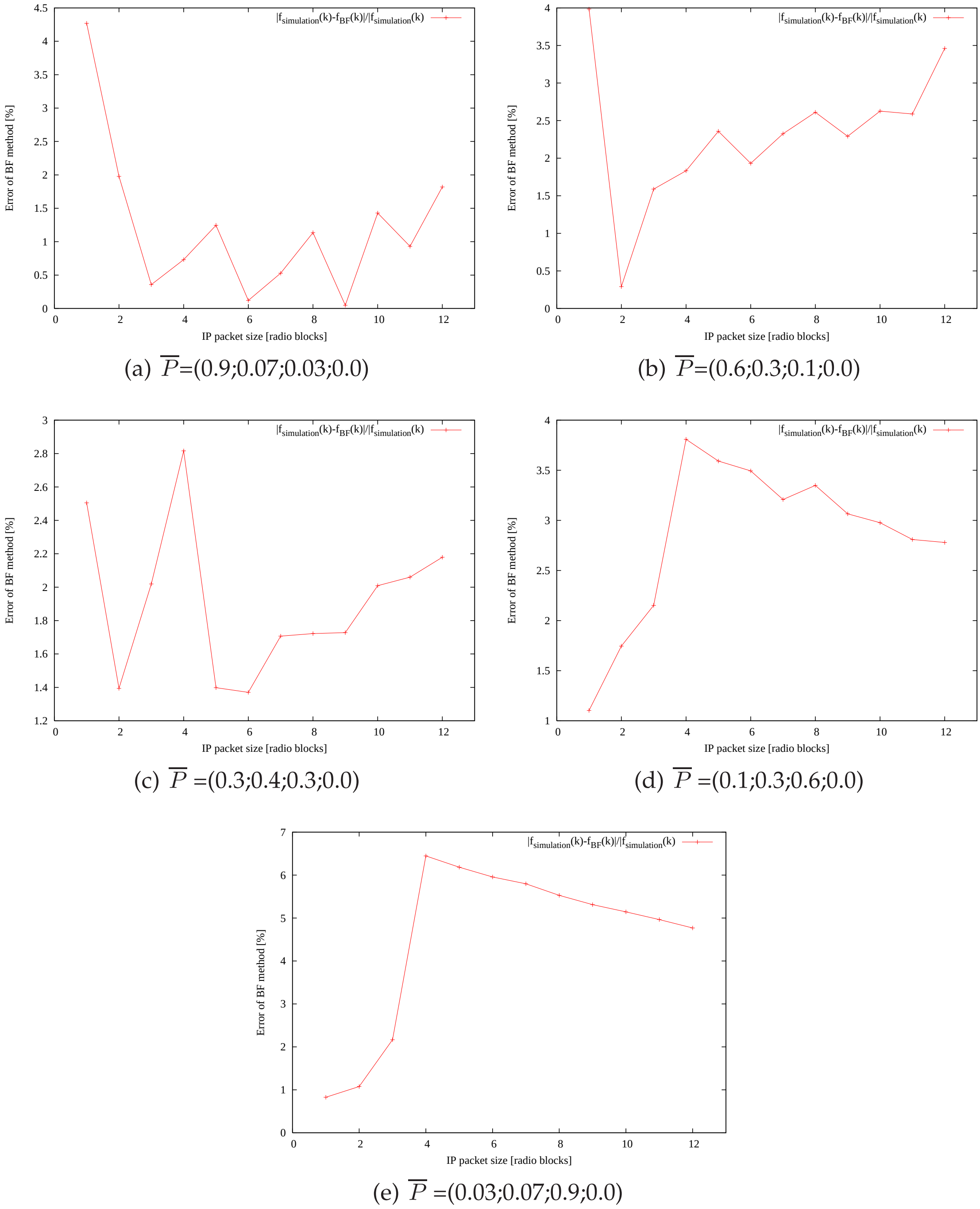
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.8: *Error of Brute Force method - results for* $\overline{\Delta} = [0;17;27;0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

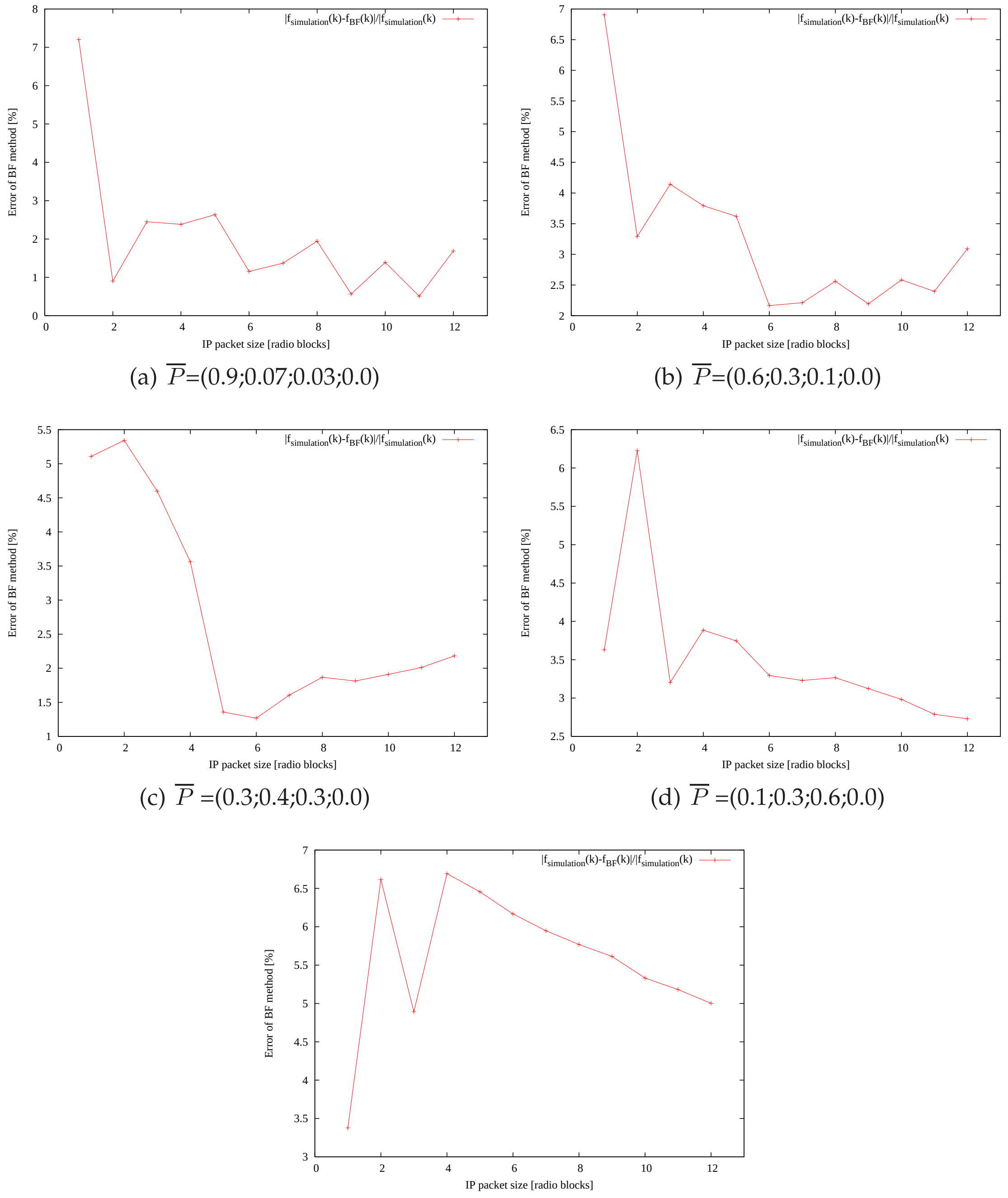
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.9: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

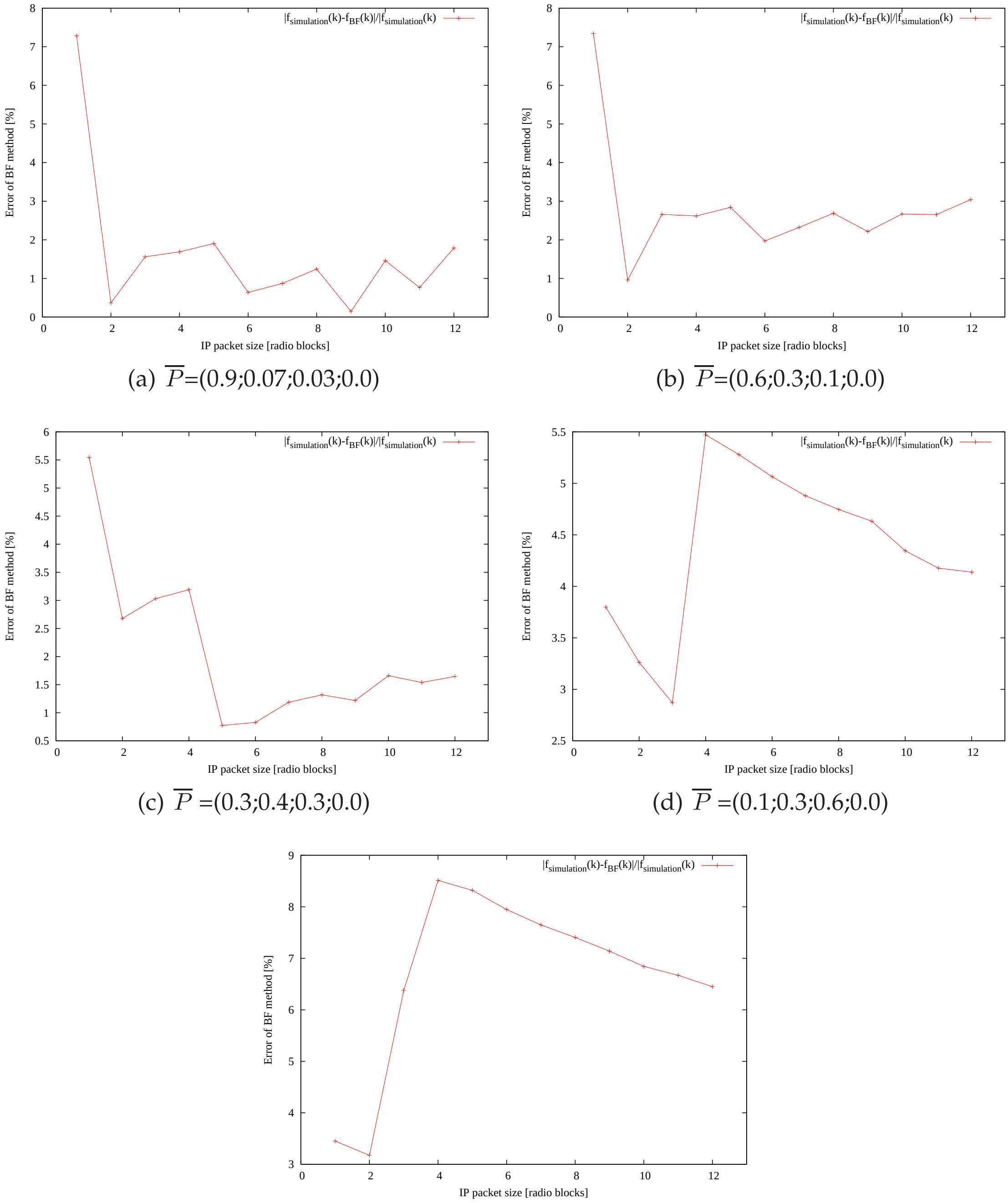
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.10: *Error of Brute Force method - results for* $\overline{\Delta} = [0;21;31;0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

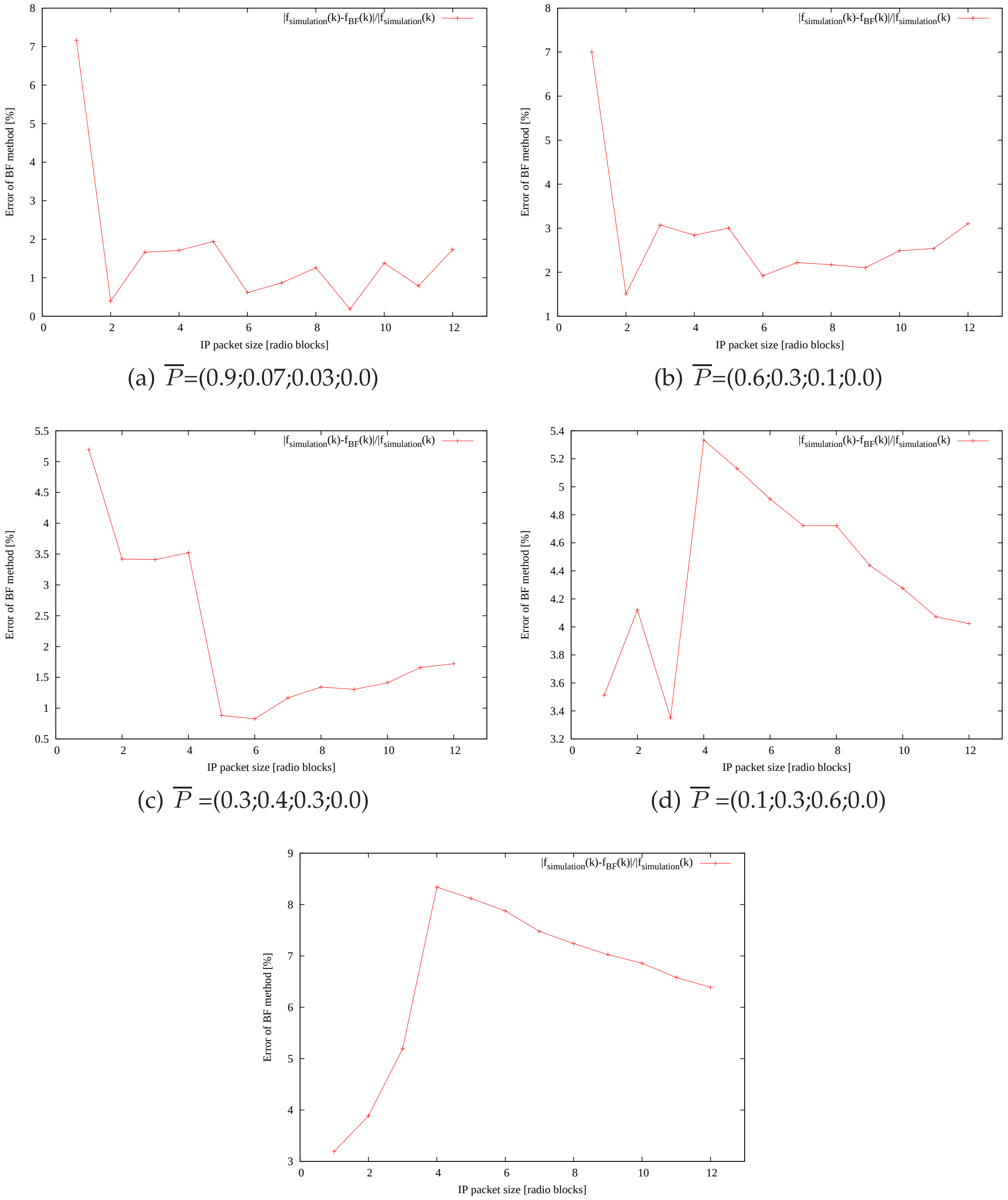
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.11: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

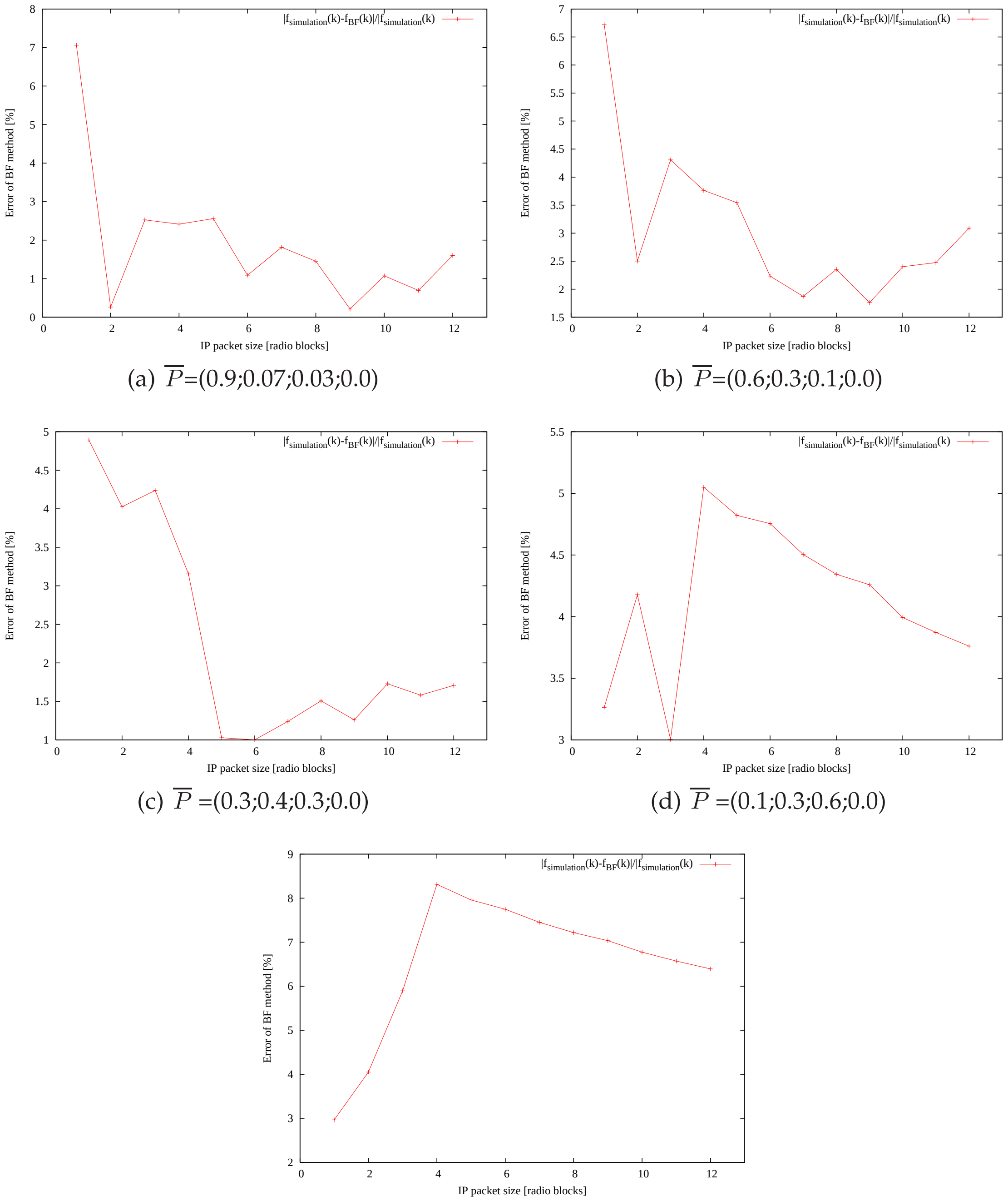
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.12: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

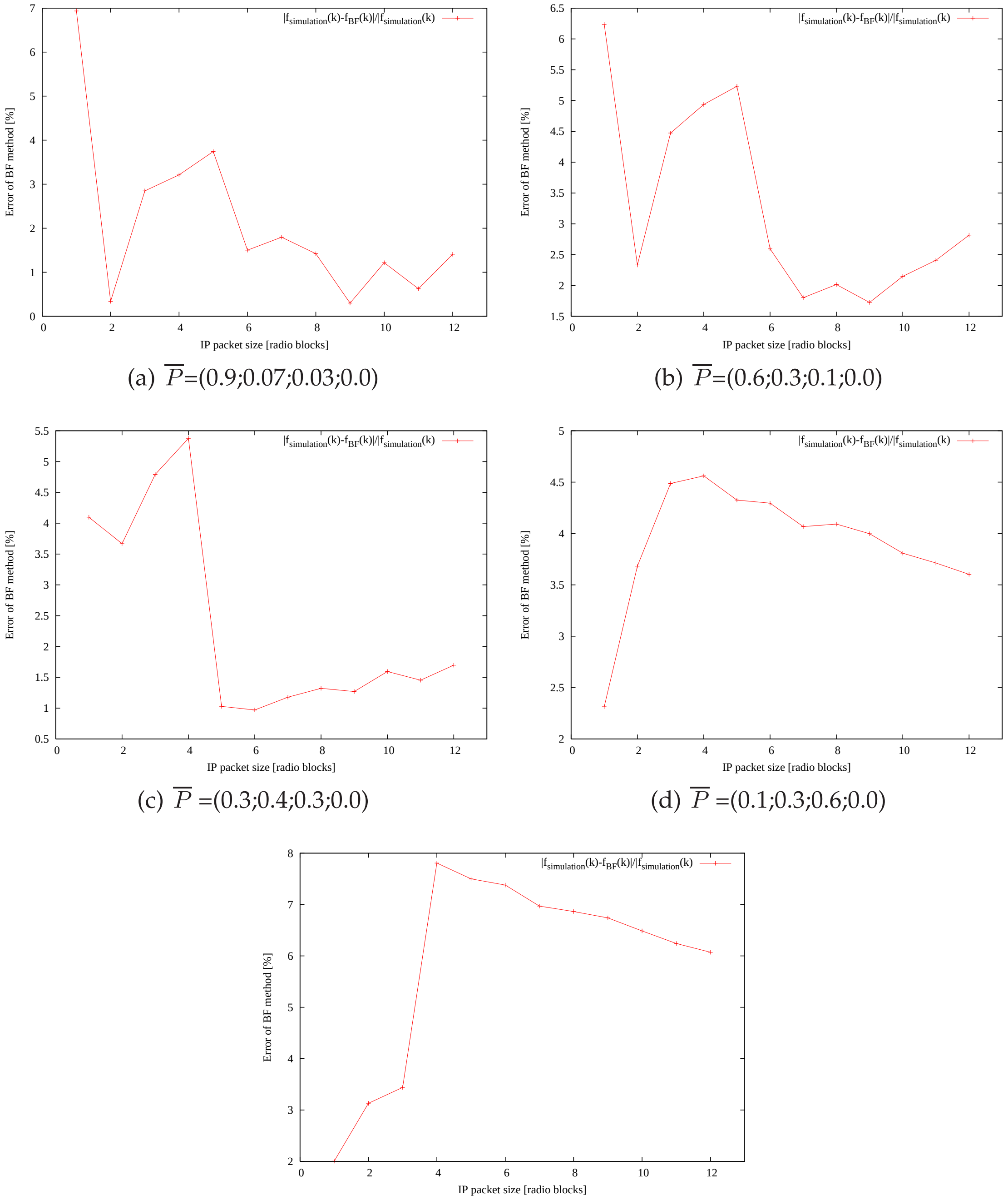
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.13: *Error of Brute Force method - results for* $\overline{\Delta} = [0;22;37;0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

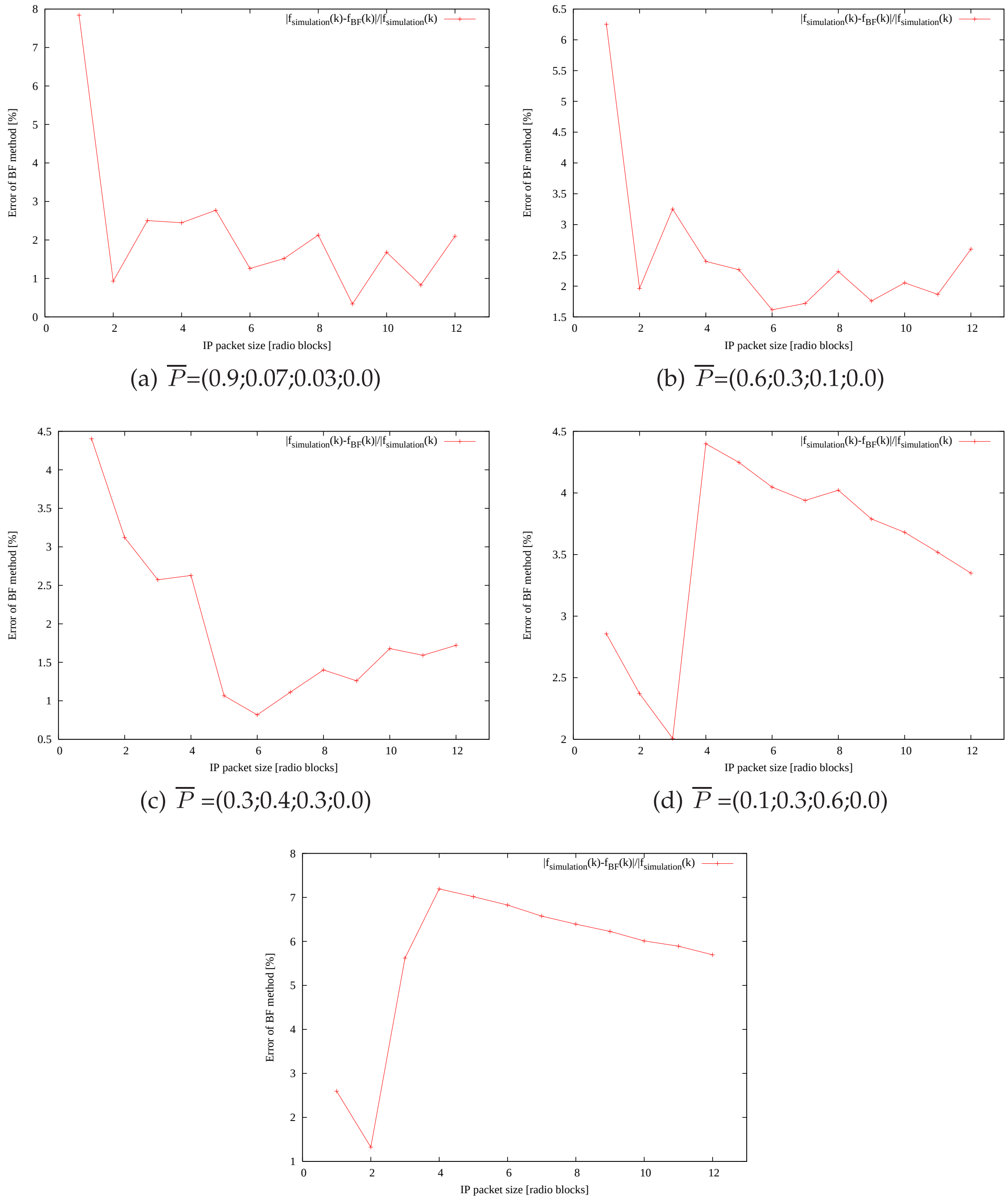
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.14: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure F.15: *Error of Brute Force method - results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

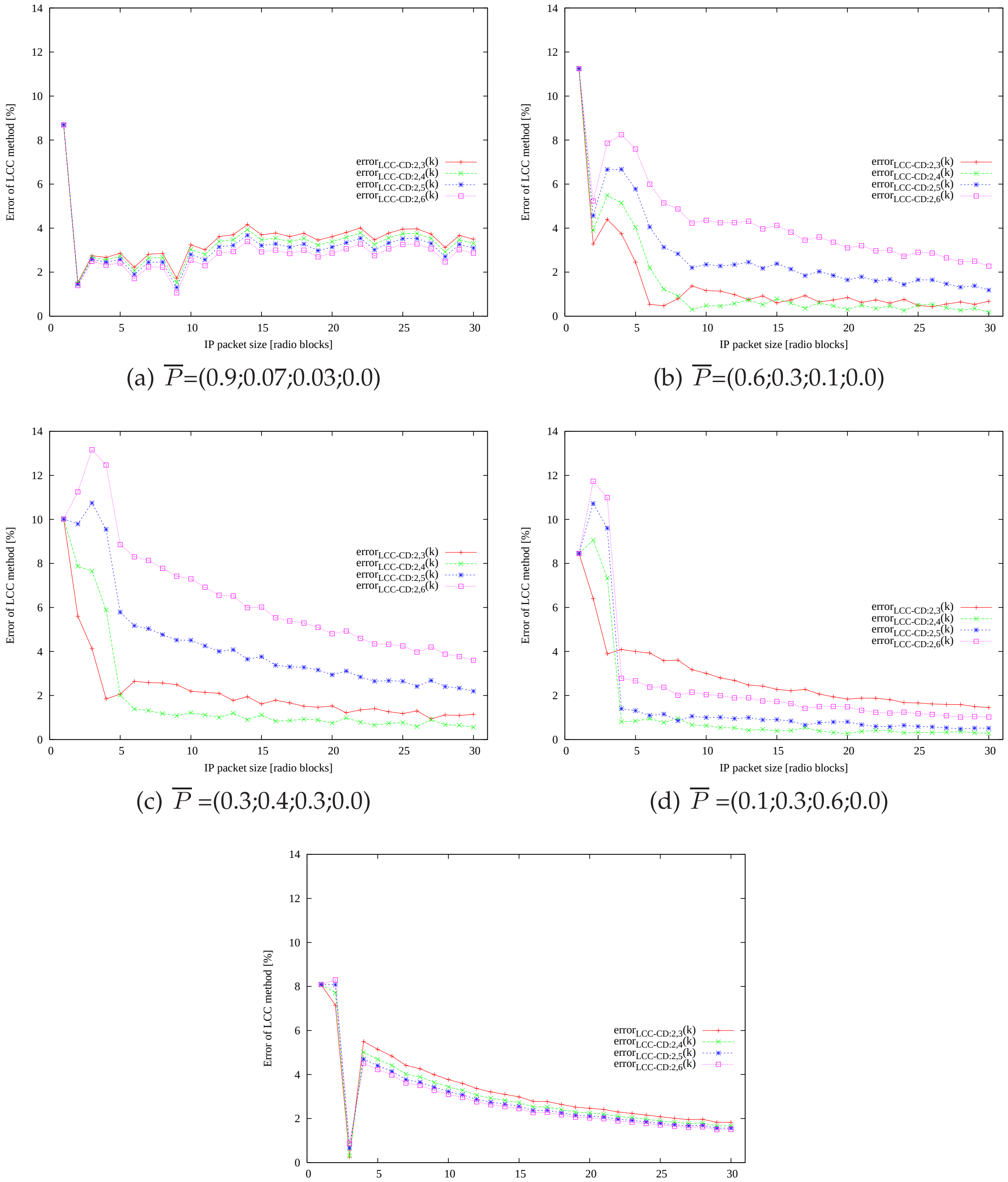
APPENDIX G

Error of Low Computation Complexity method

This appendix presents the comparison of simulation results to those given by the LCC method. The error introduced by the LCC method is calculated according to the formula presented below and is plotted as a function of the IP packet size.

$$error_{LCC-CD:m,n}(k) = \left| \frac{f(k) - f_{CD:m,n}(k)}{f(k)} \right| \cdot 100\% \tag{G.1}$$

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

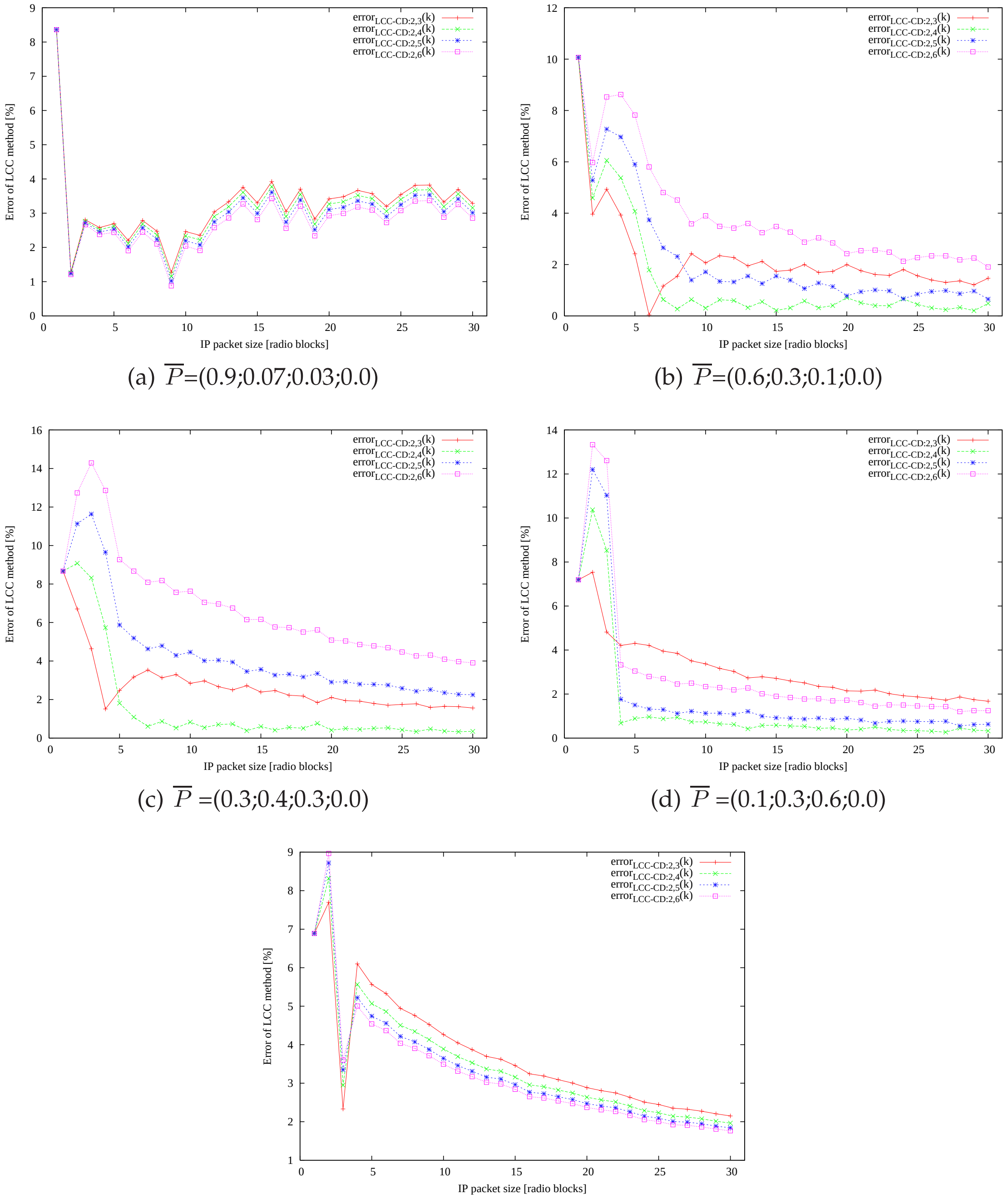
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.1: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

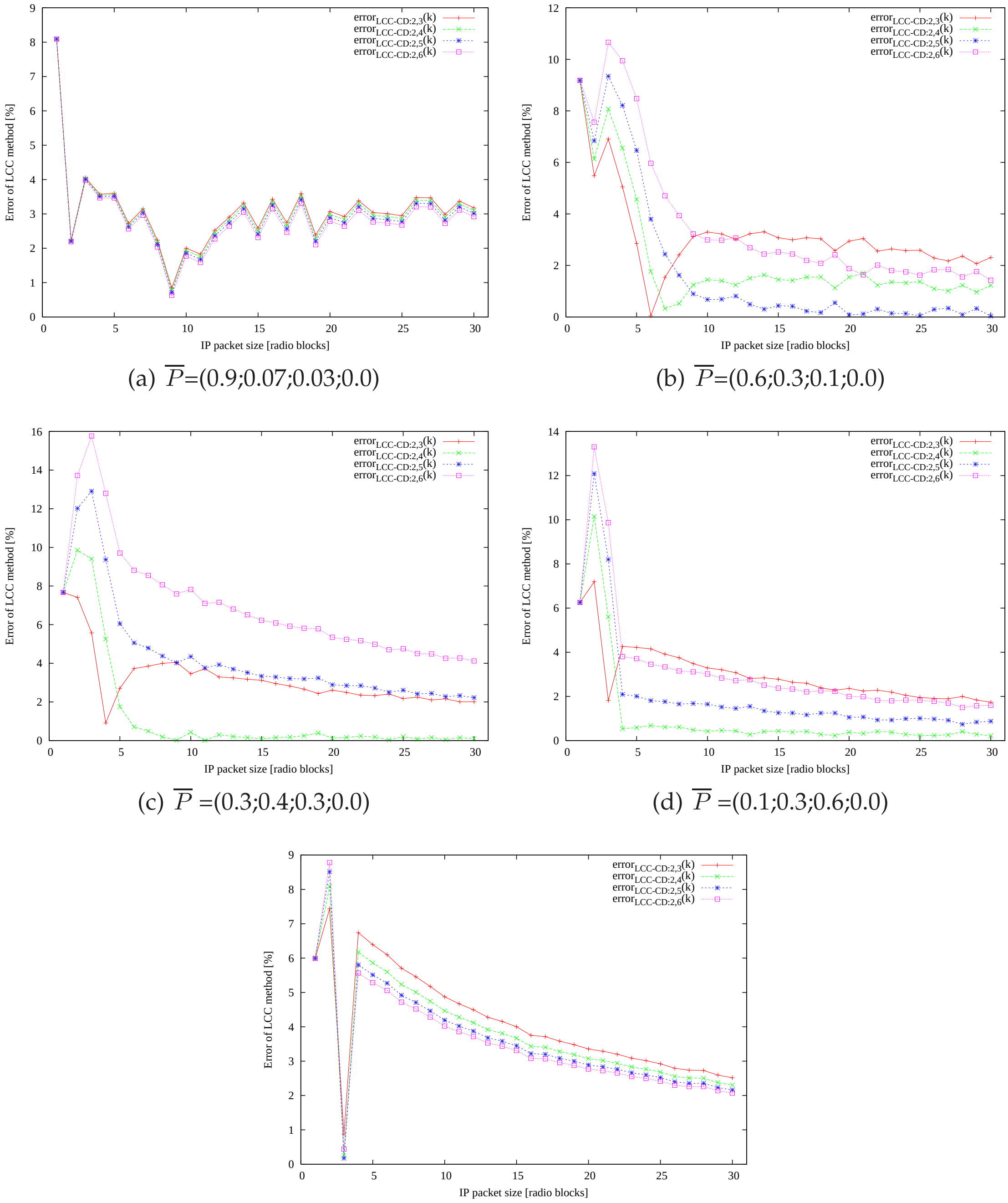
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.2: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

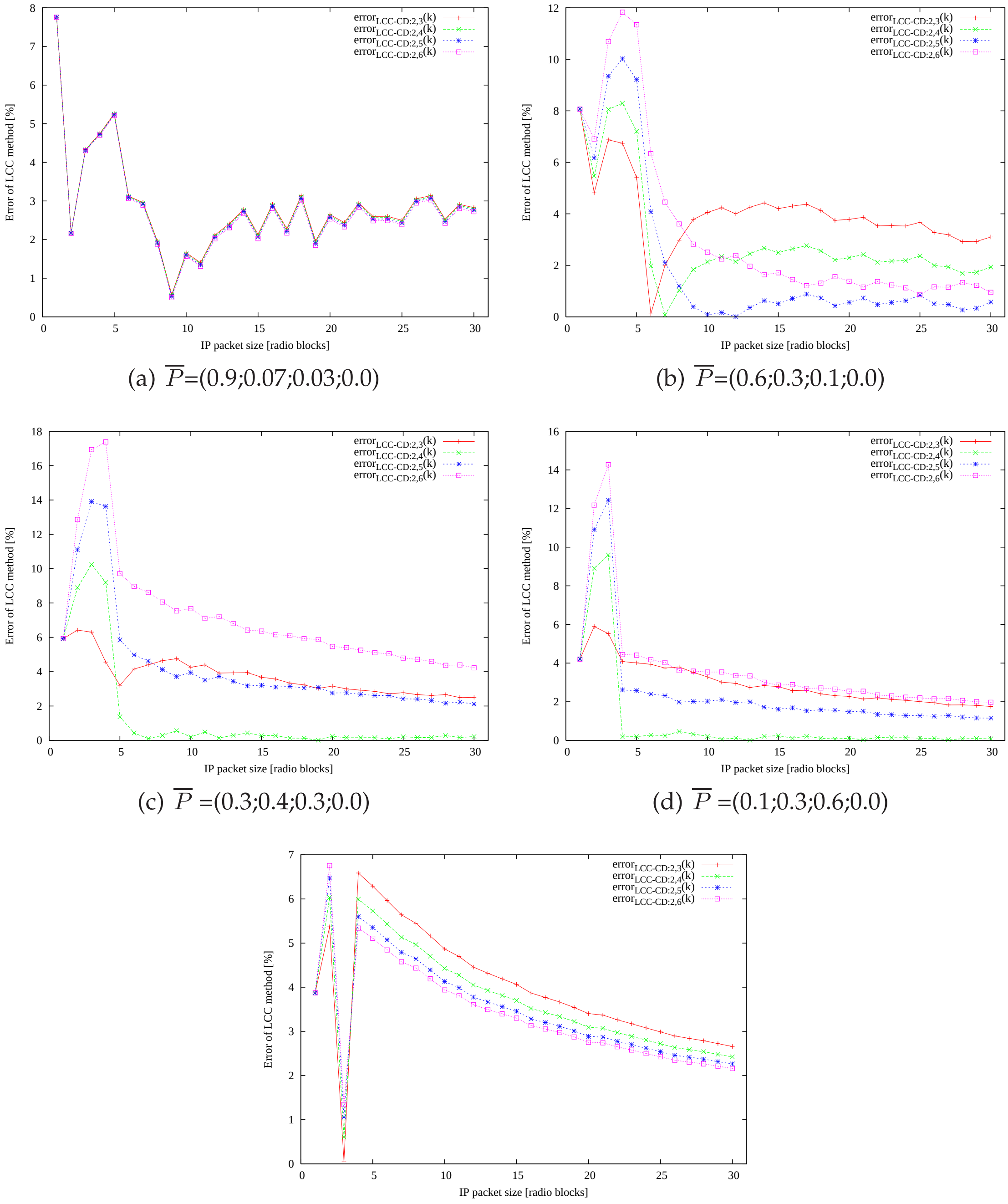
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.3: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

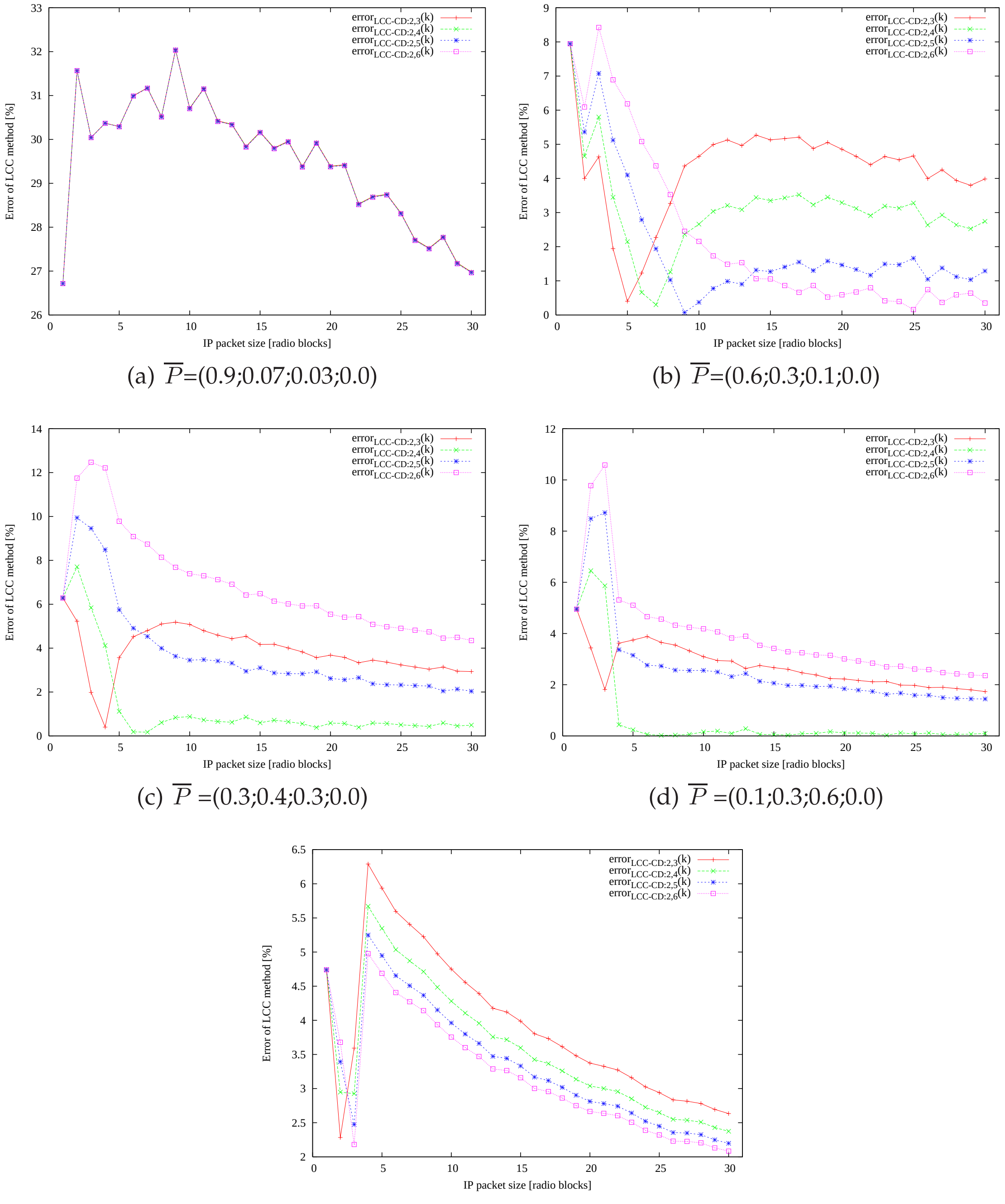
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.4: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

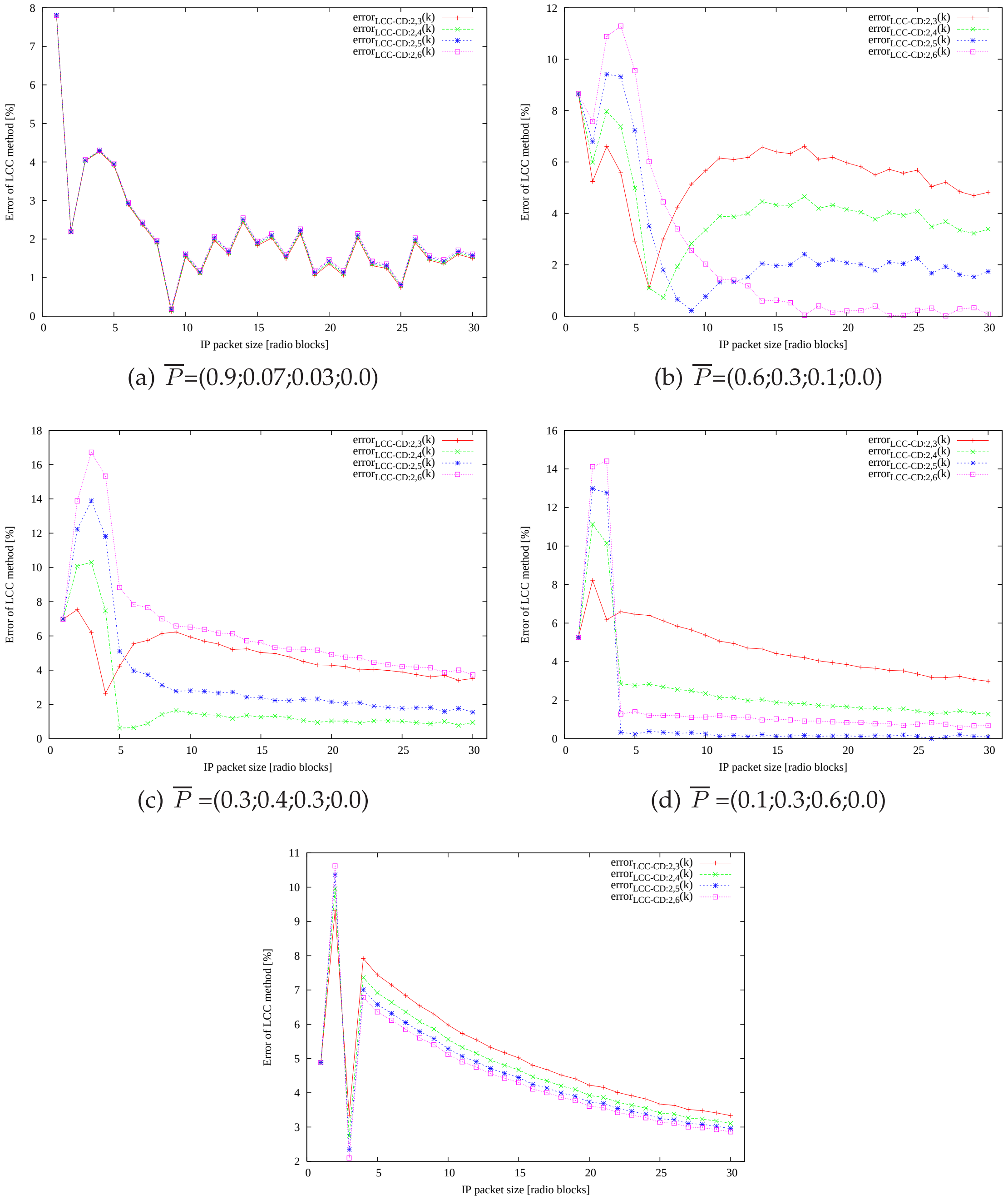
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.5: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

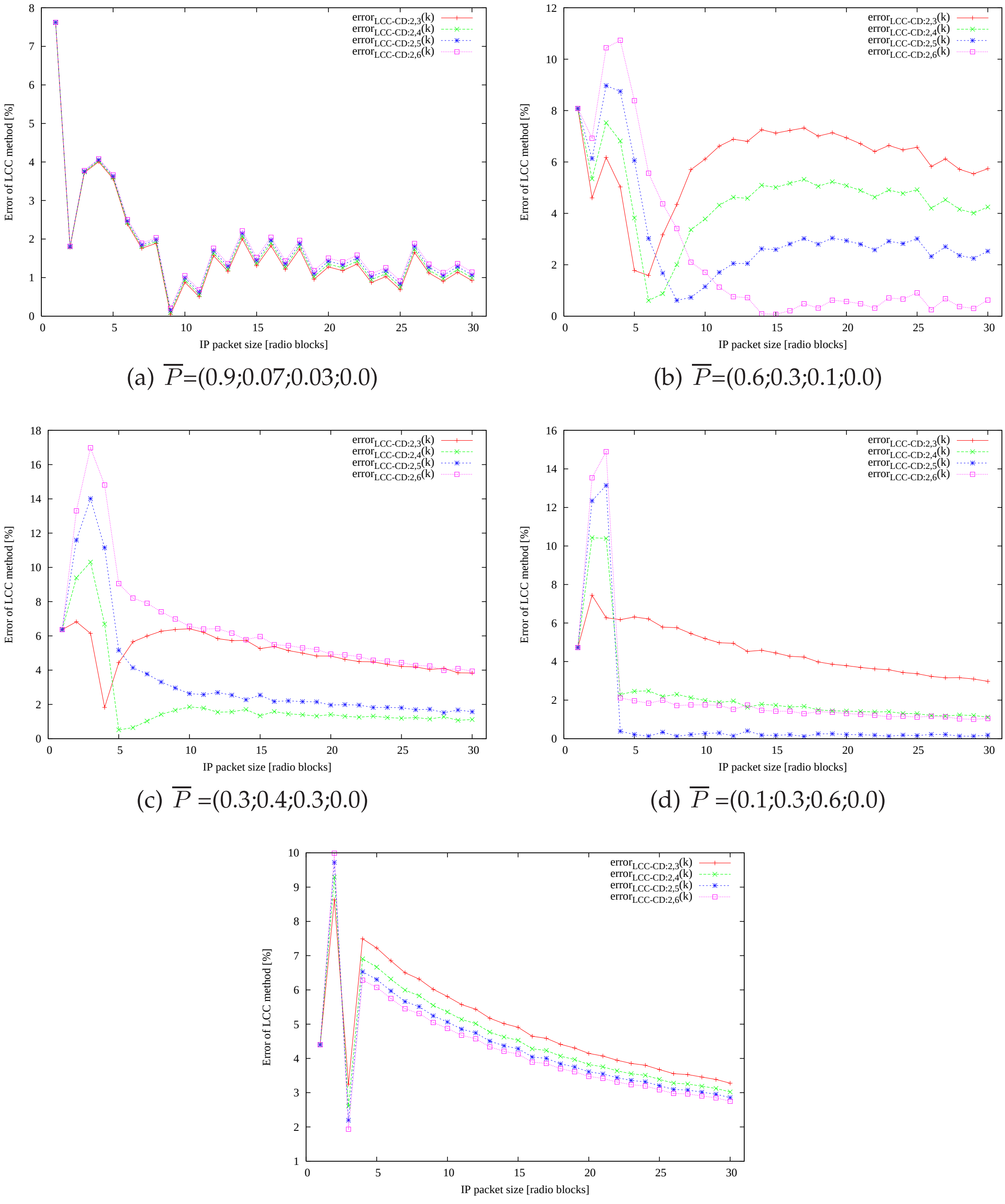
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.6: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

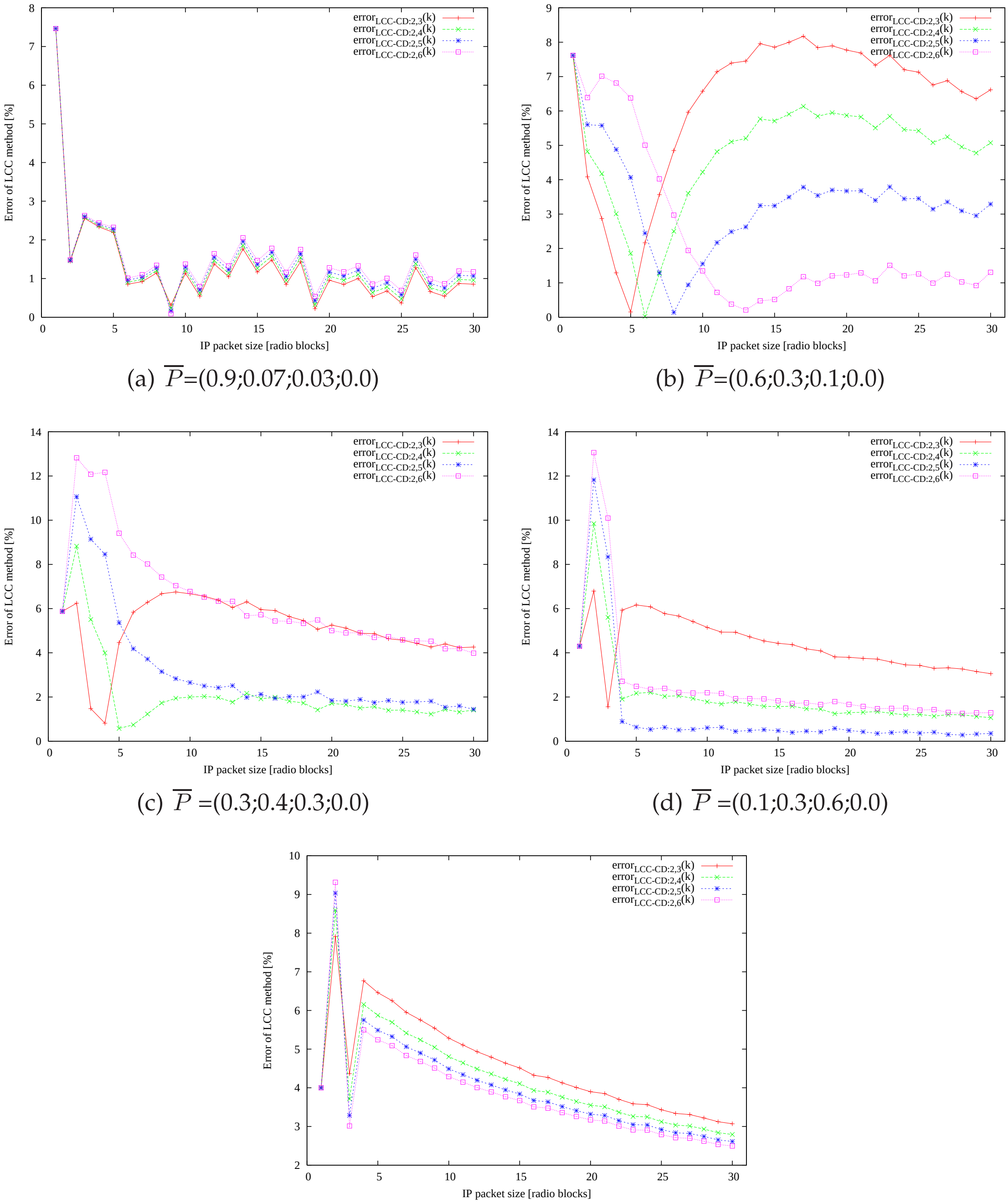
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.7: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

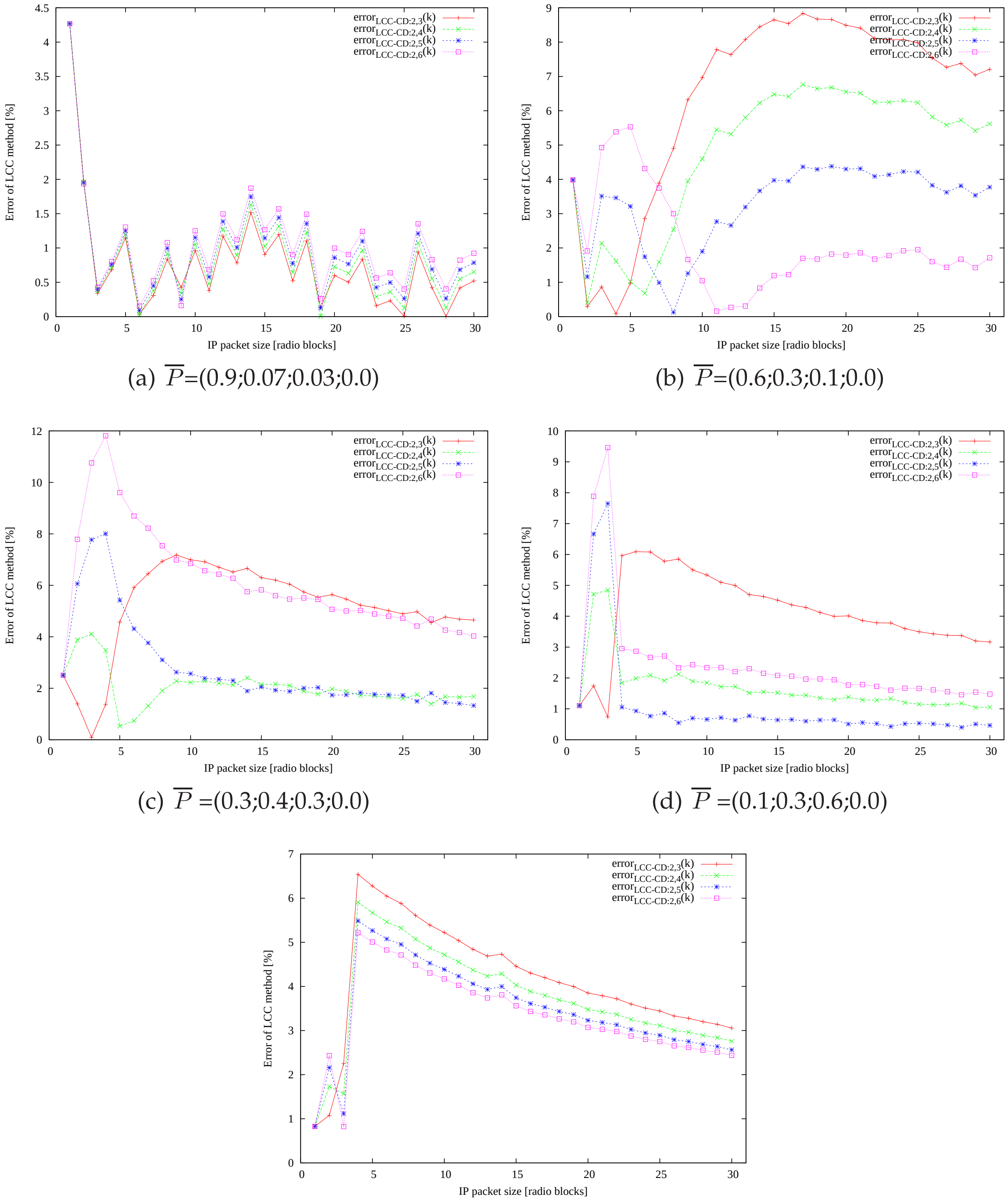
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.8: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

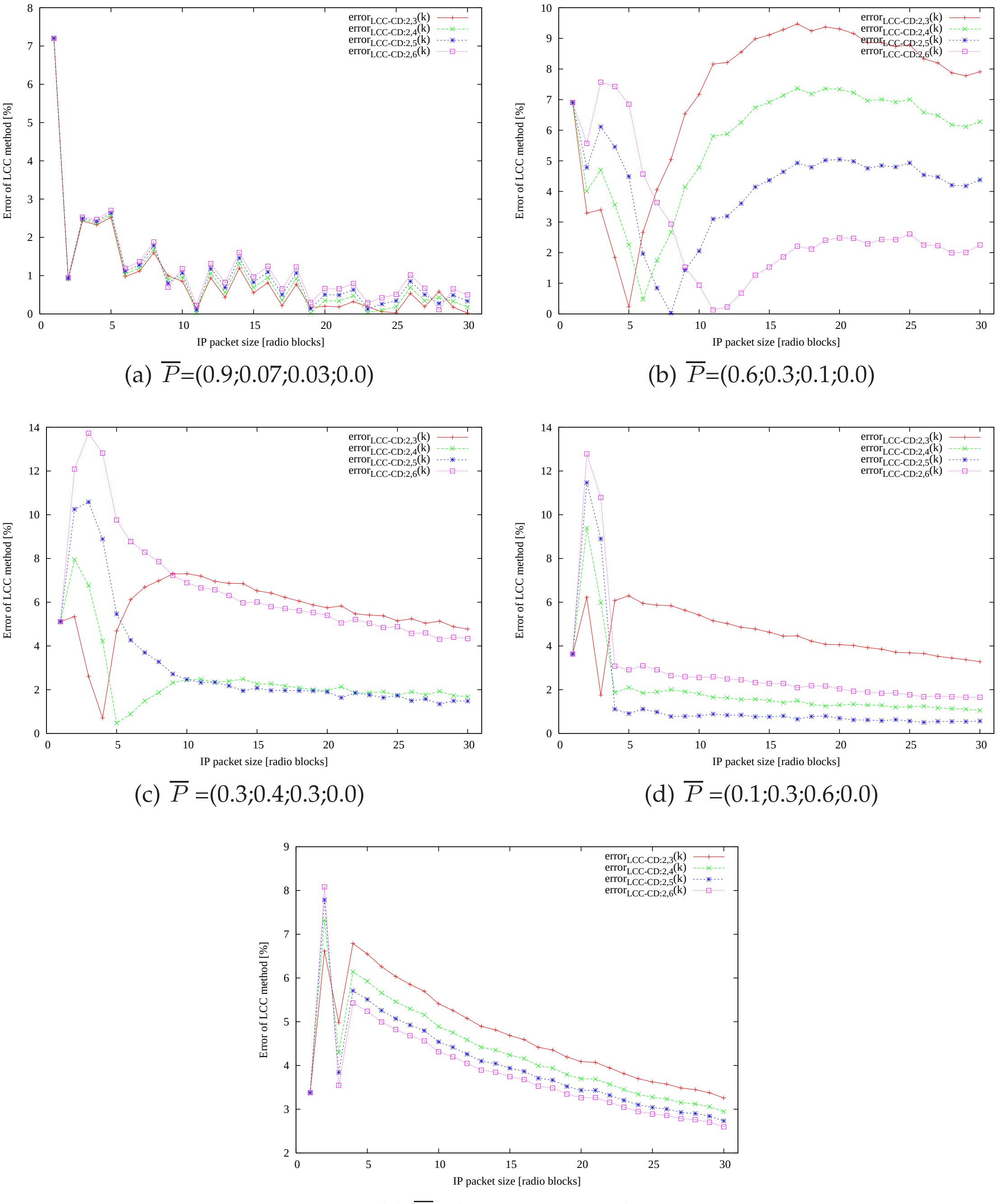
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.9: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

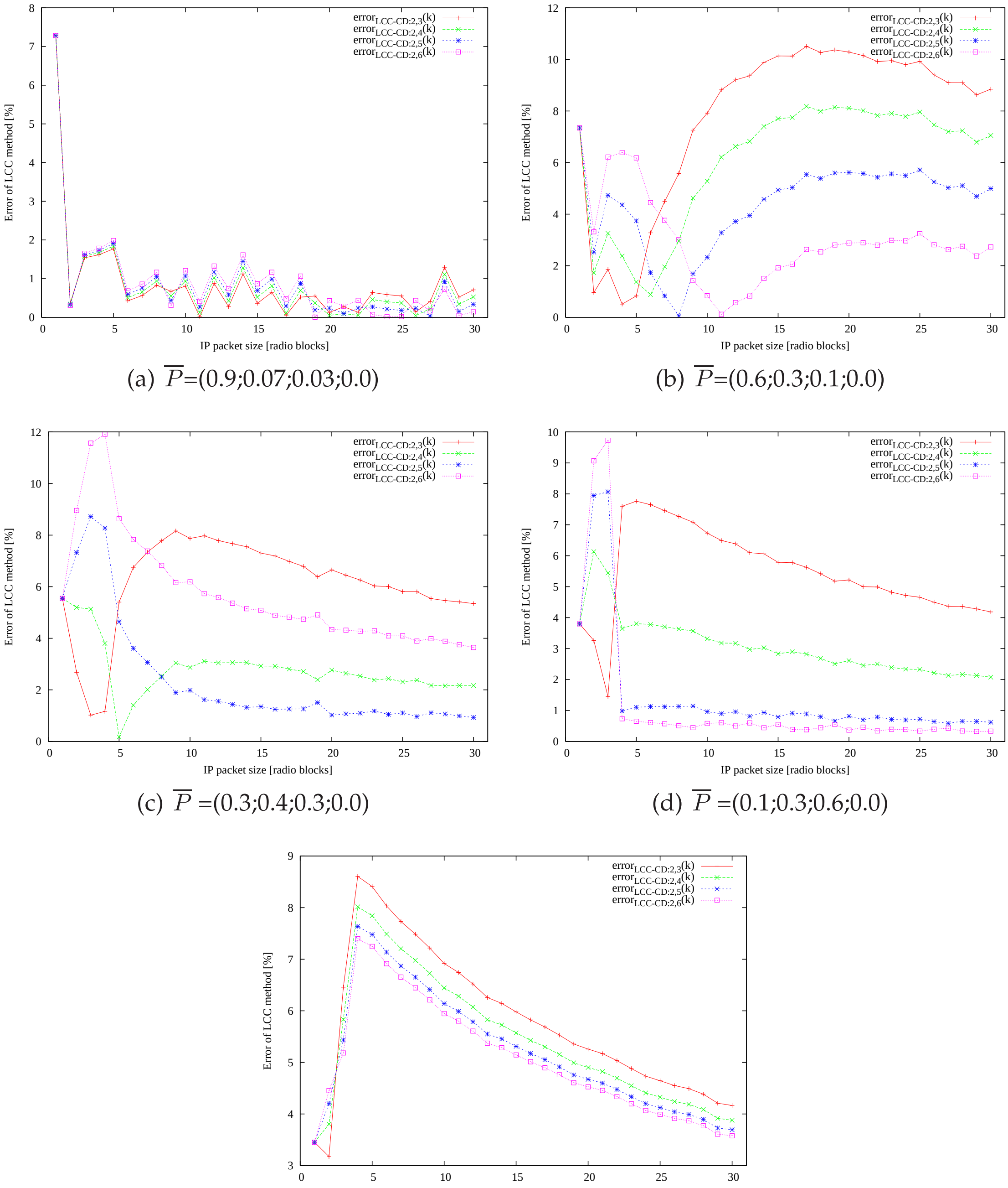
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.10: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

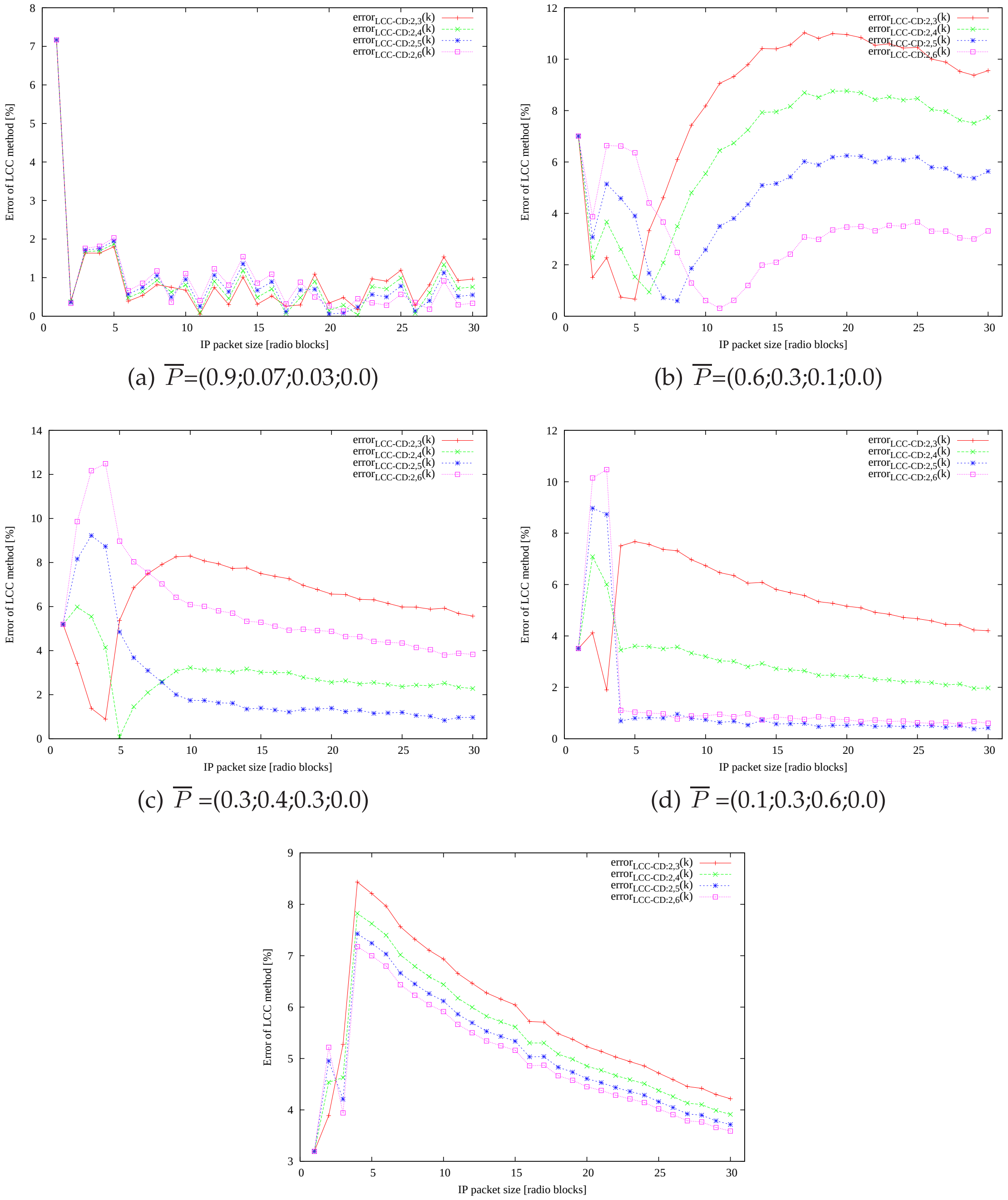
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.11: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

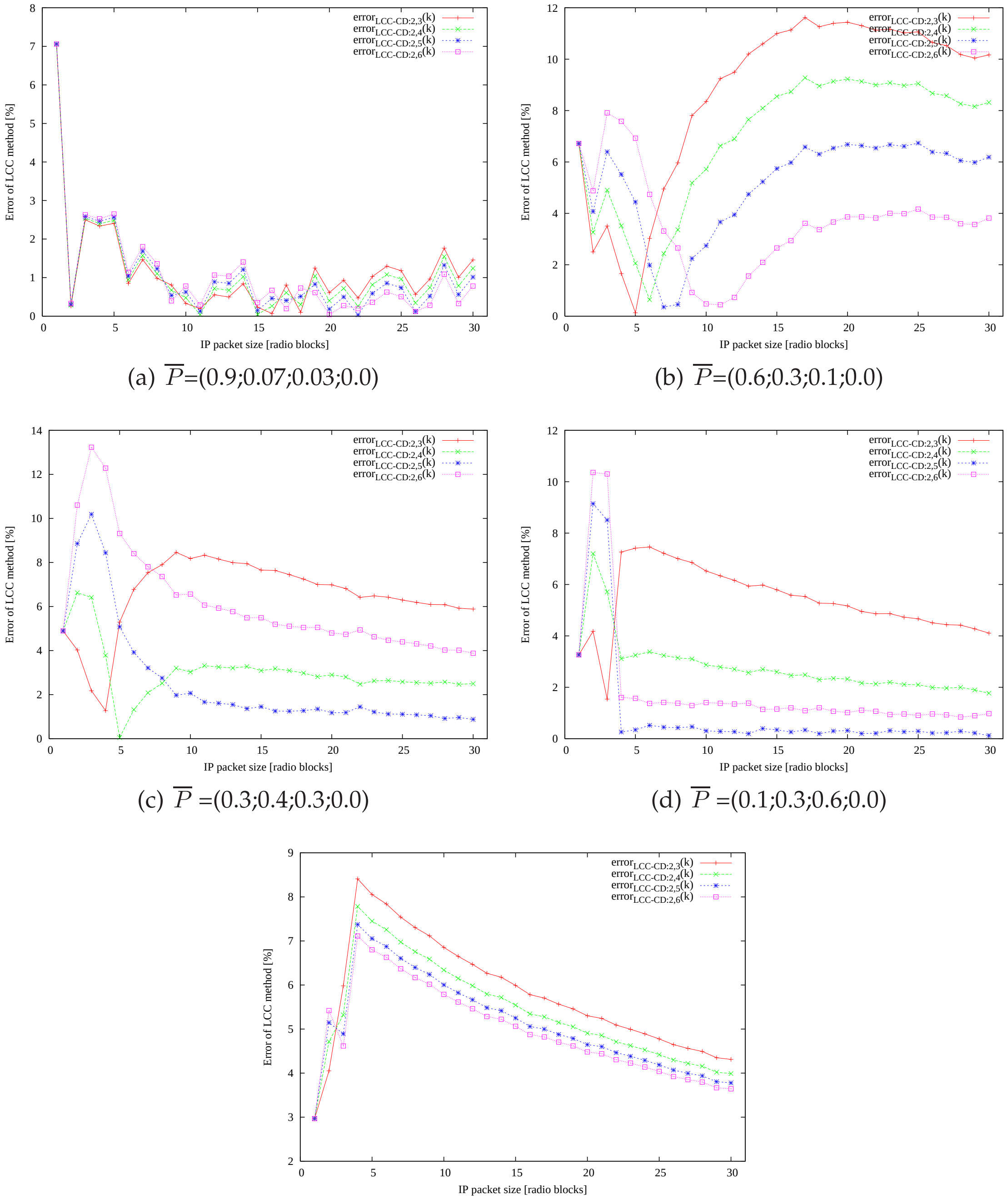
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.12: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

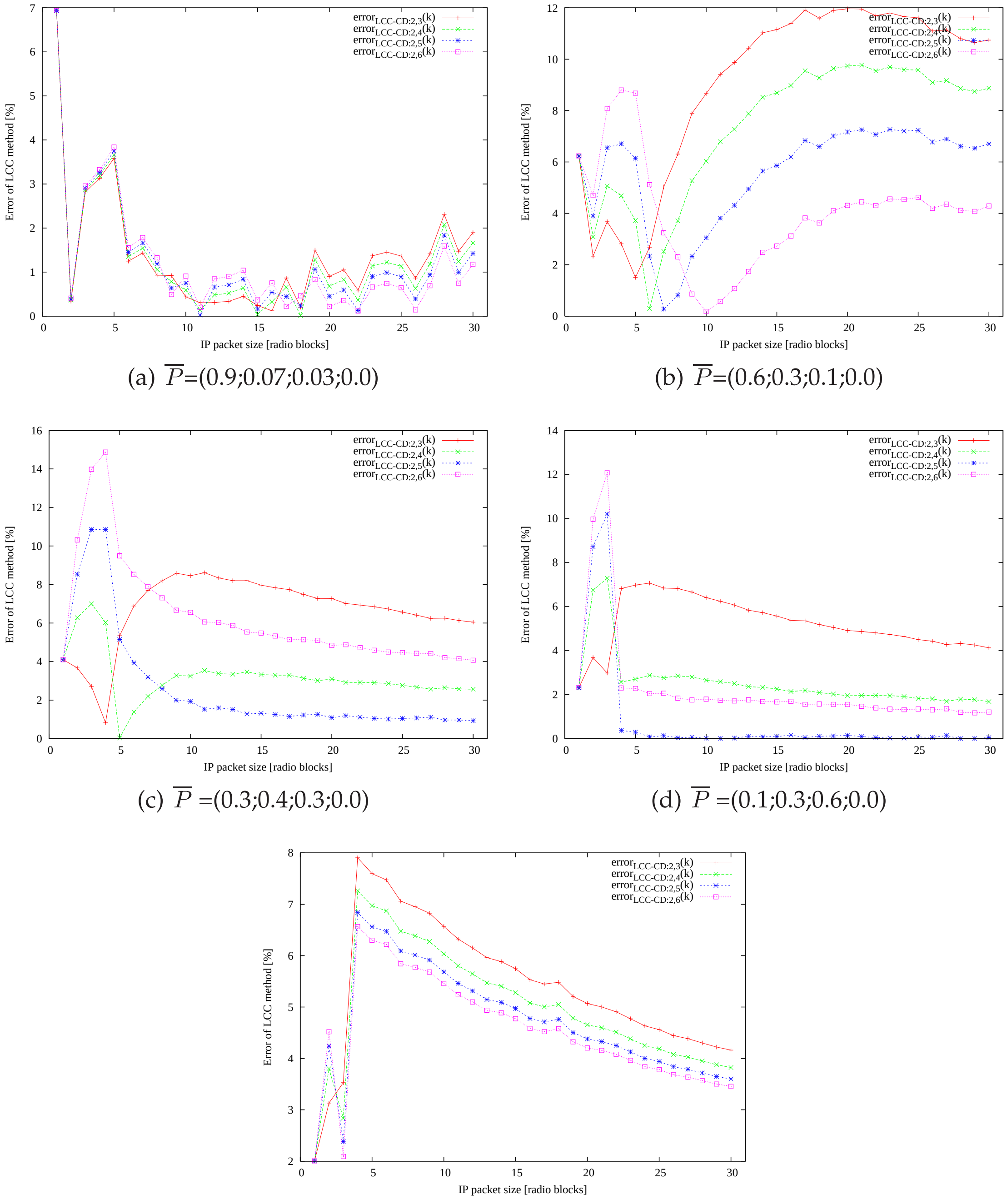
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.13: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

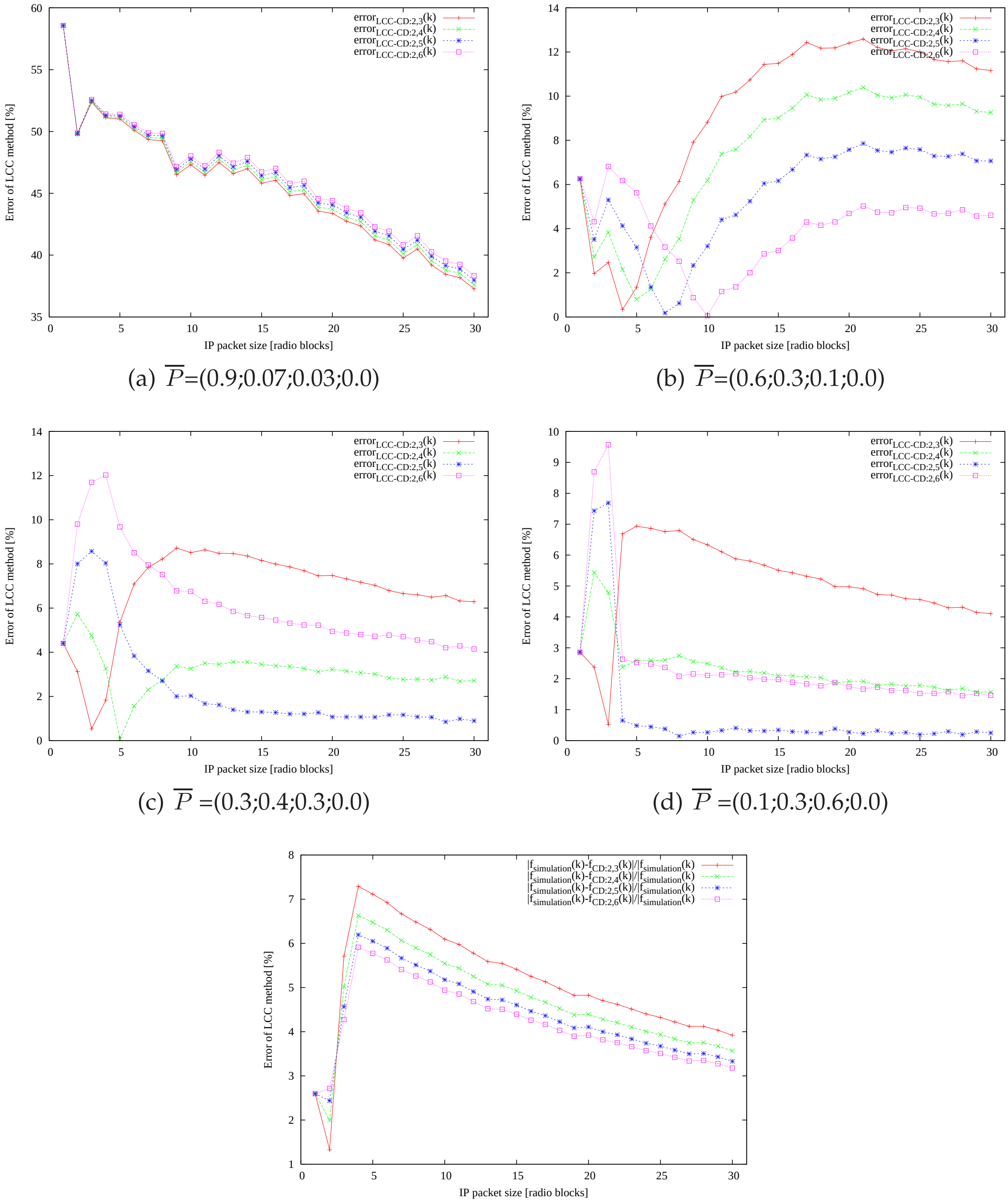
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure G.14: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

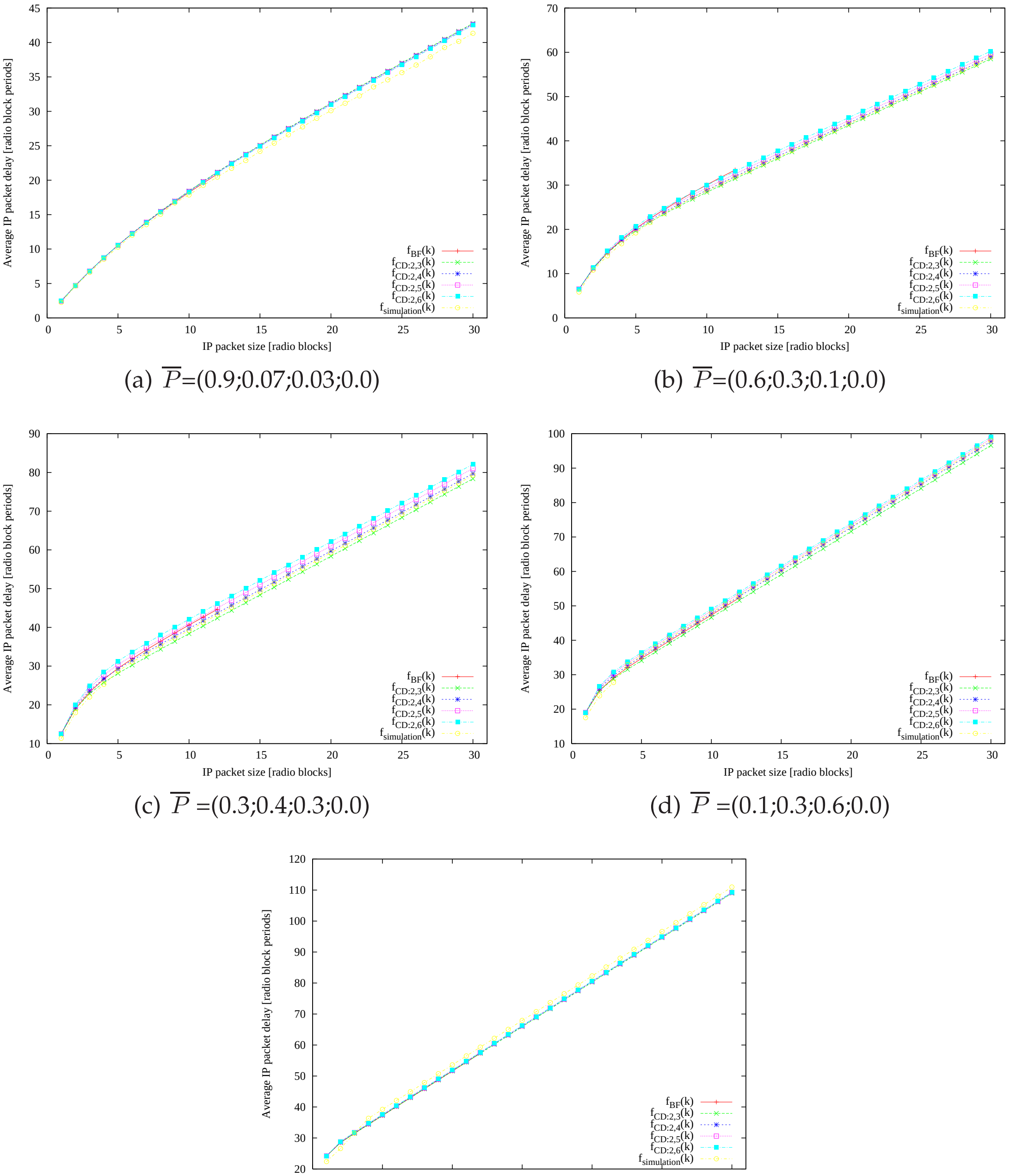
Figure G.15: *Error of Low Computation Complexity method - results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.

APPENDIX H

Final comparison between Simulation, BF and LCC results

This appendix presents the relationship between values of the average ARQ component of IP packet delay obtained from simulation and the same values obtained as a result of use of BF and LCC prediction techniques. The LCC method plots consists of four trends for each plot, each trend representing different curve fitting points used in the LCC method.

Each page shows results for five different radio channel conditions, represented by five different $\overline{P}$ vectors, for a given $\overline{\Delta}$ vector. There are 15 different ARQ loop settings analysed, 15 different $\overline{\Delta}$ vectors, which are specified in section 4.5.2.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

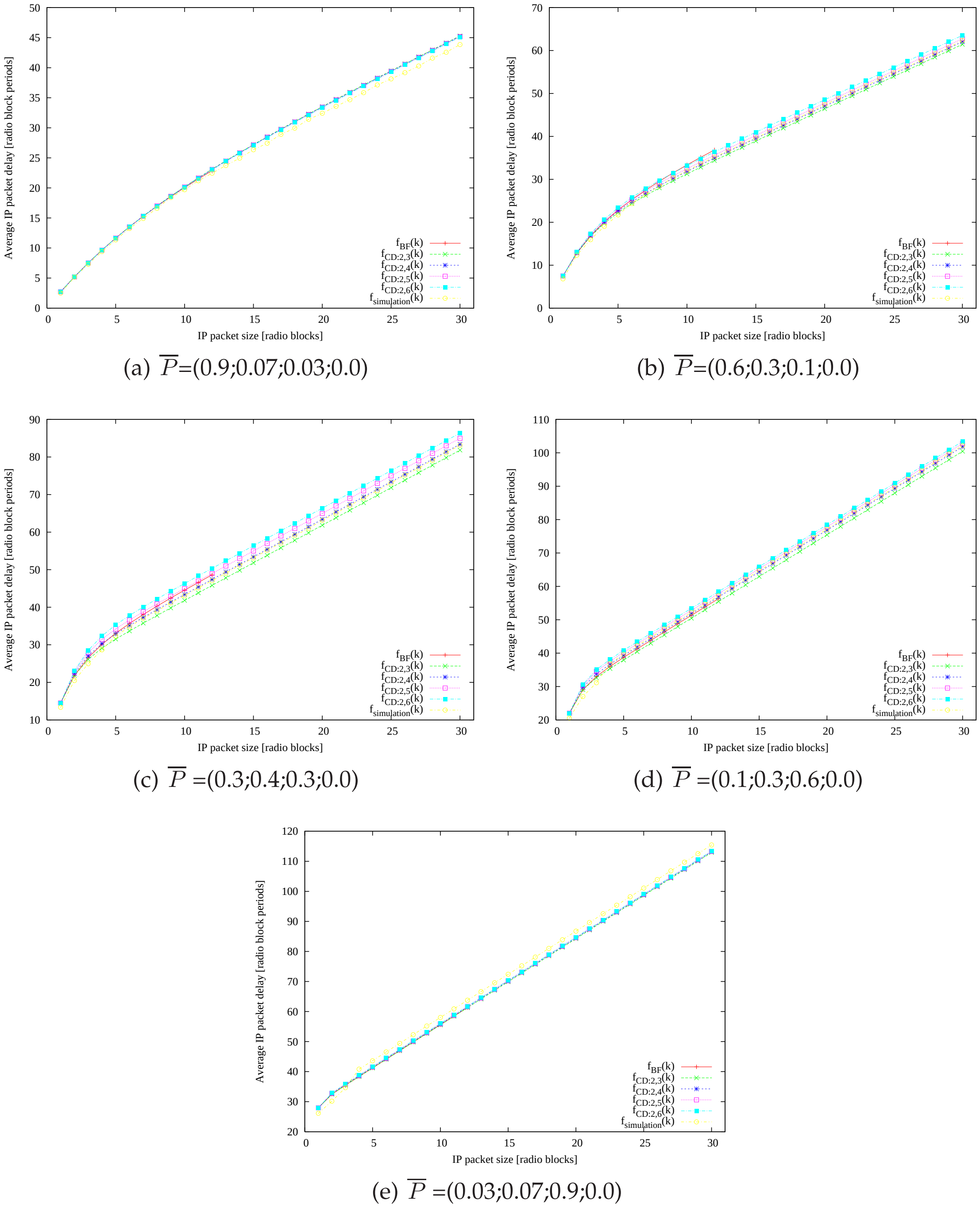
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.1: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 8; 13; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

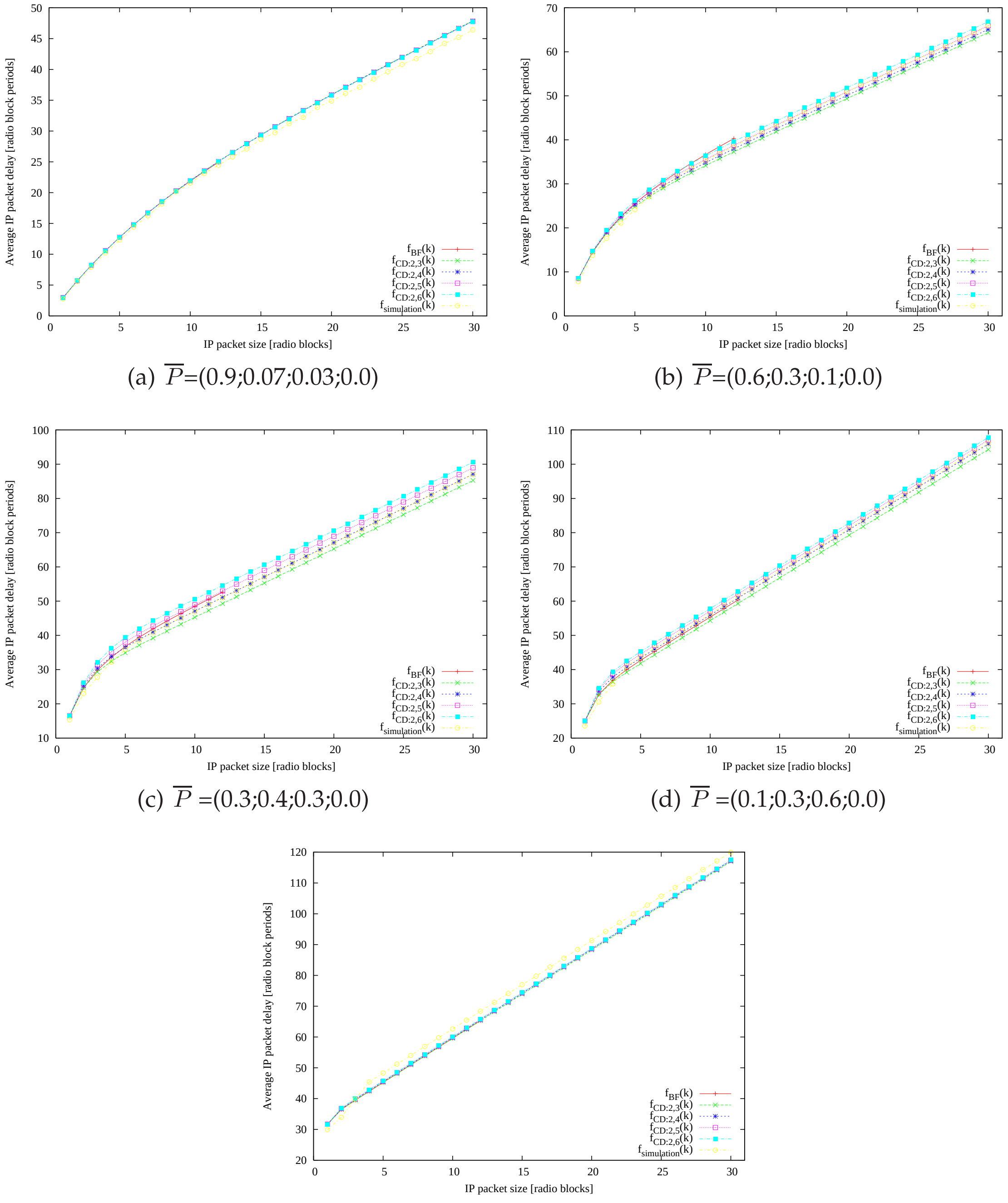
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.2: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 10; 15; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

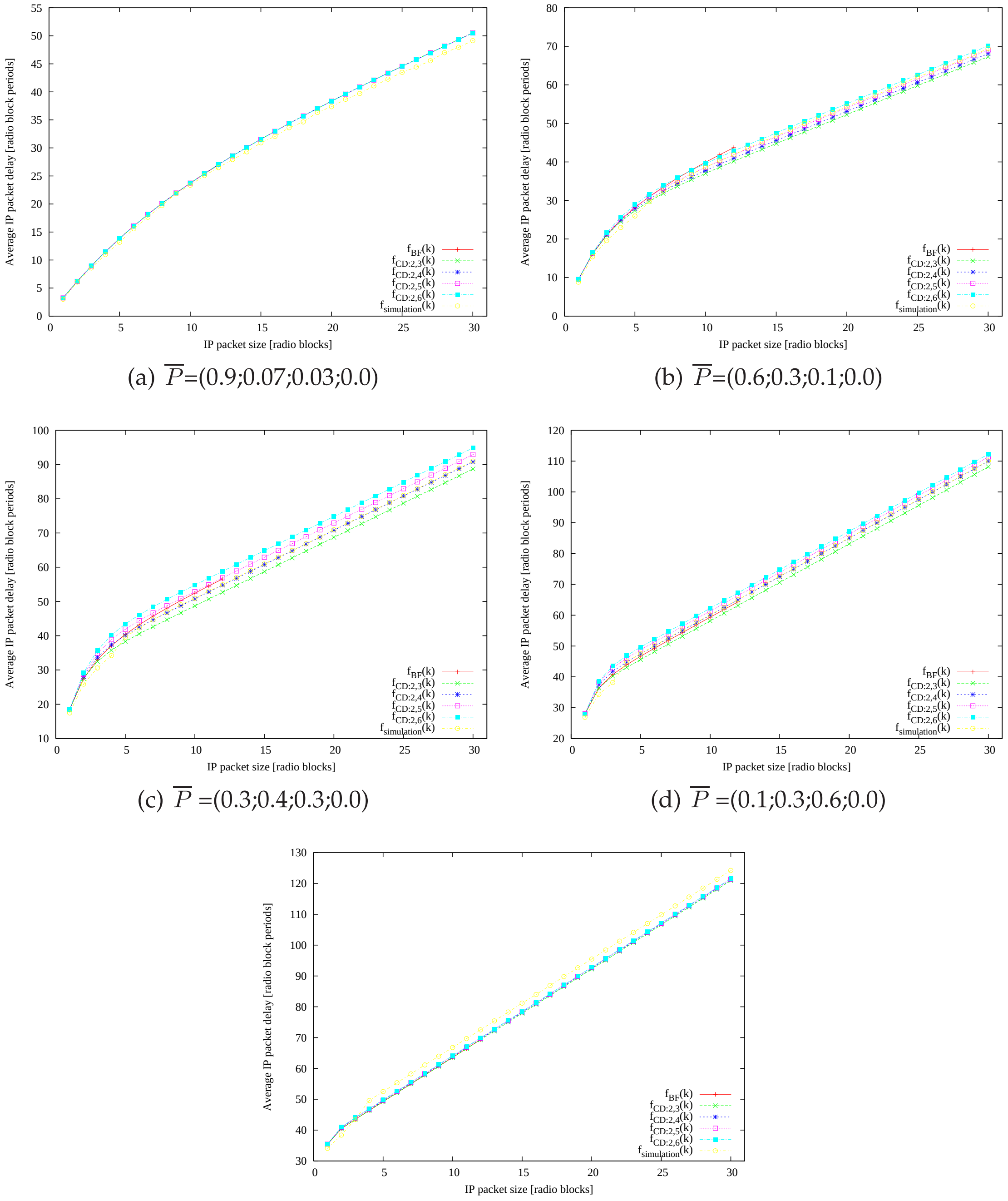
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.3: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 12; 17; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$*.*



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

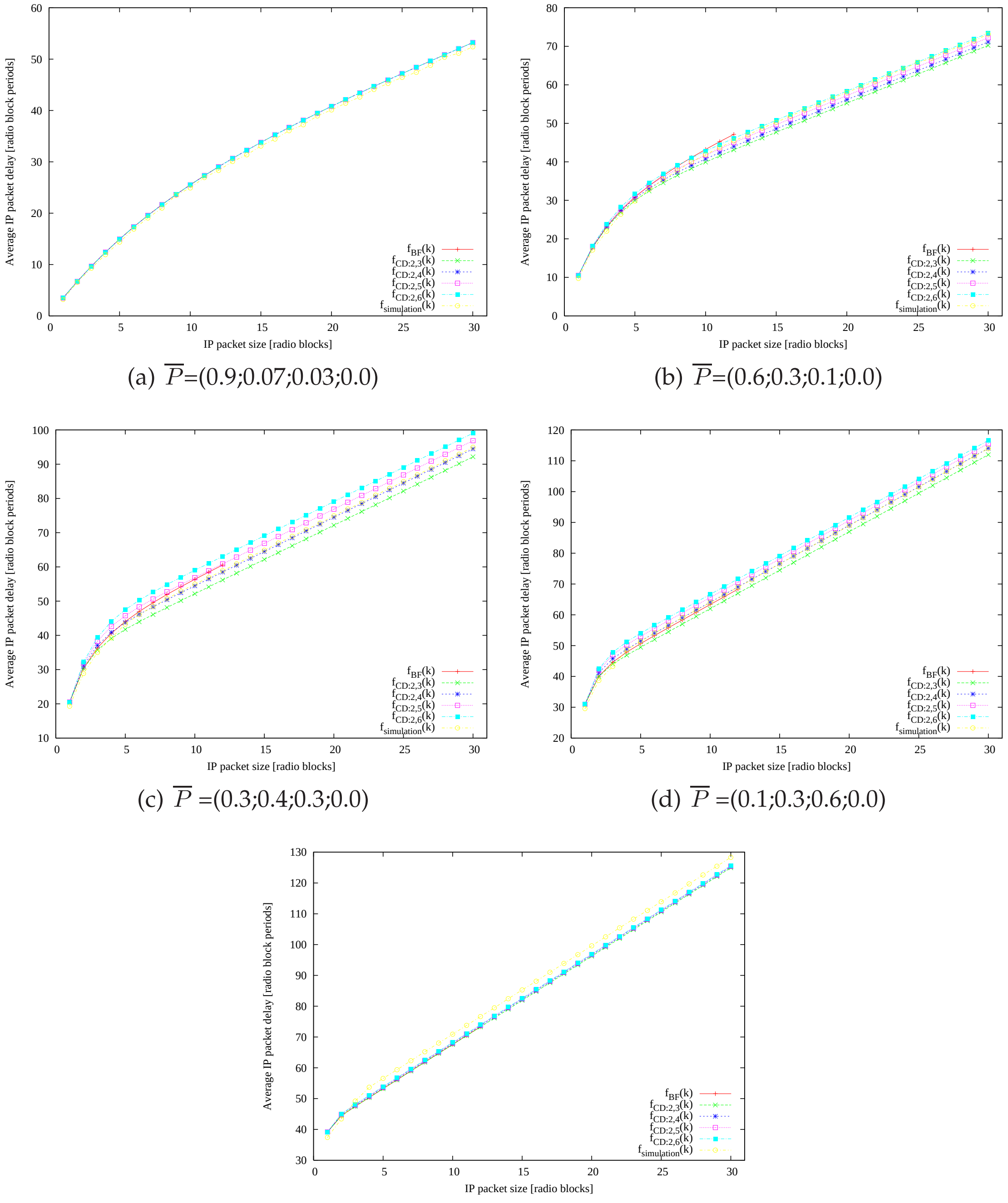
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.4: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 14; 19; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$*.*



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

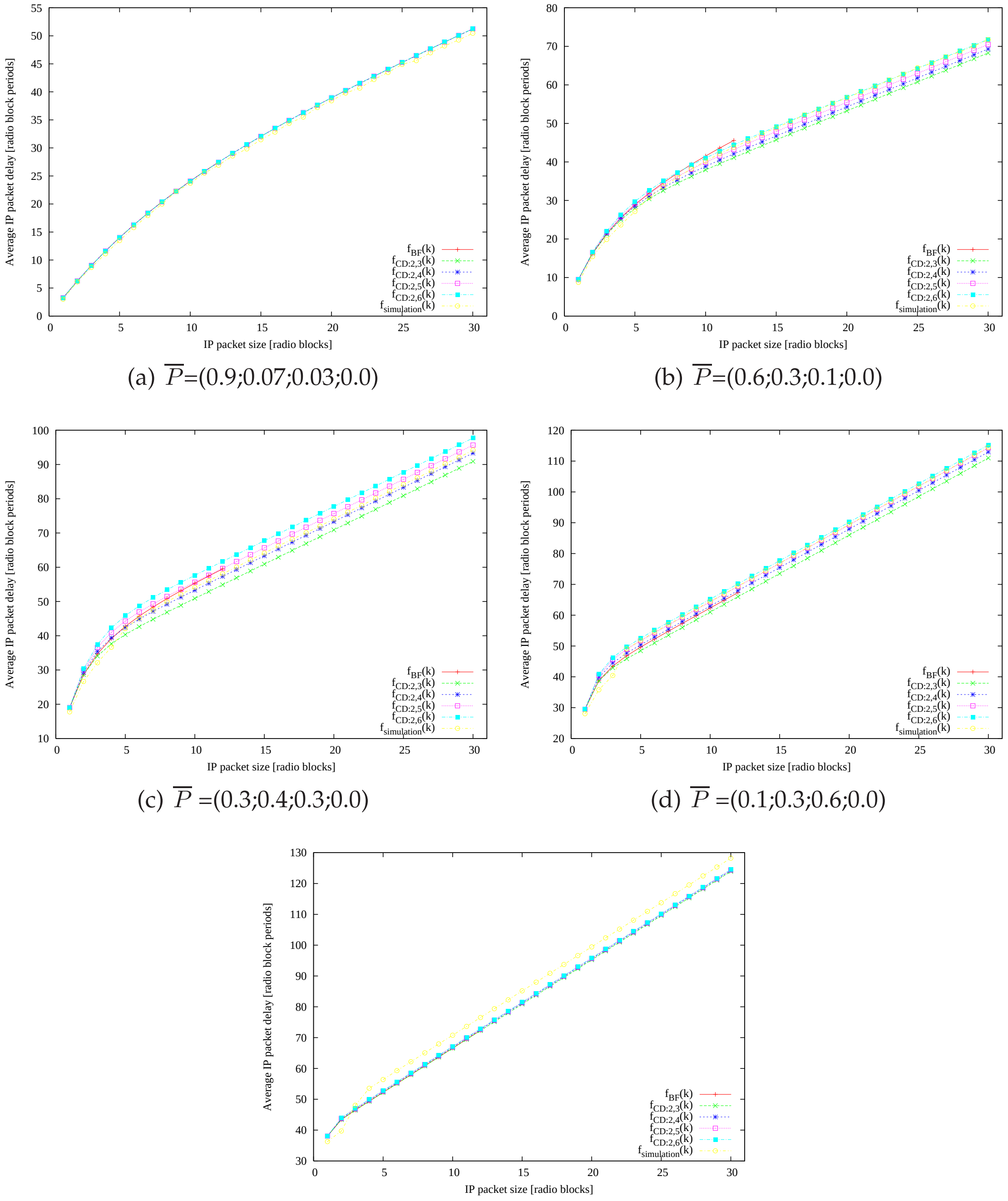
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.5: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 16; 21; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

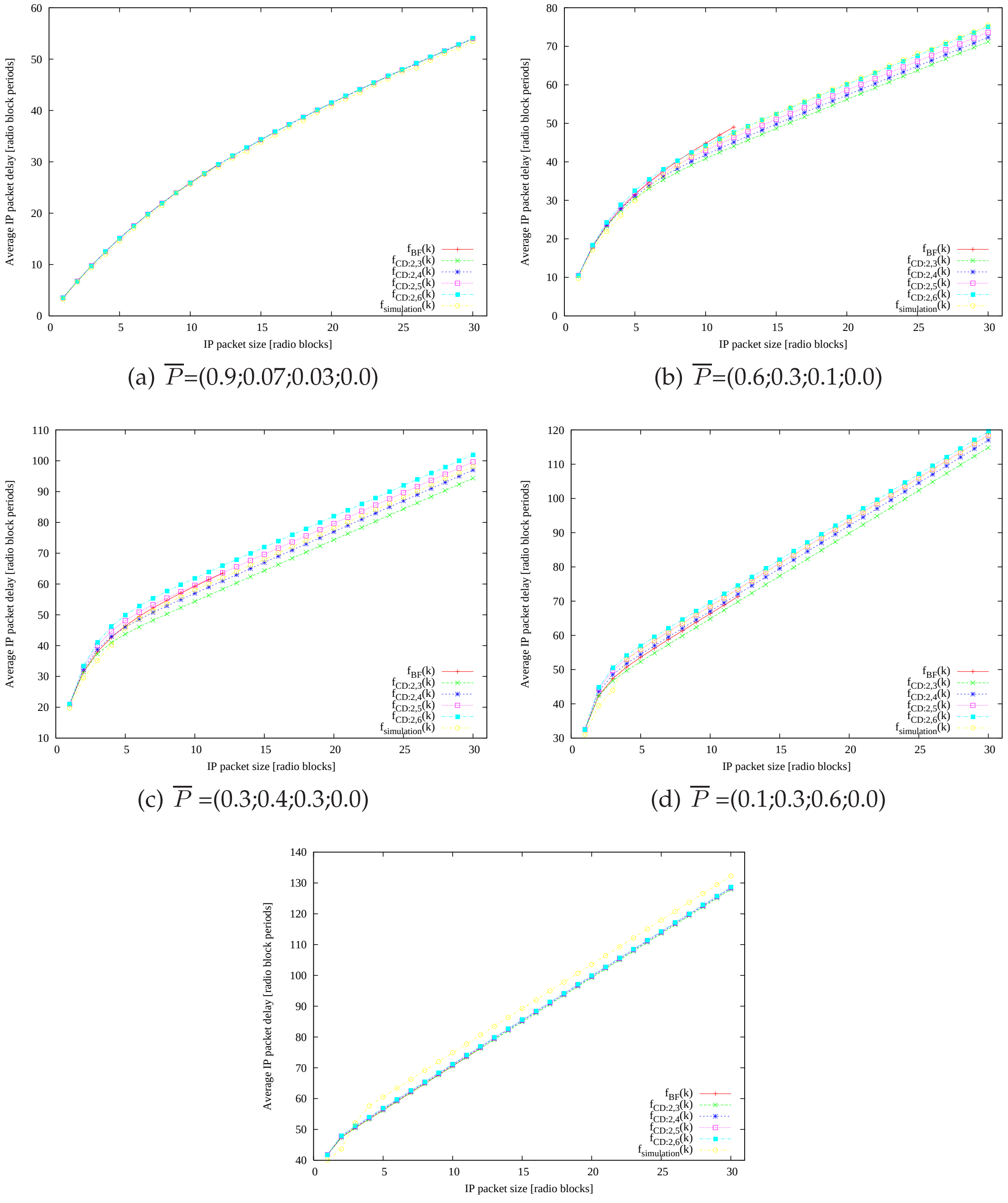
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.6: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 13; 23; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

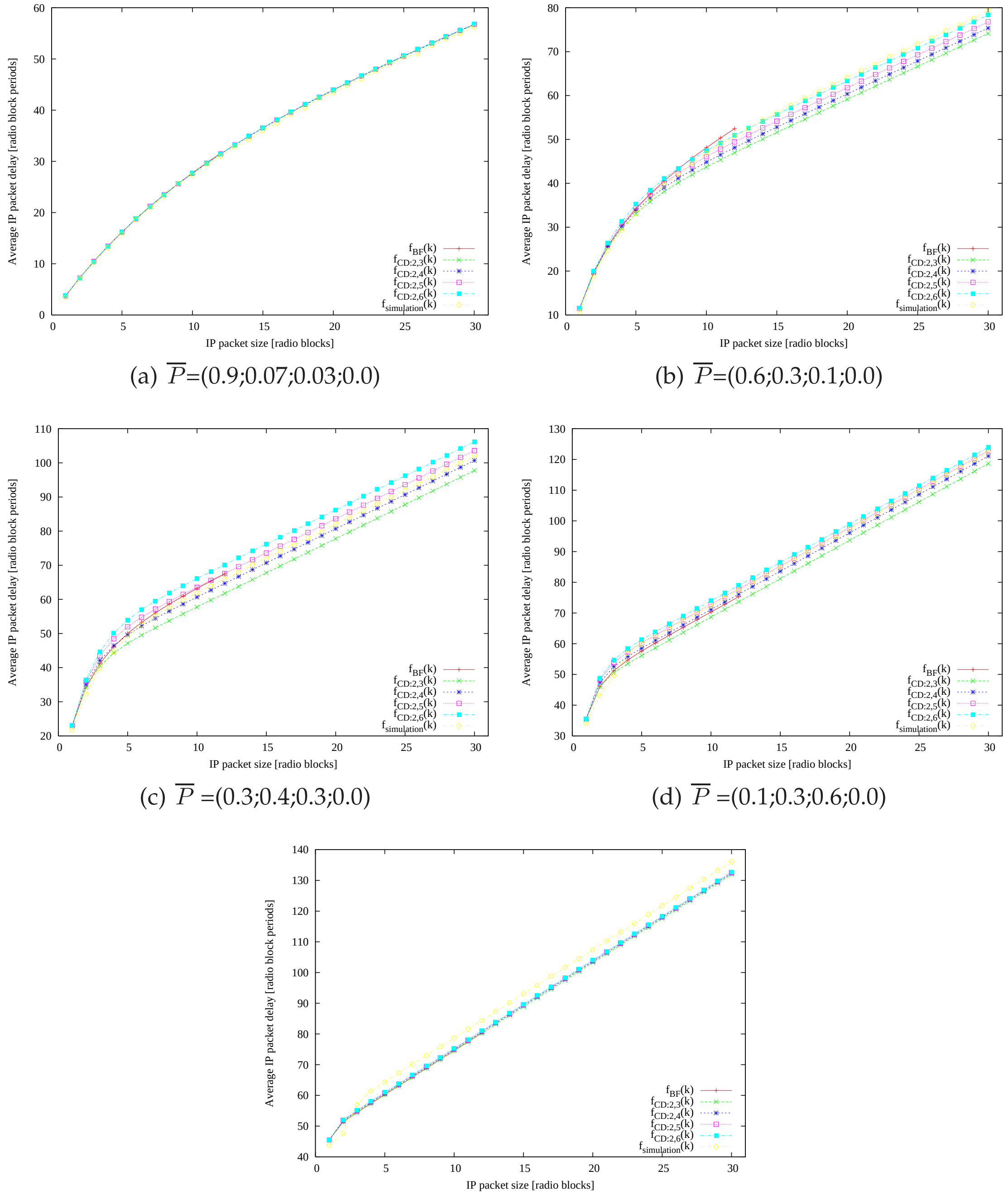
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.7: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 15; 25; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

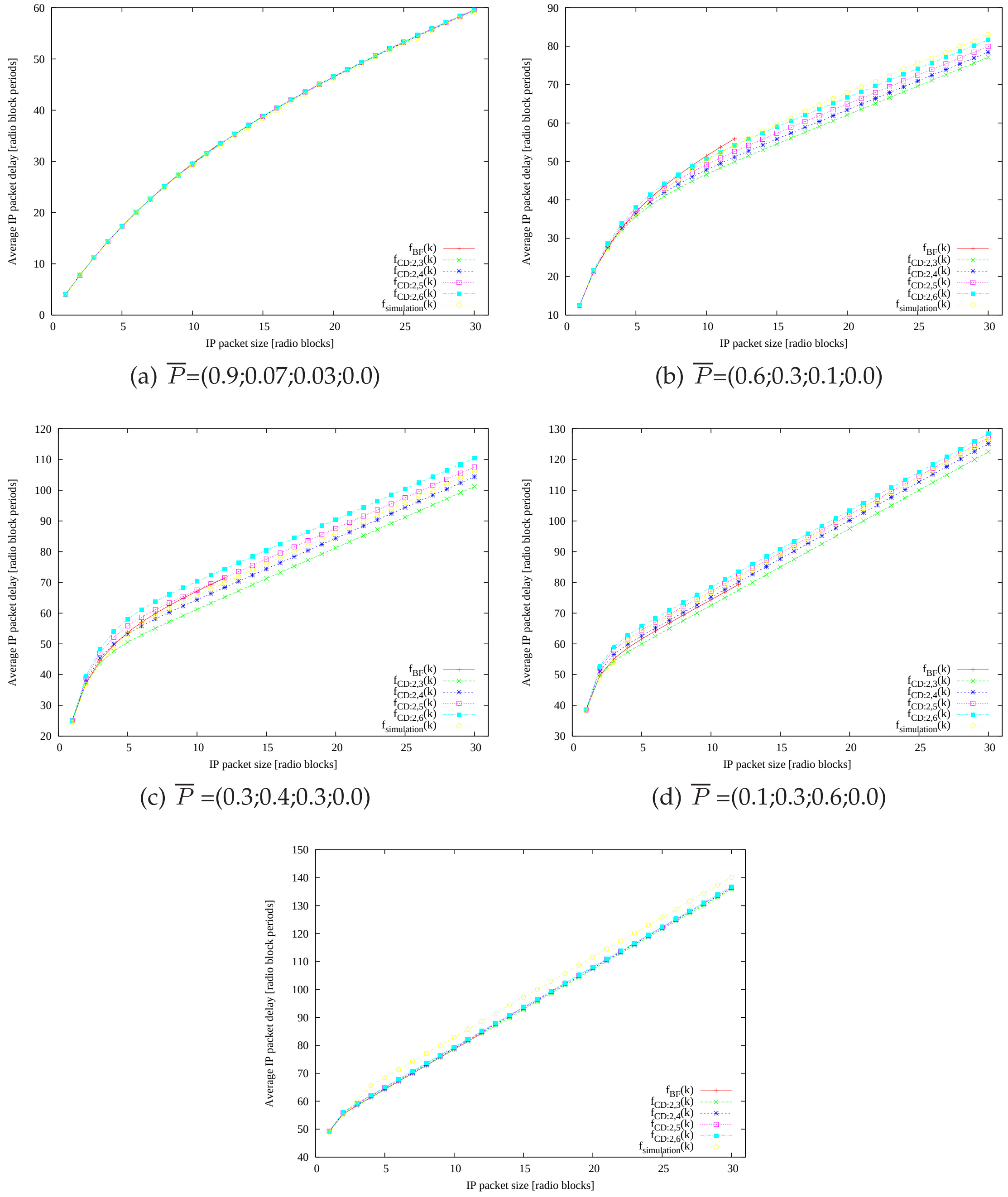
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.8: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 17; 27; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

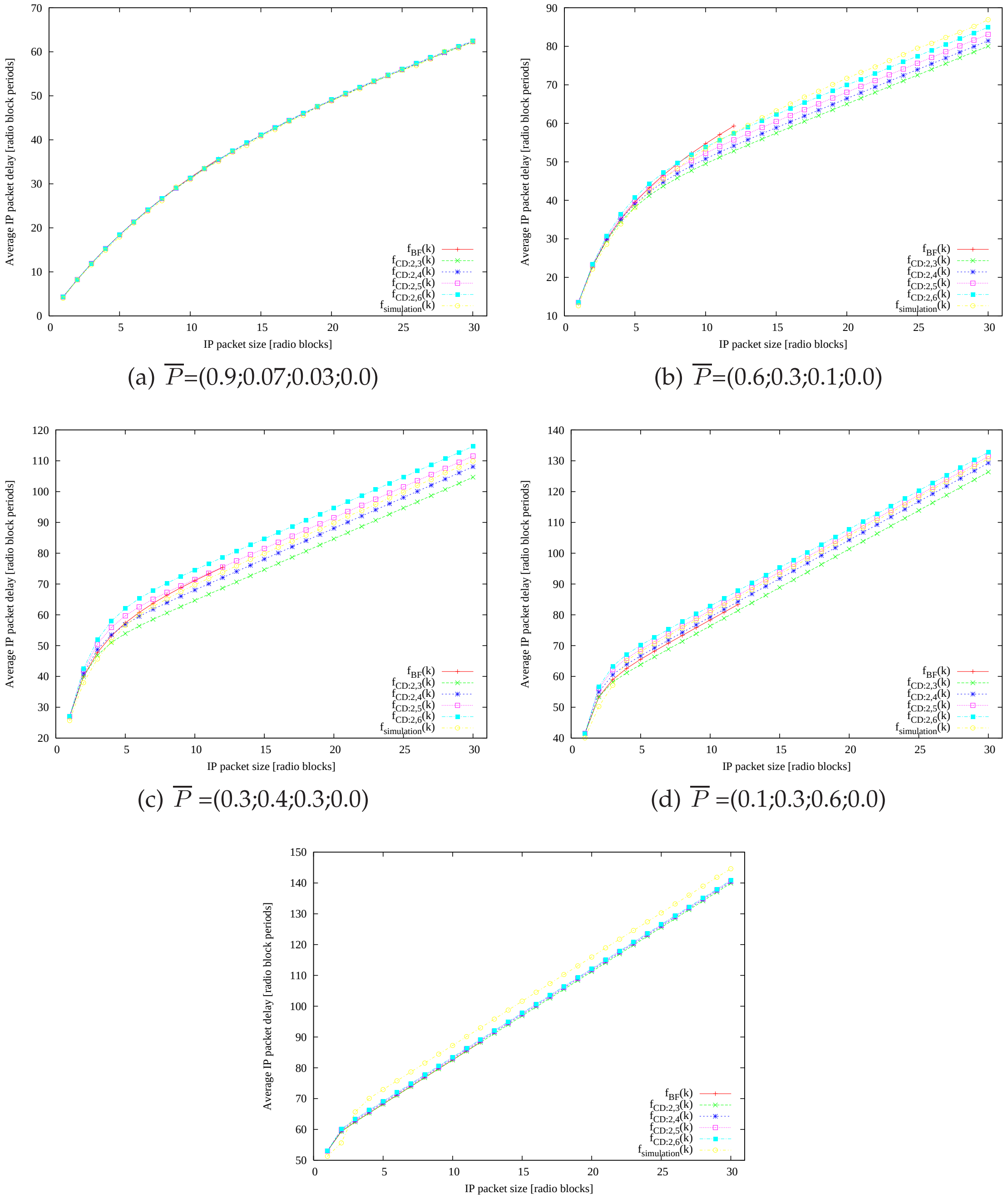
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.9: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 19; 29; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

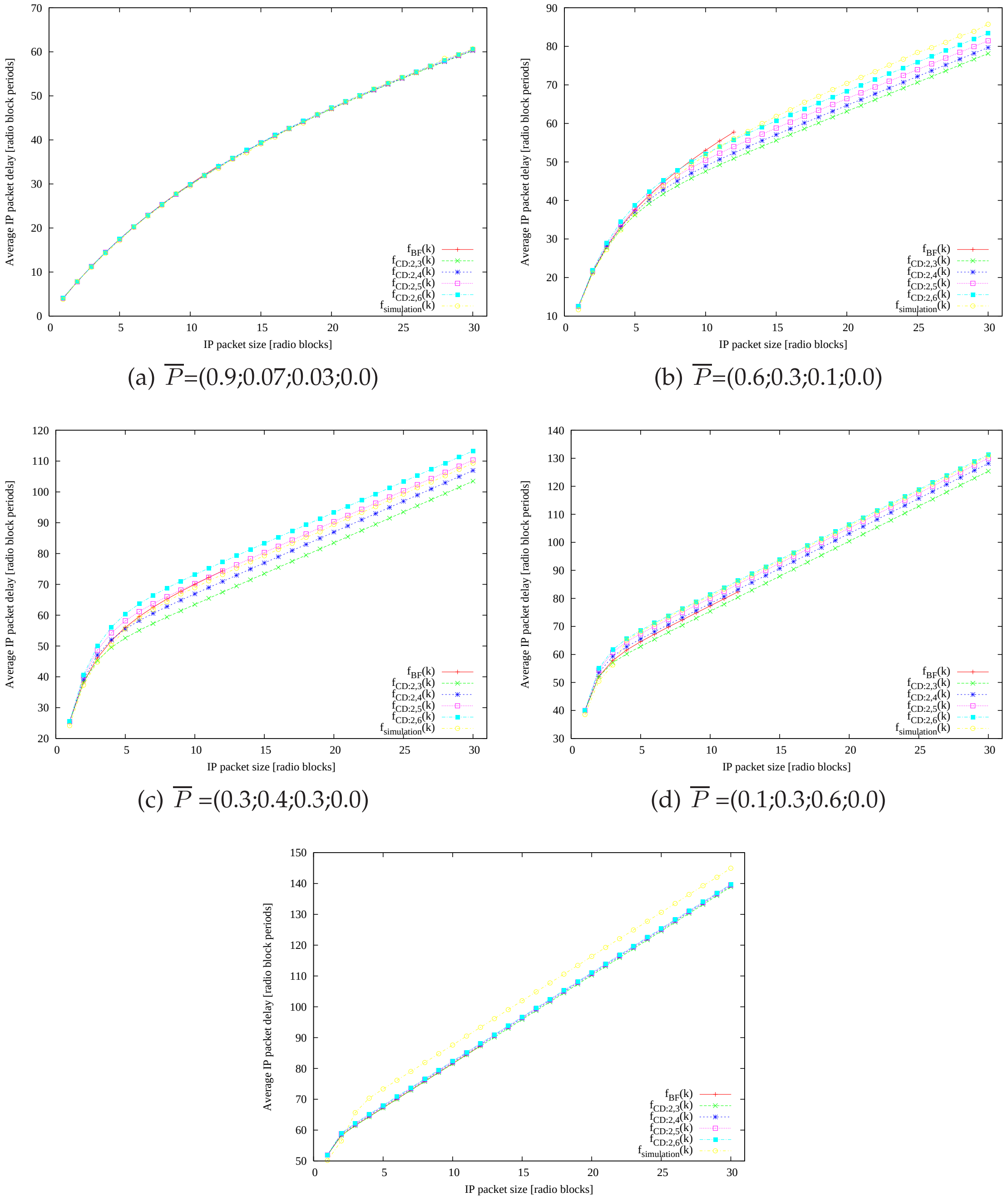
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.10: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 21; 31; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

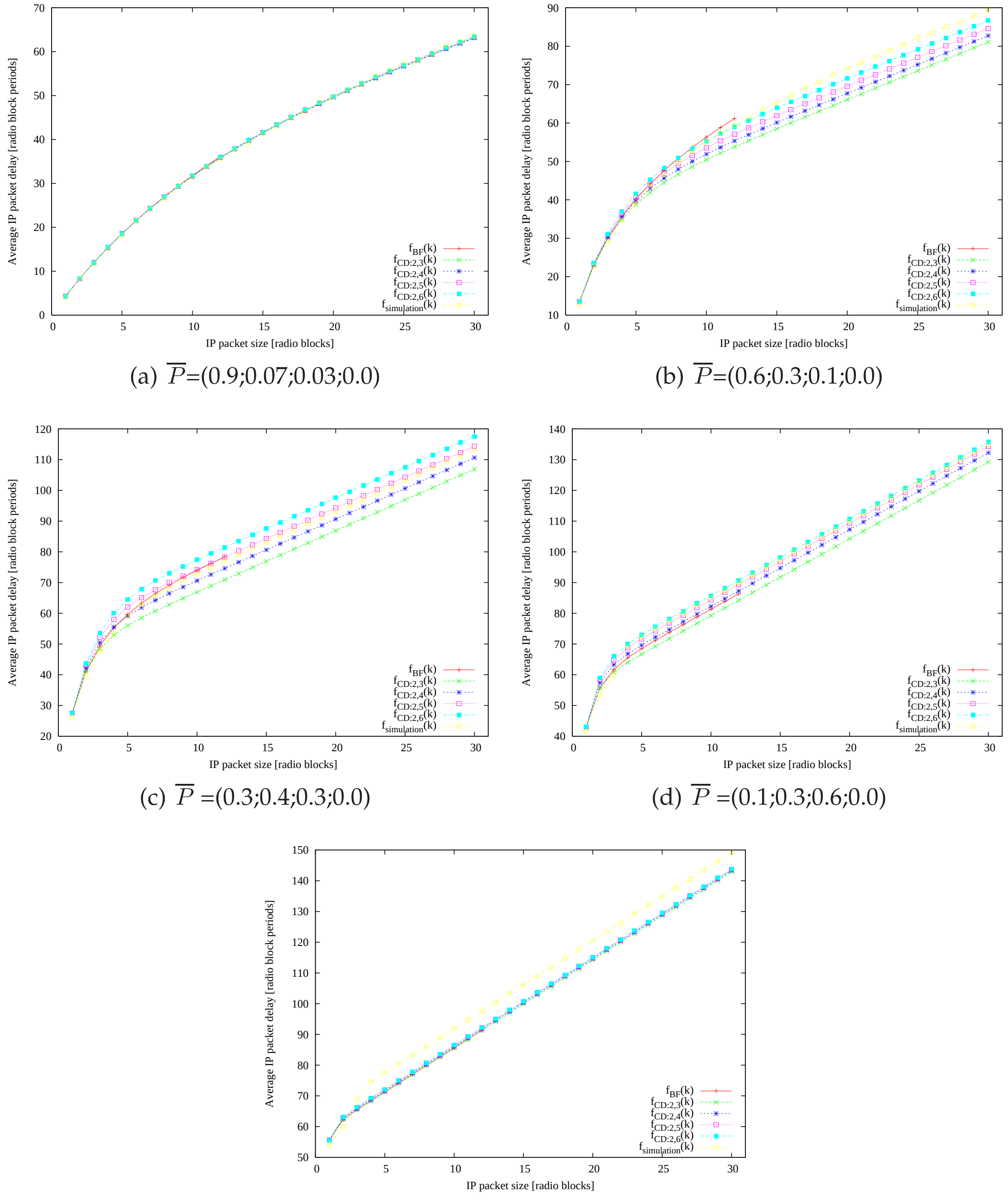
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.11: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 18; 33; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

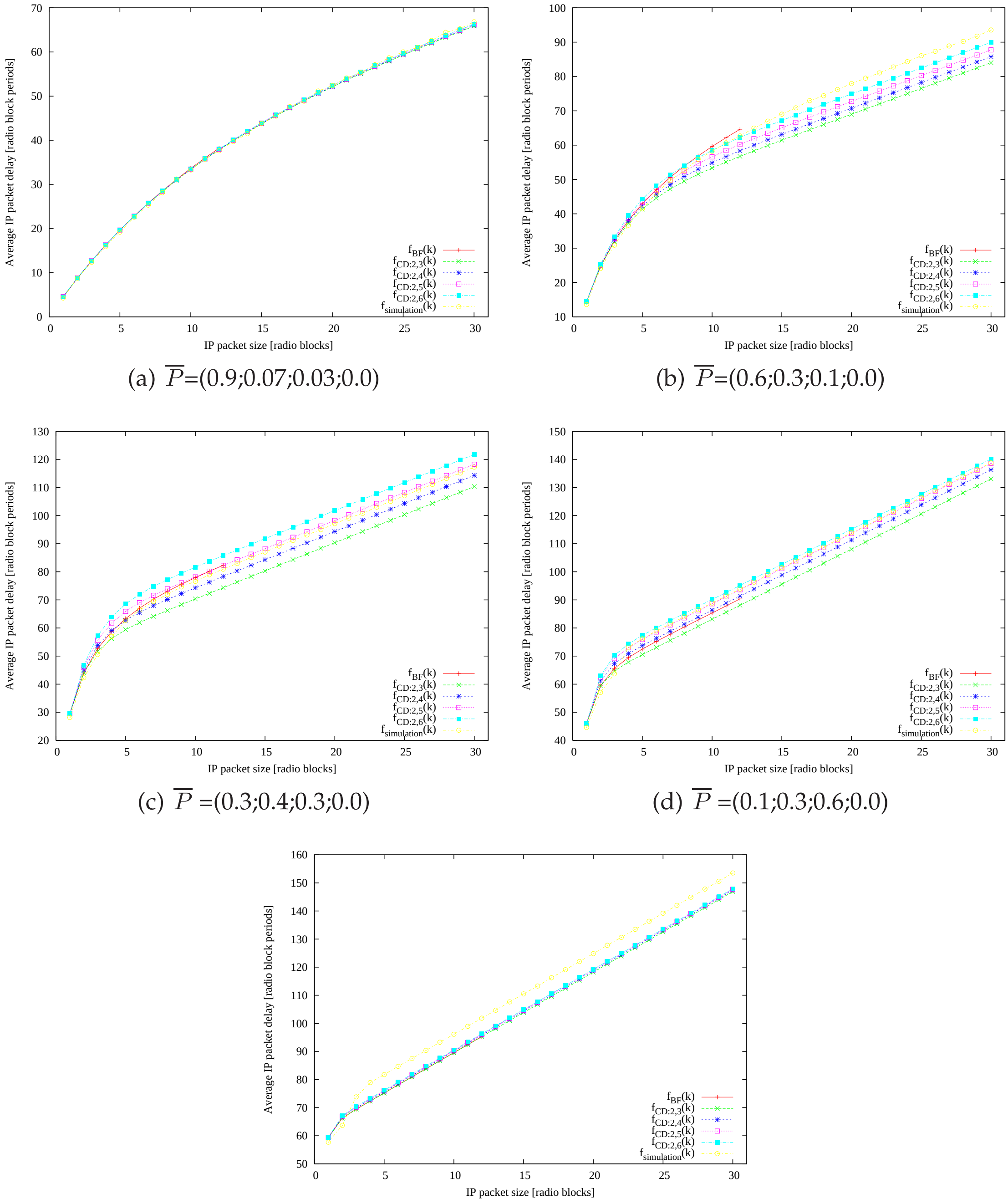
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.12: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 20; 35; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

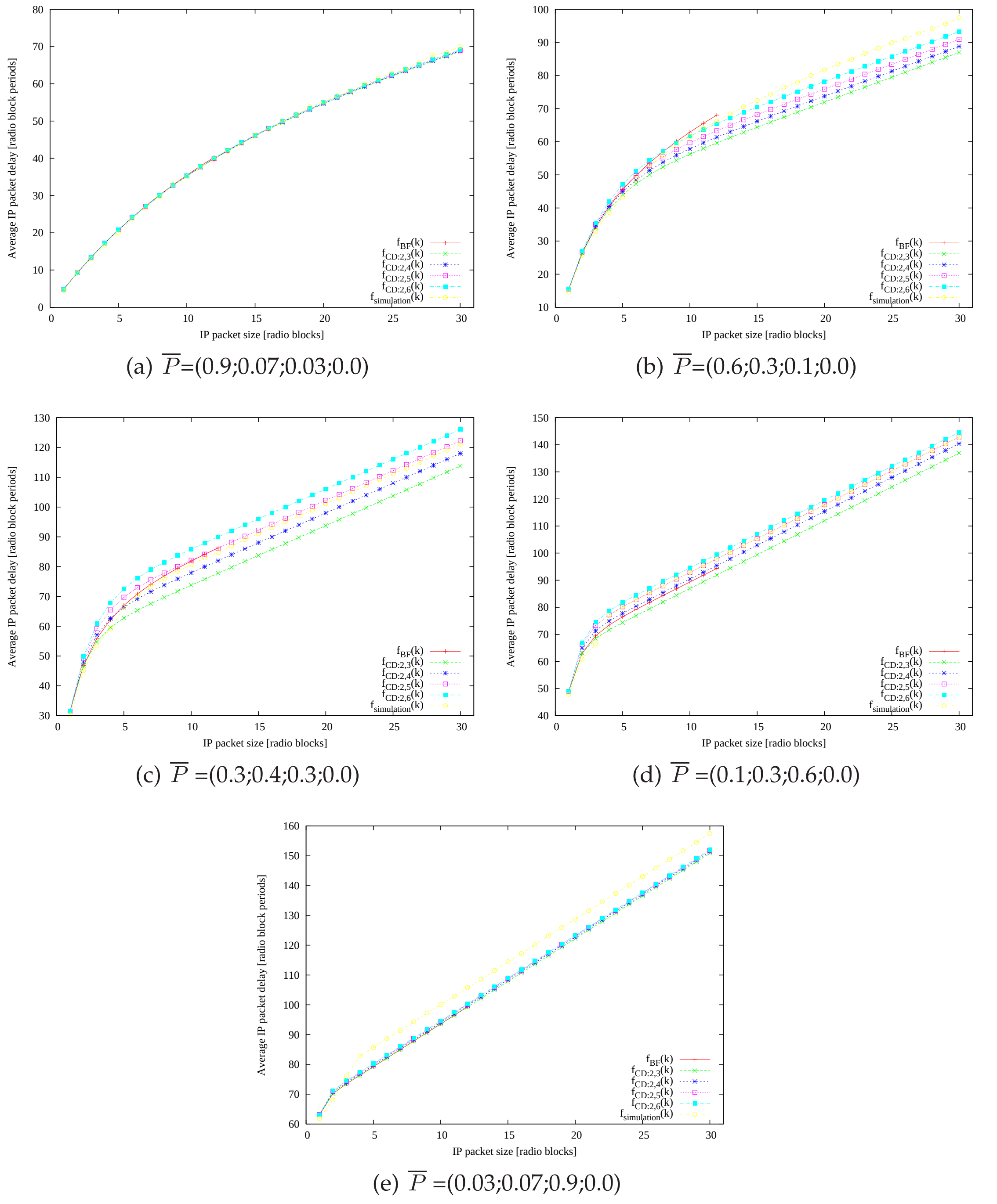
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.13: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 22; 37; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

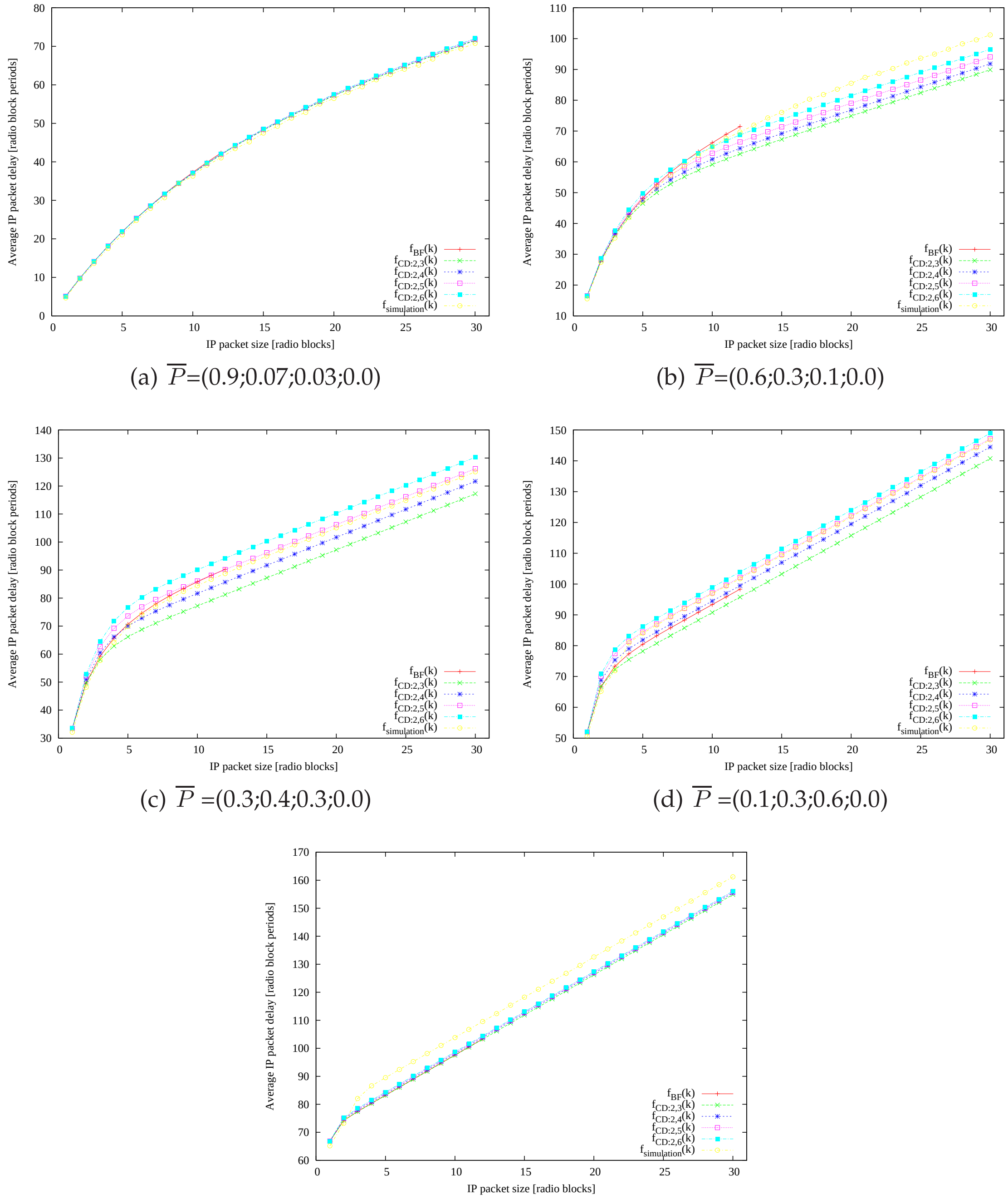
(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.14: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 24; 39; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.



(a) $\overline{P}$=(0.9;0.07;0.03;0.0)

(b) $\overline{P}$=(0.6;0.3;0.1;0.0)

(c) $\overline{P}$ =(0.3;0.4;0.3;0.0)

(d) $\overline{P}$ =(0.1;0.3;0.6;0.0)

(e) $\overline{P}$ =(0.03;0.07;0.9;0.0)

Figure H.15: *Final comparison between simulation, BF and LCC - results for* $\overline{\Delta} = [0; 26; 41; 0]$ *and* $\overline{P} \in \{P_0, P_1, P_2, P_3, P_4\}$.