Performance Analysis of CDMA Data Networks with Rate and Delay Variations

by

Vikas Paliwal

B.Tech., IIT Kanpur

A thesis submitted to the

Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

Ottawa, Canada

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

"*Performance Analysis of CDMA Data Networks with Rate and Delay Variations*"

submitted by Vikas Paliwal, B.Tech
in partial fulfillment of the requirements for the degree of
Master of Applied Science.

Thesis Supervisor
Professor Ioannis Lambadaris

Chair, Department of Systems and Computer Engineering
Professor Rafik Goubran

Carleton University
September 24, 2004

Abstract

The behavior of wireless links in third generation wireless data systems based on cdma2000 standard have profound impacts on the performance of transport layer protocols. This is due to greater variations in round trip times (RTTs) experienced by the TCP agents over these links, primarily due to two reasons - error recovery through link layer retransmissions and allocation/de-allocation of higher data rate channels. In this thesis, an exhaustive study of the nature and impacts of these two features is presented. Based on simulations and analysis, this thesis makes the following key contributions:

First, the conditions under which sharp delay variations and residual errors occur in presence of link layer recovery mechanisms are identified and the impacts of these in degrading the system performance and TCP throughputs is presented based on an implementation of cdma2000's link layer protocol, Radio Link Protocol(RLP), that was developed as part of this work.

The second part of this thesis quantifies the problem of network congestion due to variable rate radio links. This problem is characterized using an analytical model for losses based on continuous flow approximation as well as an extensive simulation setup. We investigate this problem under variety of scenarios and suggest possible solutions to counter the problem.

Acknowledgments

My greatest fear of writing a page of acknowledgements is that someone will be forgotten. There are so many people whose influence will, either directly or indirectly, manifest itself in this work, and I would very much like to thank them all.

I will start by thanking my thesis supervisors Professors Ioannis Lambadaris and Biswajit Nandy. The last year under their guidance has been a wild ride, and they had a profound, positive influence on my views of both science and life. I thank them for their patient attempts to help me understand the field of communication networks.

As well, I would like to thank Dr. Parsa Larjiani. His patience and helpful advice throughout my research is very much appreciated. Not only did he help me in identifying a challenging and satisfying thesis topic but also helped me throughout this work by means of his invaluable feedback and comments.

At the same time, I would like to thank the entire Broadband Networks Lab at Carleton. I have never encountered a group so exceptional, both for their individual talents and for the cooperative, friendly, and synergistic environment they foster. Learning from all of them has been a pleasure. A special thanks is extended to Rupinder Makkar for his enthusiastic help and support in getting the initial head-start on the project. Beyond the group at Carleton, it is a pleasure to acknowledge useful conversations with numerous users of *ns2* simulator who supplied me the useful feedback while I was developing the RLP module. As well, a thanks to my friends in Ottawa - Makarand, Vishal and Nikhil - for providing me the moral support during the last two years. Finally, thanks to all of my friends and family back there in India.

To my parents

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Objective of the Thesis 5

1.3 Thesis Outline 6

1.4 Thesis Contributions 6

2 Data Services in cdma2000 Networks **9**

2.1 Physical Layer 11

2.2 Radio Link Protocol (RLP) 12

2.3 Point-to-Point Protocol (PPP) 14

2.4 Transmission Control Protocol (TCP) 14

3 Simulation Setup **16**

3.1 Physical Layer 17

3.2 Frame Types 18

3.3 RLP Operation 19

3.4 Model Verification 20

4 Impact of Wireless Errors **24**

4.1 Introduction 24

4.2 Results and Analysis 24

4.2.1 Link Layer 25

4.2.2 IP Layer . . . 32
4.2.3 TCP Performance . . . 33
4.3 Supplemental Channel Allocation Decisions . . . 37

5 Congestion due to Rate Variations **40**
5.1 Introduction . . . 40
5.2 Modeling Bandwidth Oscillations . . . 44
5.3 Simulation Model . . . 48
5.4 Results and Analysis . . . 48
5.4.1 Dependence on window size, bandwidth swing and round-trip delay . . . 48
5.4.2 A worst-case scenario . . . 51
5.4.3 Comparison with a scenario with variable number of users . 53
5.4.4 Impact of rise time for aggregate bandwidth switching . . . 53
5.4.5 Typical load scenario . . . 56
5.5 Stable Operation of RED Mechanism . . . 59
5.5.1 Factors contributing to queue overshoots at BSC . . . 61
5.5.2 Results . . . 63

6 Future Work and Conclusions **67**
6.1 Wireless Errors . . . 67
6.2 Rate Variations . . . 68
6.2.1 Modifications to queue management . . . 68
6.2.2 Fixing TCP . . . 74
6.2.3 Scheduling bandwidth allocation/deallocation . . . 75
6.3 Conclusions . . . 76

List of Figures

1.1 A network view of a mobile's TCP flow. 5

2.1 Protocol stack used in cdma2000 data network. PDSN is packet data service node and BHL is back-haul link protocol over T1 link. 10

2.2 An example of link-layer error-recovery through RLP retransmissions. 11

3.1 Implementation of RLP agent in *ns2* 17

3.2 Two-state Markov chain for simulating time-correlated frame errors (p and q are state transition probabilities from *good* state to *bad* state and vice versa). 19

3.3 State transition diagram for an RLP sender 21

3.4 State transition diagram for an RLP receiver 21

3.5 Comparison of analytical model[5] and developed simulator. 22

4.1 Impact of physical layer frame errors (i.i.d) at link layer. 26

4.2 Frame delays for various levels of correlation for a constant mean FER. 30

4.3 Normalized IP packet delays. 34

4.4 TCP behavior for $R_{phy} = 9.6$ kb/s (sequence numbers are in *modulo*-80 fashion). 36

4.5 TCP throughputs for various values of raw link rates and physical layer FERs (i.i.d.) 37

4.6 TCP behavior for R_{phy} = 153.6 kb/s (sequence numbers are in *modulo*-80 fashion). 38

5.1 Buffering scheme in cdma2000 data services (The symbols used are described in table 5.2). 41

5.2 TCP trace for SCH allocation. 43

5.3 Model for losses due to rate change in a radio link. 45

5.4 Variation of losses due to bandwidth change with, (a) mobile's higher switched rate, b_{lh} , (b) TCP sender's window size, W, and, (c) round trip propagation delay, T. (Other parameters are kept constant at b_{ll} = 9.6 kb/s, b_b = 38.4 kb/s, B= 5 kB, packet size for *ns2* simulations = 500 bytes) 50

5.5 Plots for simultaneous 16X SCH allocation to all 100 mobiles at 50s. 52

5.6 Plots for more users joining simultaneously at 50s. 54

5.7 Illustration of rise and fall times, burst and delay durations. 54

5.8 Impact of rise time of aggregate bandwidth on queue behavior at shared buffer. 55

5.9 Fractional dead-period (90% underutilization definition) for various values of burst-delay duration. 58

5.10 Scheme for increasing the radio links' rates. 61

5.11 Queueing behavior in shared buffer and link queues with varying levels of overload. For each of the figures, the number of packets dropped in 5 seconds after the channel allocations (i.e. between 50-55s) are also shown. The values of t_r and β are constant at 1s and 30% respectively. 65

5.12 Threshold for stable operation with default settings 66

6.1 Setup for RED(from [20]) 69

6.2 Throughput versus Average Queue Size 71
6.3 Proof of non-optimal operation of RED 71
6.4 Modified RED algorithm 73
6.5 Enhancement offered by Newreno TCP's fast recovery algorithm as against Reno TCP in figure 5.2. 75

Nomenclature

1X-RTT	Single Carrier Radio Transmission Technology
3GPP2	Third Generation Partnership Programme 2
ACK	Acknowledgement Packet
ARQ	Automatic Repeat Request
BHL	Back Haul Link
BSC	Base Station Controller
BTS	Base Transceiver Subsystem
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
DBP	Delay Bandwidth Product
EWMA	Exponential Weighted Moving Average
FCH	Fundamental Channel
FER	Frame Error Rate
IETF	Internet Engineering Task Force
i.i.d.	Independent, Identically Distributed
IMT	International Mobile Telecommunications
IP	Internet Protocol
MS	Mobile Station
ns2	Network Simulator
PDSN	Packet Data Service Node
PPP	Point to Point Protocol

QM	Queue Management
QoS	Quality of Service
RED	Random Early Discard
RF	Radio Frequency
RLP	Radio Link Protocol
RTO	Retransmission Timeout
RTT	Round Trip Time
SCH	Supplemental Channel
TCP	Transmission Control Protocol

Chapter 1

Introduction

1.1 Background and Motivation

It is now well understood that data services will dominate the cellular market in the future. The current trend in International Mobile Telecommunications-2000 (IMT-2000) is to move towards new technologies and corresponding standards that provide enhanced data services in cellular networks. However, providing high data rate services through wireless cellular networks is challenged by two main problems specific to wireless networks - scarce channel resources and channel errors. The first problem can be resolved by dynamic allocation of channels, and the latter can be handled using suitable local error recovery mechanism. Both of these approaches help in solving the respective problems, but introduce other serious problems, which this work seeks to analyze and quantify. This thesis is motivated by the following issues: delay variability due to losses and congestion due to dynamic channel sharing.

In order to mitigate the problem of wireless errors and to provide reliable data services, several link layer retransmission mechanisms e.g. Radio Link Protocol 3 (RLP)[1] in IS-2000, have been proposed to overcome the wireless errors through link layer retransmissions and hide these errors from transport layer protocols. These mechanisms help in overcoming the frame errors at link layer but they pose an additional problem of high variability in delay over the wireless link.

Some faster Automatic Repeat Request (ARQ)[2] mechanisms for newer standards like 1XTREME[3] have been proposed to overcome the losses with lesser delay variability, but the efficacy of all such mechanisms has to be thoroughly analyzed.

An accurate analysis of delays over wireless links is desirable for a variety of reasons. First, local retransmissions are an additional overhead for the link and hence decrease the link's effective data rate. Under extreme error conditions or poorly designed retransmission settings, link layer might not be able to recover the losses and Transmission Control Protocol (TCP) might have to come into action for error recovery thereby causing prolonged delays and reduced throughputs. Also, the IP (Internet Protocol) packets traversing over the link experience an additional delay due to retransmissions, thereby resulting in inflation of the retransmission timeout (RTO) of TCP and hence delayed recovery in case of loss of a packet. Finally, there may be cases where the delay variability is so intense that it causes the TCP sender to timeout resulting in TCP's retransmission and hence an even greater loss in throughputs.

Several attempts have been made to model the impact of link-layer retransmissions on TCP [4, 5, 6]. However, these efforts lack completeness in one way or the other. An analytical model for RLP has been presented in [5] that is integrated with a very simplistic model for TCP that lacks much of the complexities of TCP. Also, these models have been developed for i.i.d. (independent and identically distributed) frame errors only and developing similar models for correlated block errors is a complex process and simulations are the only means to address the need for an accurate model. On the other hand, in another approach [4, 7], an extensive model for TCP is used but the link-layer details are simplified by means of introducing delays of desired durations at desired intervals. Such an approach is an oversimplification of delay introduced due to error recovery through link-layer retransmissions and lacks analysis for a given frame error rate (FER)

with specific fading-induced correlation structure. Other approaches like [6, 8] give insufficient details of implementation rendering them irreproducible for other researchers. Previous works in this area have thus been marked by oversimplistic models at either the transport or the link layer.

In this study, we have developed a simulation tool for Code Division Multiple Access-2000 (cdma2000) data network in widely used Network Simulator (*ns2*)[9] simulator. Our model[10] involves elaborate implementation of protocols like RLP, Point to Point Protocol (PPP) and their integration with physical layer frame error models and transport layer protocols like TCP. In the following chapters we will cover these protocols and discuss their implementations. Our extension to the simulator helps in an accurate and broader analysis of additional delays induced over wireless links due to link layer error recovery. In this thesis, one of our main objectives is to produce an extensive, detailed analysis of delay variability due to link-layer retransmissions for various values of FERs with varying correlation structure and its impact on performance of upper layer protocols.

While the losses can largely be mitigated by suitable link layer retransmission mechanisms, limited radio frequency (RF) spectrum is still a problem that needs to be addressed effectively. One solution to effectively manage the limited resources is to dynamically share them among the users. This scheme calls for assignment of additional data rate channels to mobile users for specific durations based on users' demand, radio conditions and data backlog.

TCP[11] is the most widely used transport layer protocol. However, TCP performs best under conditions that do not apply to wireless networks. One such problem is when the bandwidth of one of the links is changed abruptly due to allocation of additional channels and a TCP sender begins to receive acknowledgements (ACKs) at a faster pace (sometimes referred to as *ACK compression*) and in response that, it puts new data packets into the network at a rate that

might be excessive for some other link in the network, which leads to loss of many packets, possibly an amount of data covering the sender's entire window might be lost. This scenario for cdma2000[12] networks is shown in figure 1.1, where a mobile user has a TCP connection with a fixed host in the internet. The radio link beyond the base-station controller(BSC) is a variable rate link that might cause the TCP sender to inject more data into the network than it can bear. The new bottleneck can shift to any of the network nodes in the path of TCP connection. However since BSC is the terminal point where all mobiles' data flows split into various link buffers and it serves all the mobiles under it, it is most likely to be the new bottleneck. Thus, because of radio link's rate variability, it might be possible that bottlenecks keep on shifting between the radio link and one of the intermediate shared buffers, e.g. BSC's input buffer.

Congestion control in networks has traditionally been assumed to be required because of greater number of users joining to share the network's resources. In this work, we introduce a new phenomenon of network congestion due to variable bandwidth radio links, with a fixed number of users, and we show that it is a more severe source of congestion. The reason being that a new user always begins with a window size of one packet and then increases it exponentially, but even then it never injects packets into the network at a rate significantly larger than that of the bottleneck. On the other hand, in a variable rate scenario, due to ACK compression, the TCP sender, even in a steady congestion avoidance phase, might be fooled into believing that greater bandwidth is available in the whole network and begins to send at that rate and, because of its large window size, it keeps on putting packets into the network until the highly undesirable events of multiple packets drops at one of intermediate routers and a possible prolonged timeout afterwards are over. These issues demand special attention to the problem of congestion due to bandwidth oscillations which, together with delay variability

due to channel errors, forms the underlying motivation for this thesis.

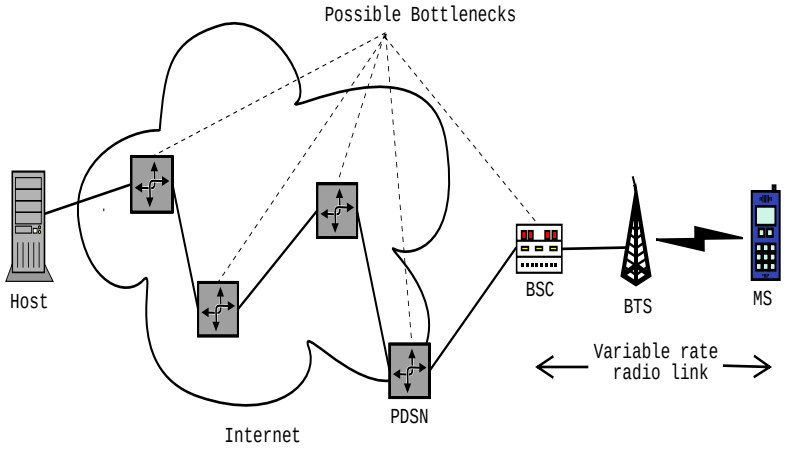


Figure 1.1: A network view of a mobile's TCP flow.

1.2 Objective of the Thesis

This thesis presents an evaluation of cdma2000 data networks for issues related to performance of the system. The specific objectives of this thesis are:

- Developing a simulation environment of cdma2000's link layer protocol, RLP, in *ns2* simulator using C, C++ and Tool Command Language (Tcl).
- Validation of developed model by comparisons with other existing simplistic analytical models.
- Simulation-based analysis of delay variability due to wireless losses under varying degrees of error rates and correlation and study the following issues:
  - Delay behavior over wireless links.
  - Data rate reduction.
  - Residual error rate.
- Evaluation of congestion problem with specific focus on the following:

- Analyzing the problem in some suitable mathematical form with, if needed, some simplifying assumptions.
- Establishing the problem by means of simulations.
- Comparing with equivalent congestion problem in wired domain.
- Analysis for typical load scenarios.
- Analysis of impact of system configuration on the extent of problem.

1.3 Thesis Outline

Rest of the thesis is organized as follows:

Chapter 2: Presents a brief overview of cdma2000 data system. Brief introduction of various components that include protocols, channel rates and buffer management is presented.

Chapter 3: Describes the simulation tools developed for the purpose of this study.

Chapter 4: Presents an exhaustive analysis of loss recovery through local retransmissions. This analysis is presented at various levels of protocol stack. A scheme for channel allocations based on the delay models is also presented.

Chapter 5: Presents the congestion scenario due to bandwidth variations. The problem is first modeled using continuous fluid model and compared against simulation results. Additionally, simulation results for multiple user cases are presented for a range of setup parameters.

Chapter 6: Finally, a discussion of results, conclusions and possible working directions for future work are presented.

1.4 Thesis Contributions

The following are the main contributions of the thesis:

- Detailed implementation of link-layer protocol RLP in simulation environment.

- Degree of time-correlation of errors and physical layer data rate, in addition to mean error rates, are shown to have profound impact on performance of wireless networks.

- It is observed that for modest levels of FERs with little correlation between errors, frame errors are effectively handled by link-layer error recovery mechanisms. We further observe that a smaller level of average FER with high degree of correlation produces similar throughputs as a high value of average FER with little correlation. We further derive conditions under which a high data rate supplemental channel may be assigned to a mobile based on its current FER and its tolerance for delay variability.

- Technique for estimating residual error rates in faded scenarios is presented and reduction in data rates for various channel rates is exemplified. Previous claims of RLP being able to bring down error rate below 1% based on i.i.d. assumptions are shown to be incorrect for correlated cases.

- An analytical model for losses due to bandwidth changes is presented and verified by means of simulations, which quantifies the dependence of losses on sudden bandwidth switching on higher switched rate, current window size, propagation delay etc.

- Congestion problem is examined under following test cases:

  - Worst case scenarios for simultaneous switching to higher data rates shows prolonged dead periods of system under-utilization.

  - Comparison with equivalent congestion scenario in wired networks with variable number of users shows that bandwidth oscillation is a greater

source of congestion than a change in the number of users.

- Scenarios with gradual increase in data rates for mobiles show that, for rise times not significantly larger than round-trip propagation delays, the problem persists.
- Typical load scenarios for bandwidth allocations shows that, for certain configurations, overall system utilization can fall to 75%.

Chapter 2

Data Services in cdma2000 Networks

In this chapter we present details of the cdma2000 networks that are essential for our studies. The cdma2000's single carrier radio transmission technology (cdma2000-1X RTT) [12] standard is a third-generation wireless specification that is tailored for supporting the data services together with voice services. This standard evolved from earlier second generation (2G) standards, IS-95 and IS-95B. cdma2000 differs from the earlier specifications in terms of greater data rates of supplemental channels for data services, better closed loop power control in both downlink and uplink and faster allocation/de-allocation of supplemental channels. The standards organization third generation partnership programme 2 (3GPP2) is in the process of developing new standards based on proposals for new evolutions of cdma2000 standards, namely the EVDV (Evolution, Data and Voice)[3] and EV-DO (Evolution, Data Optimized)[13]. The latter two can work on core 1X-RTT infrastructure with some changes that are aimed at providing data services with even greater bandwidth needs like voice over IP (VoIP) and multimedia.

The protocol stack for data services in 3G-1X system is shown in figure 2.1. The key objective for reliable data services is to minimize the impact of RF errors as much as possible. In wired networks where link losses occur rarely, loss recovery can easily be done by transport layer protocols, but in wireless networks, RF errors

invariably occur quite frequently and some link layer mechanism for mitigating these losses is desirable. For this purpose, link-layer recovery mechanisms such as RLP are used. The basic idea of these protocols is to hide wireless losses from upper layer protocols like TCP. Figure 2.2 shows an example of sending a TCP packet and receiving its ACK for a TCP connection, with the two end-points being a host in external internet and a mobile, that includes a lossy wireless link between base-station (BTS) and mobile (MS) and an underlying recovery mechanism at link-layer between the base-station controller (BSC) and MSs. We now proceed on to an elaborate description of the protocols and network layers that are of specific relevance to this work.

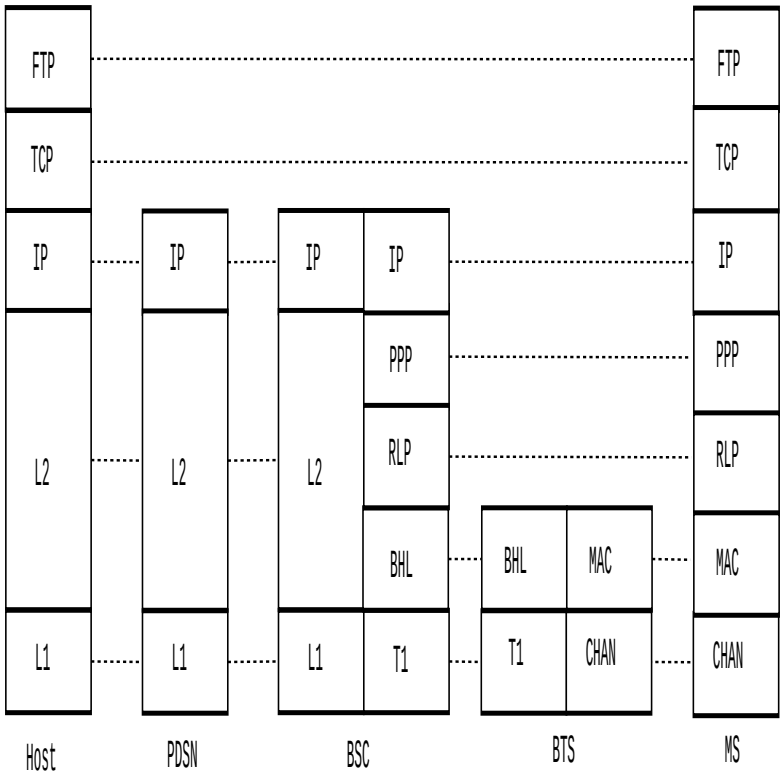


Figure 2.1: Protocol stack used in cdma2000 data network. PDSN is packet data service node and BHL is back-haul link protocol over T1 link.

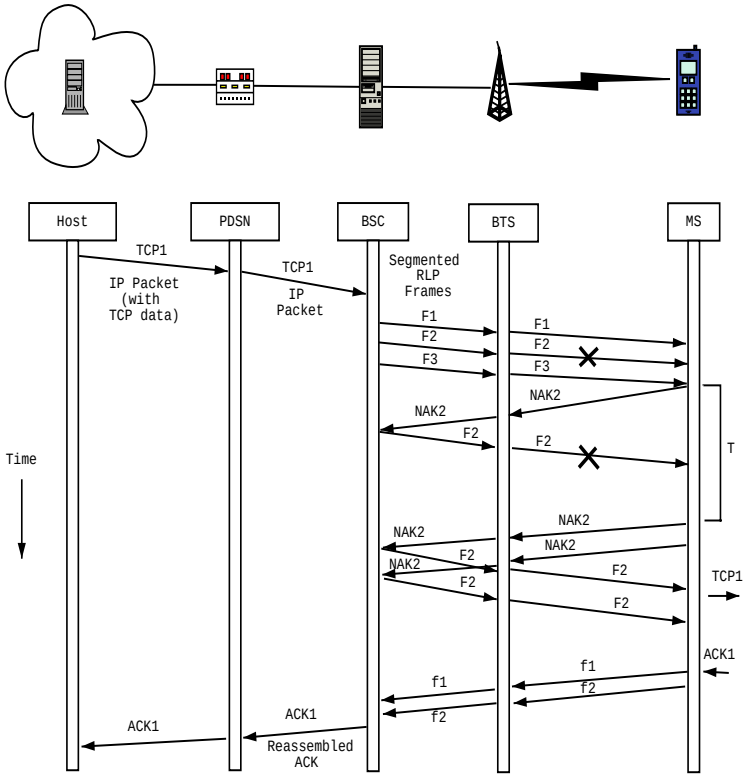


Figure 2.2: An example of link-layer error-recovery through RLP retransmissions.

2.1 Physical Layer

The 3G-1X system based on IS-2000 standard uses a single-carrier 1.25 MHz channel. The physical layer channels are pilot channel, access channel, fundamental channel, sync channel, control channel, supplemental channel, transmit diversity channels etc. A detailed description of these can be found in [12]. The data services, we are focusing upon are provided on top of the fundamental channel (FCH), or a supplemental channel (SCH). Every active mobile session has one fundamental channel and additionally a SCH of one of the rates can be allocated to it based on the demand of the particular mobile and availability of resources. SCH allocations are made for certain durations (20ms - 5.2s) based on some scheme such as *finite-burst* mode [16]. Table 2.1 shows the various data rates of SCHs.

Table 2.1: Allowed rate in kb/s for SCH assignments.

Rate	Rate Set 1 (RS1)	Rate Set 2(RS2)
1X	9.6	14.4
2X	19.2	28.8
4X	38.4	57.6
8X	76.8	115.2
16X	153.6	230.4

2.2 Radio Link Protocol (RLP)

In order to mitigate the losses over the wireless link, several link-layer protocols are used to recover the lost data. For instance, cdma2000 systems use the Radio Link Protocol 3 (RLP3) [1] to overcome the losses. RLP performs the following two functions.

Fragmentation and Assembly

Upon receiving an IP (Internet Protocol) packet, RLP entity first puts it in the *new-data* buffer and after the packet leaves the queue, it is fragmented into RLP frames of size corresponding to the current data rate and puts other information viz. frame sequence number, L_SEQ, in its 5-byte header. The frame size is chosen in a manner so as to transmit the frame in a 20ms slot for current value of available data rate. Assuming that physical layer rate remains constant during the transmission of an IP packet, the fragmentation mechanism, for a given raw physical layer data rate, R_{phy} kb/s, slot duration for frame transmission, $slot$, frame header of hdr_len bytes, and an IP packet of size, S_{ip} bytes, will generate $N_{rlp}(R_{phy}, S_{ip})$ RLP frames of size S_{rlp} bytes each where,

$$N_{rlp}(R_{phy}, S_{ip}) = \lceil (S_{ip}/(((R_{phy} \cdot slot)/8) - hdr\_len)) \rceil \tag{2.1}$$

$$S_{rlp} = (R_{phy} \cdot slot)/8 \tag{2.2}$$

In case there is a waiting IP packet in *new-data* buffer, part of its data is put on the vacant space on the payload of last 20ms RLP frame of current IP packet.

Frame size is strictly controlled by data rate and any change in it is immediately applied in determining the size of future frames. After transmitting each frame, the RLP sender puts the frame in a *retransmission* buffer, so that it can send the frame again in case the receiver demands so.

On the receiver RLP, the received RLP frames are put in a *resequencing* buffer until all the outstanding frames are received. The frames that are received in sequence are passed on to higher layer and in case a frame is detected to be lost, automatic repeat request (ARQ) mechanism as explained next is used. If ARQ mechanism is unable to recover the lost frame, the link-layer passes the available data with holes to higher layers.

Automatic Repeat Request (ARQ)

The ARQ mechanism employed in cdma2000 data networks is of selective repeat type. The RLP receiver does not acknowledge correctly received frames, instead it only requests those frames that it detects to be lost or finds to be in error. The RLP receiver maintains two variables, $L\_V(N)$ and $L\_V(R)$. The first one is the sequence number of frame needed for sequential delivery to upper layers and other one is the next frame expected by the receiver. Whenever the RLP receiver detects that incoming frame's sequence number, L_SEQ, is greater than $L\_V(R)$, it creates a NAK_LIST (NAK stands for negative acknowledgement) entry for each of the missing frames. A NAK_LIST keeps record of all the lost frames. Other setup parameters for link-layer are - number of NAK rounds, n, retransmit timer, T, and number of NAK frames to be sent in i-th NAK round, $NC[i]$. Upon receipt of every frame, NAK_LIST is updated by removing the entry for an outstanding frame that is received correctly on retransmission and sending NAK control frames for those missing frames whose retransmit timer, T, has expired for a particular round. If RLP is unable to recover the frame after $n-1$ rounds, it starts the n-th NAK round and starts an abort timer, A, to wait

for the missing frame after which it passes the available data to the higher layers. Note that since all timers are frame counters, the specifications stipulate sending $IDLE$ frames when the sender has no data to send so that receiver does not have to wait indefinitely for its timers to expire.

In brief, RLP can be thought of as a mechanism that provides added reliability over an otherwise lossy wireless link but comes at an additional cost of variable propagation delays over the wireless link.

2.3 Point-to-Point Protocol (PPP)

Point to Point Protocol(PPP) forms a data link between the BSC and MSs. Since IP packets are fragmented to RLP frames at BSC, PPP takes over the routing of data frames to its destination at BSC. PPP consists of variety of other features namely, providing a way of encapsulating the IP datagrams on a serial link and a link control protocol (LCP) for establishing, configuring and testing the link connection.

2.4 Transmission Control Protocol (TCP)

TCP is the most widely-used transport-layer protocol that provides reliable transport of data between two end-points. It has its own elaborate mechanism for sensing available bandwidth in the network path. It does so by adjusting the window sizes (maximum amount of data that can remain unacknowledged) based on receipt of acknowledgements. *Slow-start* and *congestion avoidance* are the two algorithms that dictate the sending rate of a sender. Its loss recovery mechanisms, *fast retransmit* and *fast recovery* are designed to effectively recover the lost segments after losses. Besides, TCP uses various timers for retransmission and maintaining the connections. TCP has evolved over years of discussions and proposals in Internet Engineering Task Force (IETF) and many versions of TCP

viz. Tahoe, Reno, Vegas, Newreno, SACK(selective acknowledgement) etc, which differ from each other in their congestion control algorithms, can be found in existing implementations. Most of the data transfer in today's internet takes place over TCP and it has become the *de-facto* transport protocol in existing internet. Several applications like File Transfer Protocol (FTP) can be run over TCP. A detailed description of these protocols can be found in [11].

So far, we have looked at the various protocols and features that are part of cdma2000 data services architecture. On one hand, we have certain features specific to wireless networks like RF losses and bandwidth variability and on the other, we have protocols like TCP that were designed with underlying assumptions of nearly fixed propagation delays and bandwidths. In the following chapters, we will see how these interact with each other and will analyze them from a systems perspective.

Chapter 3

Simulation Setup

Even though 3GPP2 had standardized the RLP in December 1999, no open implementations of it in discrete event simulators like *ns2*[9] or OPNET[14], that could be used to analyze the performance, exist to our knowledge. To meet this need for an accurate simulation tool for RLP, an extensive implementation of RLP's ARQ mechanism was made as part of this work in *ns2*. This implementation involves a new `RlpAgent` class for performing the required ARQ functions. Additionally, `PppAgent` is also implemented that sets up a PPP link between two nodes in *ns2*.

As a design choice, *ns2* was chosen over other simulators due to its open source nature with all the source code available for developing a good understanding. As well, *ns2* has implementations for most of the networking protocols of TCP/IP suite in it together with router algorithms like Random Early Discard (RED) that are of specific interest to us. As a result, only those modules that do not exist in *ns2* needed to be implemented, and even in developing these parts, significant use of *ns2*'s existing hierarchy was made. This leads to a compact tool that can be easily integrated with the huge library of other *ns2* modules. The operation of the agent implementation of RLP is shown in figure 3.1. Essentially, this agent captures an IP packet from a link or an upper layer agent and segments it into frames and transmits them. On the receiver side, it reassembles the IP packet from the constituent frames. The error models available in *ns2*'s [9] library are used

to model the channel errors. Detailed descriptions of the classes `RlpAgent` and `PppAgent` and the way of configuring them for simulations is shown in appendix. We will cover some of the features of the implementation in this chapter.

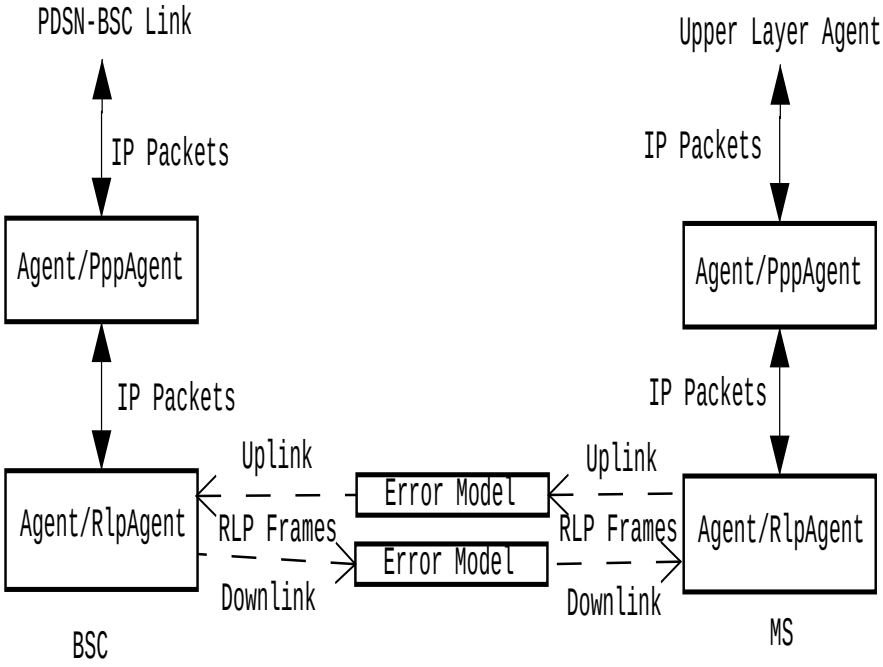


Figure 3.1: Implementation of RLP agent in *ns2*

3.1 Physical Layer

The physical layer of cdma2000 networks is primarily composed of two parts:

Variable Rate Link: This is modeled by segmenting the IP packets using the size governed by equation 2.2 and transmitting each frame in the slot duration whose default value is 20ms. The actual rate of the links between the two RLP agents is chosen arbitrarily large so that the radio link's rate is only governed by the variable `rlp_rate_` (see Appendix). The default value of this rate is set to base rate of 9.6 kb/s for RS1.

RF Losses: The errors on the wireless links can be in either scattered form or in form of sustained blocks due to fading. The former can be easily modeled using an i.i.d. error model of a given frame error rate, say ϵ. For latter, the

correlation exhibited by the Rayleigh fading can be modeled using a first-order two-state Markov chain[15] as shown in figure 3.2, wherein the channel alternates between a *good* and a *bad* state with a transition matrix,

$$S_T = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}. \tag{3.1}$$

During the *good* state, all the frames are sent correctly over the channel and are all in error during a *bad* state. The mean FER for such a correlated model is $\epsilon = p/(p+q)$, and the mean residence times in good and bad states are, $\tau_{good} = 1/p$ and $\tau_{bad} = 1/q$, respectively. The advantage of using this model is that there exists a direct mapping of fading margin, F, and normalized Doppler frequency, $f_D T$ (where f_D is the maximum Doppler shift and T is the duration of packet slot) to the parameters, p and q of the Markov chain. As an example, when $f_D T$ is small (e.g. 0.01) the fading level is very high and p and q have lower values and converse is true for a higher value of $f_D T$ (e.g. 1.0) when errors are nearly i.i.d. This scheme allows for simulation of any desired level of correlation of errors in the forward and reverse links of a particular mobile. In our simulations, we have put the error-model in the downlink only. Similar results can be derived by putting an equivalent error-model in the uplink.

3.2 Frame Types

For this implementation, several new frame (packet) types were defined:

Data frame: This corresponds to new data frames created after segmentation of IP frames. The size of RLP data frame is governed by equation 2.2. Associated with each RLP data frame is a RLP header that has following fields: `l_seq` for the sequence number of the frame, and `pkt_size_` for the amount of data carried by the frame.

NAK frame: This is a control frame that is generated by the receiver RLP

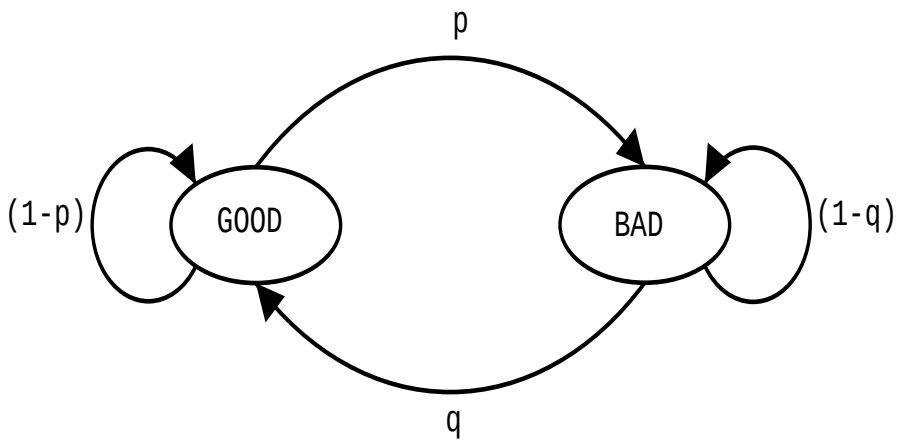


Figure 3.2: Two-state Markov chain for simulating time-correlated frame errors (p and q are state transition probabilities from *good* state to *bad* state and vice versa).

in response to detection of a missing frame. This contains the field `l_seq` for the sequence number of the missing frame.

Idle Frame: These frames are sent by an RLP entity when it has no data frame to send. This is necessary for incrementing the frame counters on the receiver side so that receiver does not need to wait for any outstanding frames indefinitely after the sender is done with the transmission of data frames.

3.3 RLP Operation

The operation of RLP protocols was explained in previous chapter. Figures 3.3 and 3.4 show the mapping of these mechanisms on the functions and attributes in our implementations. The triggering functions and conditions for state transitions are marked. These diagrams were developed during the design phase of the software and later implemented in *ns2*. Each RLP entity spends its lifetime in one of the possible states as shown. Although, in reality, an RLP entity exists both as a sender and a receiver, we have split the state diagrams for the ease of understanding.

RLP Sender: As shown in figure 3.3, the RLP sender keeps on sending an *IDLE*

frame at the expiry of its timer, `rlp_timer_`, when it has no data in its IP buffer. After each transmission, the RLP entity sets its timer equal to the duration of `slot` for frame transmission. When it receives an IP packet, it begins segmenting the IP packet into frames and transmits each of them in a single slot duration. Also, in each slot the RLP sender prepares the frame to be sent in next round with a size corresponding to its current data rate. In this mode, the RLP sender is said to be in TRANSMISSION state. In case the sender receives a request for a frame to be sent again, it enters the RETRANSMISSION state by sending the requested frames in earliest slots possible. When the sender does not have any IP packet or NAK frame it goes back to the IDLE state.

RLP Receiver: The RLP receiver spends bulk of its lifetime in the IDLE state when it is either receiving the IDLE frames or original data frames in order. The only time it needs to come into action is when it detects a frame to be in error and enters the *receiving out-of-order* state. At that point, it creates an entry in the NAK list for the frame in error and requests retransmission of the erroneous frame for some limited number of times. The receiver also reassembles all the RLP frames to generate an IP packet and passes it on to higher layers.

3.4 Model Verification

To validate our model, several tests were performed against published results based on simplistic analytical models in earlier works. In one of these comparisons, same settings as in [5] have been used with, retransmission settings $\{1, 2, 3\}$, $q = 0.02$, $R = 5$, $T = 13$, $B = 1000$, physical layer rate = 9.6 kb/s. As can be seen in Fig. 3.5(a) and 3.5(b), the plots of RLP frame delay and TCP throughputs versus FER obtained using the developed simulation tool and the analytical model introduced in [5] are in close agreement for lower values of FER. The divergence from the analytical model in high FER range (beyond 25 percent) can be attributed to

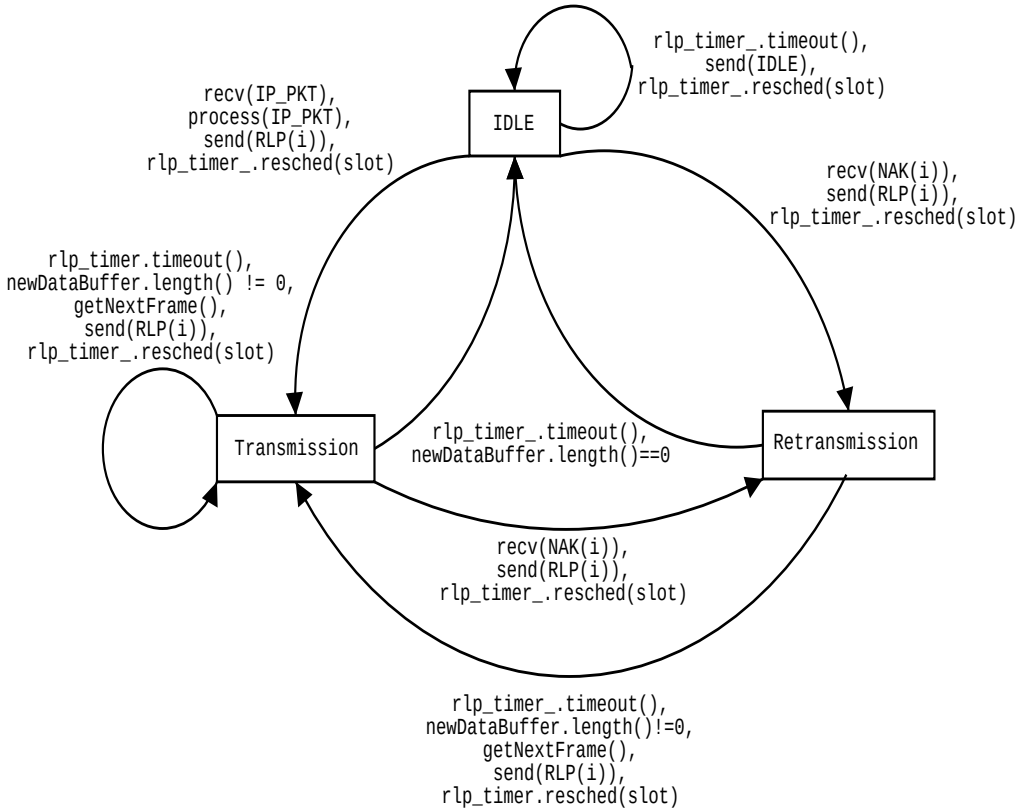


Figure 3.3: State transition diagram for an RLP sender

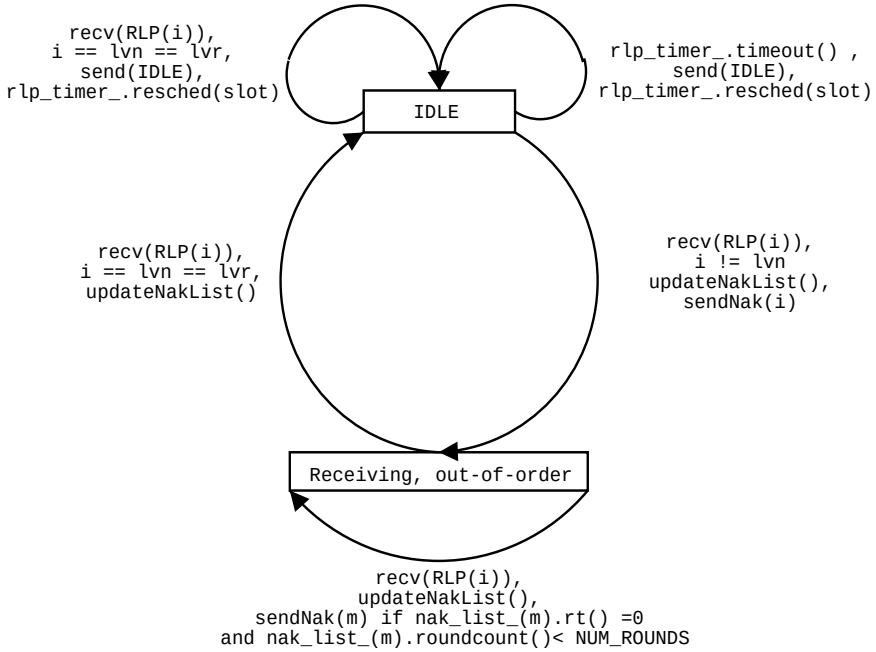
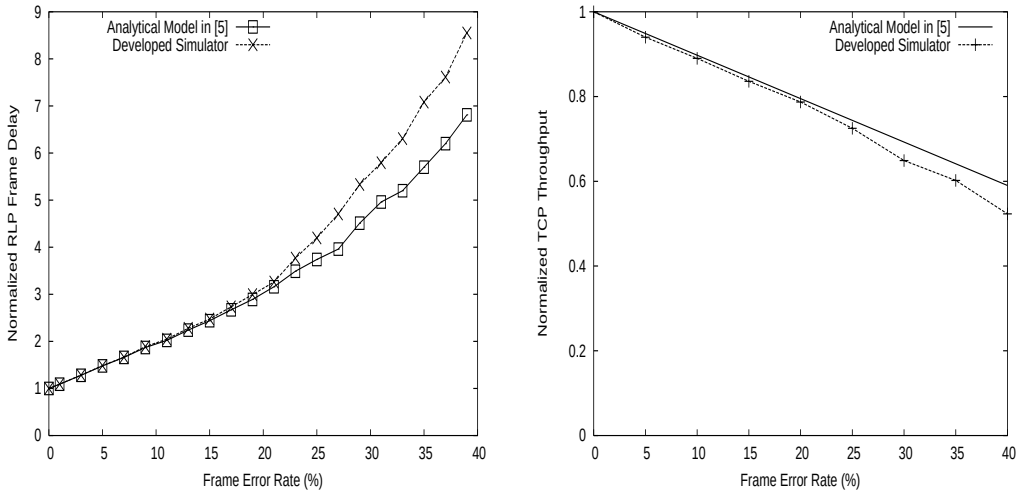


Figure 3.4: State transition diagram for an RLP receiver

some conceptual differences between the analytical model and our simulator:

- The delay associated with detection of frame loss, i.e. $D(F_1^0)$, can be greater



(a) RLP frame delay versus FER (b) TCP throughput versus FER

Figure 3.5: Comparison of analytical model[5] and developed simulator.

than as calculated in [5] because of greater queueing delays for subsequent successful future frame due to overdue *retrasmit* packets (that have greater priority than our new future frame) on the sender side. This effect is greater at higher FERs.

- A fixed value of round-trip time for sending a NAK and receiving the corresponding reply, R, has been taken in [5]. This approach is not applicable at high FERs. First, a NAK may not be always immediately sent due to overdue NAKs to be sent in the NAK queue on the receiver side. Also, as before, the retransmit frame for a received NAK may not always be sent immediately because of other waiting retransmit packets. Averaging out the value of R is impractical at high FERs as R is related to FER level and increases sharply at high FERs, so a constant value of R introduces inaccuracy.

As we can see that at high FERs, the average RLP frame delay calculated by our simulator is higher than the one predicted by the analytical model as the former takes into account the factors mentioned above. This is where the simulator excels over analytical model. So, even though the analytical model provides rich insight

into the behavior at low FERs, but at high FERs, simulations are the only way of examining the accurate RLP behavior.

In this chapter we covered some of the details of the implementation of cdma2000 data network that was made as part of this work. In the following chapters we will be making extensive use of this simulator to examine the performance in presence of wireless errors and rate variations.

Chapter 4

Impact of Wireless Errors

4.1 Introduction

The link layer in cdma2000 data networks uses local retransmission mechanisms for providing added reliability over the error-prone wireless link. The disadvantage of this approach is that the propagation delays over the link can become highly variable, sometimes in form of sharp spikes for highly correlated errors, and results in variability in round trip propagation delays as seen by end TCP senders, resulting in retransmission timeout (RTO) inflation, timeouts and retransmissions. In this chapter, we present a detailed analysis of these effects of RF errors for a variety of error models with different mean FER and correlation structure.

Table 4.1 shows the simulation parameters used for analyzing the performance of cdma2000 data network in this chapter. A single mobile user downloading a large file using FTP application is used for this purpose. The simulations are being performed using a cdma2000 data module in ns2[9] simulator, which was described in chapter 3.

4.2 Results and Analysis

The impact of link-layer retransmissions can be understood at various levels. As we progressively go up the protocol stack, it manifests in different forms. Several

Description	Symbol	Value(s)
Application		FTP
TCP version		NewReno
TCP Segment Size	S_{tcp}	960 Bytes
Receiver's buffer size		32KB
TCP Segments per Ack		1
IP Packet Size	S_{ip}	1000 bytes
RLP NAK rounds	n	3
NAKs in each round	NC	{1, 2, 3}
Retransmit Timer	T	13
Abort Timer	A	13
One way radio delay (BSC to MS)	d_{radio}	40ms
One way wired delay (Host to BSC)	d_{wired}	30ms
Mobiles' data rates	R_{phy}	9.6 kb/s- 153.6 kb/s
Error Models:		
(a) i.i.d. error rate	ϵ	0% - 40%
(b) correlated model	p	0.02, 0.05, 0.1
	q	0.08, 0.2, 0.4

Table 4.1: Simulation parameters

major problems caused due to retransmissions are discussed in next sections.

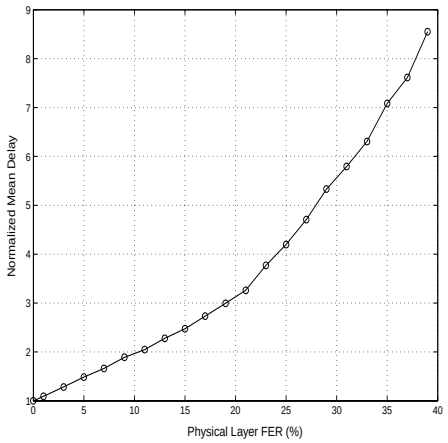
4.2.1 Link Layer

As described earlier, error recovery in wireless networks is mainly done at link layer using protocols like RLP [1]. This layer locally hides the losses from upper layers using retransmissions. These retransmissions introduce several performance-related issues.

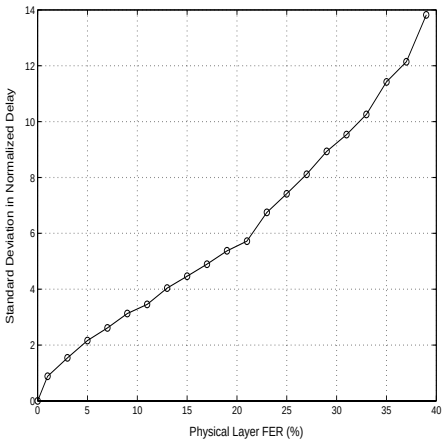
Delay Behavior

Figures 4.1(a), 4.1(b) show the normalized values of mean delays (as a ratio of transmission time of single frame, i.e. 20ms) for various values of frame error rates. It can be seen that both the average delay and deviation in it increase with increasing frame error rates, producing greater variability in delays. Previous

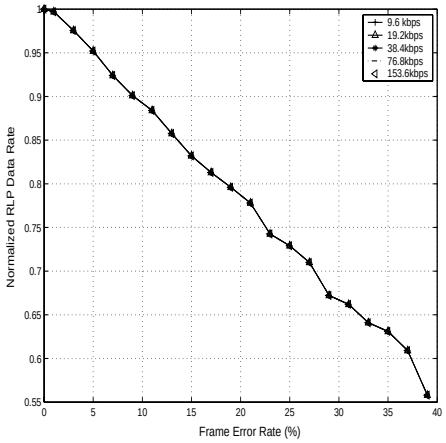
work [5] focussed on mainly the mean values of delay. However it has to be clearly understood that increase in delay is not a major problem for higher layer protocols like TCP, instead it is the high variability in delay (Figure 4.1(b)) that produces poor performance. As can be seen that for higher FERs, the effective delay over the link can increase many times more than its nominal value. For lower FERs, the variability is low and hence the moderate jitter can easily be absorbed in aggregate transmission.



(a) Normalized mean RLP delay due to various FERs (i.i.d.)



(b) Standard deviation in normalized delay



(c) Effective RLP data rate due to re-transmissions

Figure 4.1: Impact of physical layer frame errors (i.i.d) at link layer.

From Eqn. 3.1, it can be seen that lower values of state transition probabilities,

p and q, produce higher levels of correlation. Fig. 4.2 shows the normalized frame delay for FERs of varying correlations. It can be noted that correlated frame errors produce greater spikes in delays than i.i.d. errors of the same average FER, which indicates at greater degradation of performance due to fading resulting in correlated errors. Also note that extreme levels of correlation can lead to loss of a frame after all the NAK rounds and link layer is unable to retrieve the frame which will eventually lead to loss of the IP packet.

In general, the total normalized delay experienced by an RLP frame, d_{total}, is comprised of three parts:

- Nominal delay corresponding to transmission of frame and latency of link, $d_{nom} = d_{trans} + d_{radio}$. The transmission delay, d_{trans}, is simply the transmission time of frame and is the ratio of frame size, S_{rlp}, and physical layer rate, R_{phy}. d_{radio} is the latency over wireless link.

- Normalized detection delay, d_{det}, representing the delay in detection of loss of a frame. Suppose that a frame with $L\_SEQ = m$ is lost and the previous frame $(m-1)$ was sent successfully. The frame number m will be detected to be lost when one of the future frames $m+1$, $m+2$,... , is received successfully. Until that instant the receiver would not be able to detect the loss. For an i.i.d. FER model, this delay would be,

$$d_{det} = (1-\epsilon) + 2\epsilon(1-\epsilon) + 3\epsilon^2(1-\epsilon)\ldots = 1/(1-\epsilon). \tag{4.1}$$

  For correlated model, the delay corresponds to the stay in *bad* state of the channel and can be found by replacing ϵ by $(1-q)$ in the above relation and turns out be the mean residence time in *bad* state, τ_{bad}. It can be verified that detection delay is not bounded above as it can attain any high value, although with a diminishing tail probability.

- Normalized recovery delay, d_{recov}. This delay is corresponding to the time spent by the receiver in retrieving the frame after first detection of loss of the frame. It can be in between zero (corresponding to successful original transmission) and sum of all the *retransmission* timers of receiver RLP. Since, for each of the NAK round, the retransmission timer is set to retransmit timer, T, plus the number of NAKs sent in the round, this condition bounds the upper limit of recovery delay as,

$$d_{recov} \leq \sum_{i=1}^{n-1}(T + NC(i)) + A, \tag{4.2}$$

  where A is the abort timer. This means that, after detection of error in a frame, by the expiry of time equal to RHS in equation 4.2, the receiver will either be able to recover the lost frame or give up trying to doing so. A mean value of recovery delay is thus a function of FER structure and radio latency. For lower values of FERs, most of the lost frames will be recovered by first NAK round itself and hence the mean value of d_{recov} will be very low. However, for very high FERs, many frames will be recovered after several NAK rounds, some would not be received after all NAK rounds and the bound in equation 4.2 will be attained for these frames.

The total delay experienced by a frame is thus dominated by the time spent in detection of frame loss and subsequent recovery. In our simulation setup as shown in table 4.1, values of per-frame delays are shown in Fig. 4.2. Each round for this setup is of duration 13 units of frame transmission, so the upper bound on recovery delay for this setup using equation 4.2 comes out to 45 units. Fig. 4.2(a) shows the values of delays for an i.i.d. FER of 20%. It has three levels of delay spikes with the largest one being the least frequent so for this scenario almost all of the frames are recovered within first two rounds and the retransmission induced delays occur frequently (as shown by dense region created by delay spikes of first

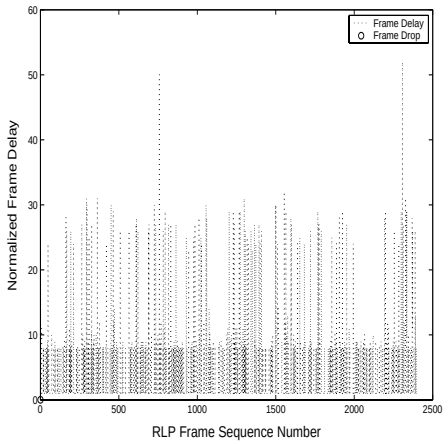
retransmission). Also, for this case, since the errors are well distributed, detection delay is negligible to few units and total maximum delay is thus in the vicinity of 50 units, close enough to the upper bound of 45 units as governed by equation 4.2. Figure 4.2(b), 4.2(c) show the delays for correlated case. Clearly, as the correlation increases, the loss-induced delays are more sharp in nature. The maximum delays are also very large, because with increasing correlation (i.e. decrease in p, q), the mean residence time in bad state is also very large, e.g. for a correlation model with $p = 0.02$, $q = 0.08$, τ_{bad} turns out to be 12.5 units, almost as much as the mean recovery delay. Fig. 4.2(d) shows the Cumulative Distribution Function (C.D.F.) for frame delays for various levels of correlation. As can be seen, the variability increases with correlation of errors.

This can be summarized by noting that with increasing delays, the mean delay and delay variability both increase, and as the correlation increases, the delays tend to be more sharp and greater because of longer delays in detection of losses due to block errors.

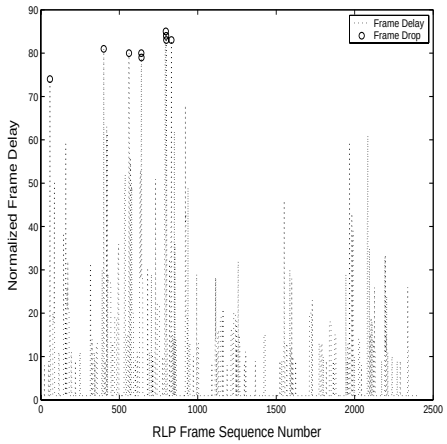
Residual Frame Error Rate

As described earlier, RLP will not try to recover a lost frame after all the NAK rounds are over. Thus, despite RLP's elaborate recovery mechanism, it is not entirely error-free and reliable. All it does is to reduce the probability of errors below a desired level. In [5], authors report that the usual $\{1, 2, 3\}$ retransmission scheme helps to overcome the errors and show that, for i.i.d. errors, the residual RLP FER, ϵ_{rlp}, is well within 1% for a maximum FER of 40%. This result can easily be obtained using an i.i.d. FER and noting that residual RLP FER corresponds to all NAK rounds being unsuccessful, i.e.,

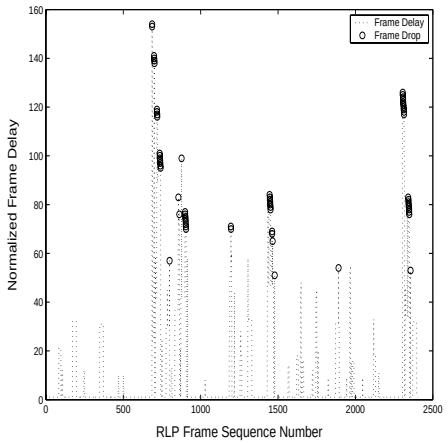
$$\epsilon_{rlp} = \epsilon^{1+\sum_{i=1}^{n} NC(i)}. \tag{4.3}$$



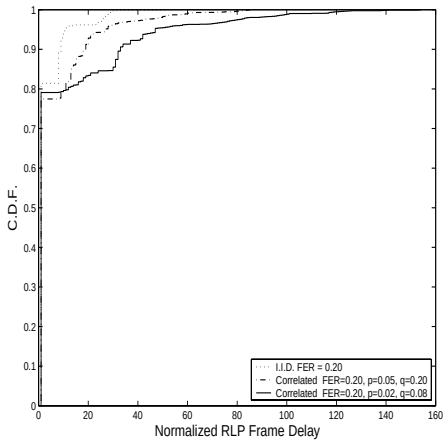
(a) i.i.d. FER, ϵ=0.20



(b) Correlated FER=0.20, p=0.05, q=0.20



(c) Correlated FER=0.20, p=0.02, q=0.08



(d) CDF of Frame Delays

Figure 4.2: Frame delays for various levels of correlation for a constant mean FER.

However, for correlated model, the residual FERs could be much higher because of sustained burst of errors. Fig. 4.2(b), 4.2(c) show this effect where, for a constant value of mean FER, as the correlation increases, the number of frames that could not be recovered after all NAK rounds increases and can not be neglected. In reality, residual FER for fading-induced correlated losses are difficult to be modeled analytically and simulations are the only means of getting an estimate. Table 4.2 shows the residual FER values for varying levels of correlation obtained using our simulation tool. It can be seen that for higher levels of correlation, RLP's retrans-

Table 4.2: Residual RLP FER for various levels of correlation and constant mean FER.

FER parameters	Residual FER (%)
i.i.d., $\epsilon = 0.02$	0.0013
correlated, $p = 0.1$, $q = 0.4$	0.015
correlated, $p = 0.05$, $q = 0.2$	0.862
correlated, $p = 0.03$, $q = 0.12$	2.312
correlated, $p = 0.02$, $q = 0.08$	5.251

mission scheme is unable to retrieve the lost frames and a residual FER of as high as nearly 5% can occur under such circumstances. This problem can be mitigated to an extent by using a more distributed retransmission setting, say $\{1, 1, 1, 1, 1, 1\}$ as in [6]. It is clear that high levels of correlation might lead to unsuitability of a particular retransmission setting and hence fading effects have to be adequately addressed while designing the network. Several approaches, including one that adaptively changes retransmission settings[17], have been proposed to mitigate this problem.

Reduction in Data Rate

An immediate consequence of retransmissions is reduction in effective data rate. Since, under error conditions, some of the frames need to be transmitted more than once, this leads to an additional overhead. Figure 4.1(c) shows the impact of retransmissions on the available data rate at link layer. It can be seen that at an i.i.d. FER of 40%, the effective data rate goes to almost half its rate while operating under error-free conditions. An example of a poorly designed retransmission scheme that significantly alters the effective data rate is one in which too many NAK frames are sent in initial rounds for low levels of FER. This illustrates that retransmission settings are of utmost importance in retrieving lost frames with minimal overhead.

We will conclude this section by commenting on some of the deficiencies in the modeling techniques employed for link layer protocols. In [5], authors have

modeled RLP and have ignored the additional queueing delays associated in detection of lost frames due to overdue *retransmit* frames. Also, in many modeling approaches, a constant value of round-trip time, R, in sending a NAK and receiving the corresponding reply has been taken. Such an approach fails to capture additional queueing delays for NAK frames on receiver side due to other NAKs on one hand and the queueing delays associated with *retransmit* frames at the sender. Therefore, averaging out a value of R is impractical as it varies a lot with FER level, and at higher FER levels, the queueing delays mentioned above need to be taken into consideration for an accurate analysis. Even though the analytical model provides rich insight into the behavior at low FERs; at high FERs, in our view, simulations are the only way of accurately examining RLP behavior. It is our belief that our simulation tool will be effective in meeting this need for a robust and accurate model.

4.2.2 IP Layer

At IP layer, apart from the error recovery, the delays are dependent on another factor - number of RLP frames per IP packet, $N_{rlp}(R_{phy}, S_{ip})$. Figure 4.3 shows the results at IP layer. It can be seen that at lower data rates, when each of the IP packet is segmented into large number of frames, the delay jitter due to link layer retransmissions are absorbed to a great extent and the effective IP packet delay is never more than twice what it would take for an IP packet to traverse through error-free wireless link (figure 4.3(a)). This is due to the fact that large number of RLP frames are created for each IP packet and recovery in case of a loss can be mostly done while the original transmission of an IP packet's frames is going on. Correlated errors of same average FER however are more harmful leading both to loss of IP packets and greater spikes in delay (figure 4.3(b)). Figure 4.3(c) shows that the delay jitter of link layer gets translated to delay spikes at IP layer if a

high rate of 153.6 kb/s is used. This is because of the fact that fewer RLP frames are generated for an IP packet which, in case of loss, are mostly recovered when the sender is done with original transmission of IP packet and recovery might only be done when several future frames have been transmitted. However, link layer as before in figure 4.3(a) is effective in recovering the losses and no packet drops occur. So, a high level of correlation of errors and a high data rate can trigger spikes in packet delay that is harmful for wireless network performance. Figure 4.3(d) illustrates that same level of physical layer of FER of 0.20 can trigger varying kinds of delay behavior at IP layer depending on the correlation structure and current data rate for the mobile.

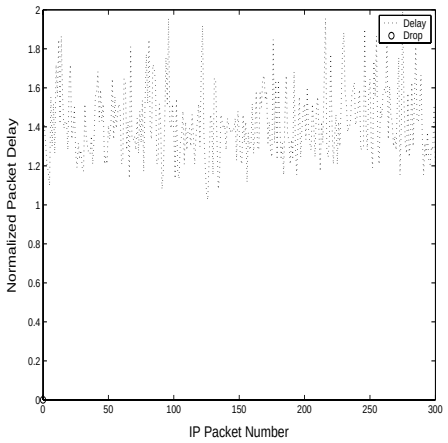
Although high data rates lead to greater delay variability, but a very low value of data rate is also not desirable as it increases the residual errors at link layer to those at IP layer, ϵ_{ip}. Note that residual IP layer error corresponds to a scenario when any of the constituent RLP frames could not be recovered at link layer, i.e.,

$$\epsilon_{ip} = 1 - (1 - \epsilon_{rlp})^{N_{rlp}(R_{phy}, S_{ip})} \approx N_{rlp}(R_{phy}, S_{ip}) \cdot \epsilon_{rlp}. \tag{4.4}$$

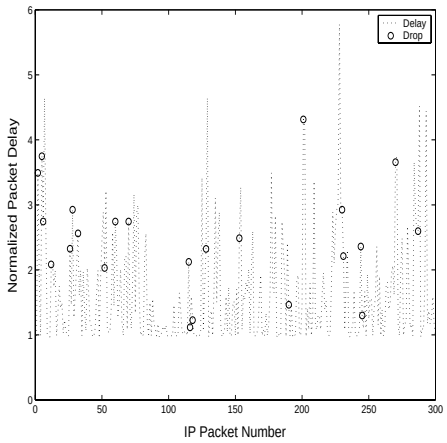
Since number of RLP frames per IP packet is large for lower rates, so residual IP layer error rate decreases with increasing data rates, R_{phy}. However, the magnification of residual errors at IP layer occurs only at extremely low data rates and for a properly designed system with negligible residual RLP errors, only the delay variation needs to be considered for better performance.

4.2.3 TCP Performance

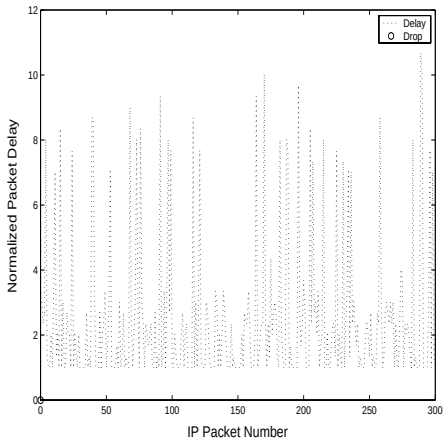
Figure 4.4(a) shows the TCP traces for an error-free link. Each point in the graph corresponds to the a sent TCP data segment or a received ACK at the TCP source. Figure 4.4(b) shows the decrease in throughput due to reduced effective link layer rate as exhibited in figure 4.1(c). However, the link layer is able to recover all of the lost frames and IP packet drops do not occur and the overall behavior is



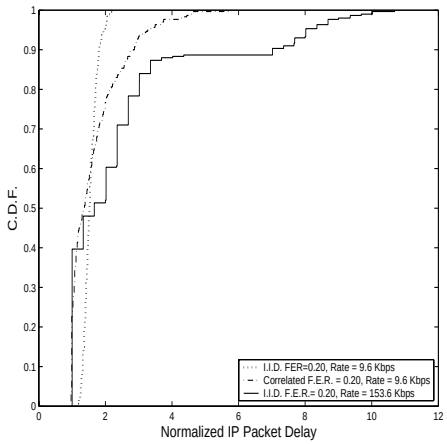
(a) i.i.d. FER $\epsilon = 0.20$, $R_{phy} = 9.6$Kbps.



(b) Correlated FER = 0.20, $p = 0.02$, $q = 0.08$, R_{phy}= 9.6Kbps.



(c) i.i.d. FER = 0.20, R_{phy}= 153.6Kbps.



(d) C.D.F. of Normalized IP Packet Delays.

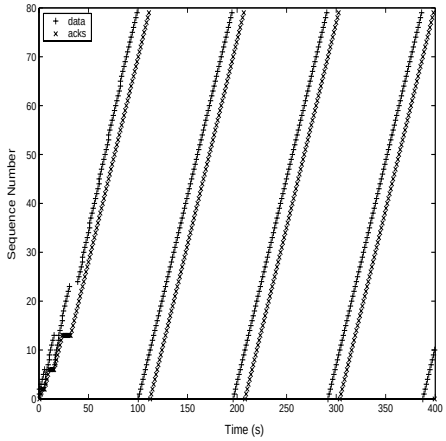
Figure 4.3: Normalized IP packet delays.

smooth. Figure 4.4(c) shows further decrease in throughput due to correlated losses at the same average FER. This is due to loss of packets due to failure of link layer to recover the lost frame after all NAK rounds. Figure 4.4(d) shows that at higher levels of i.i.d. FERs packet losses do occur and the recovery is also very slow due to inflation of RTT. It can be seen that an i.i.d. FER of 0.35 offers roughly the same throughput ($\sim$ 200 TCP segments in $400s$) as a correlated FER of 0.20 and it establishes that fading effects have to be considered while modeling the delay of wireless links.

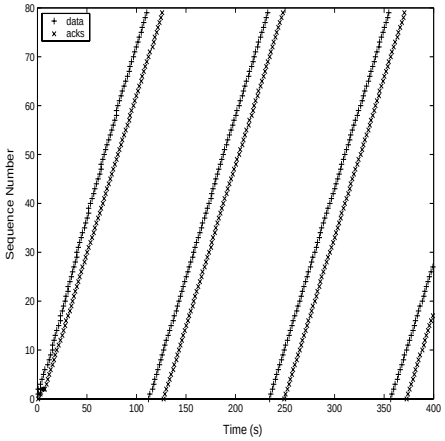
Figure 4.6 shows TCP behavior for a higher rate of 153.6 kb/s. At this high rate when fewer frames are generated for each IP packet, delay variability is high and a modest i.i.d. FER of 0.20 that did not produce significant reduction in throughput for lower rate of 9.6 kb/s (figures 4.4(a) and 4.4(b)) is enough to significantly alter the available throughput as shown in figures 4.6(a) and 4.6(b). The ACK arrival times are not regular and show large variations. As mentioned earlier, for such high data rates, the delay variation has spike-like behavior and can trigger timeouts and one such timeout occurs at around $25s$. Figure 4.6(c), 4.6(b) show further degradation when correlation and mean FERs are increased respectively.

Figure 4.5 shows the TCP throughputs for various levels of FER (i.i.d) at various data rates for mobiles. It can be clearly seen that at higher data rates, TCP throughput reduces significantly more than lower rates due to higher delay variability leading to poor performance of TCP's window mechanism. TCP can not handle sharp variations in delay over the wireless link and hence at higher FERs, decrease in throughputs is very natural. The other notable thing is that the impact of RF errors is more pronounced at higher rate for a given FER. The two important factors of FER level and data rate that impact the delay behavior of wireless links at the link layer as shown in figure 4.2(d) translate to lower TCP throughput at the transport layer.

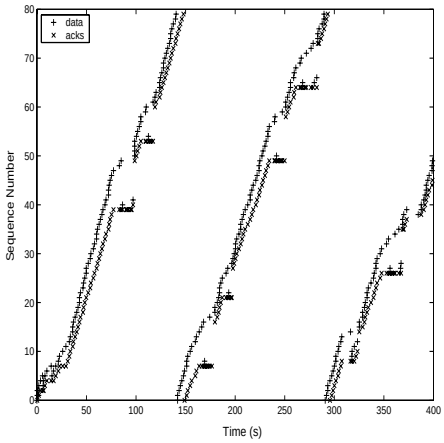
It can be clearly stated that TCP performance in presence of wireless losses and link-layer recovery is dependent on appropriate choice of design parameters such as retransmission settings, retransmission timer. Fewer NAKs in high error scenarios will produce greater residual FER and degrade performance and an excessive number of NAKs for a low error case will again degrade throughputs by triggering unnecessary retransmissions. Also, the retransmit timer, T, has to be appropriately chosen so that RLP receiver waits for adequate time before entering



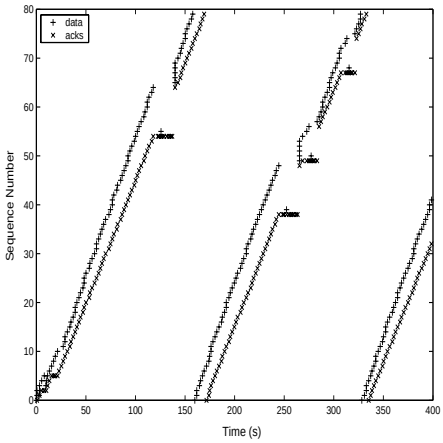
(a) $\epsilon = 0$



(b) i.i.d. FER = 0.20



(c) Correlated FER = 0.20,$p = 0.05$,$q = 0.20$



(d) i.i.d. FER = 0.35

Figure 4.4: TCP behavior for $R_{phy} = 9.6$ kb/s (sequence numbers are in *modulo*-80 fashion).

the next NAK round.

A significant consideration in designing 3G wireless data system should be devoted towards SCH allocation to mobiles depending on wireless conditions. In the following section, we discuss one possible scheme for making channel allocation decisions.

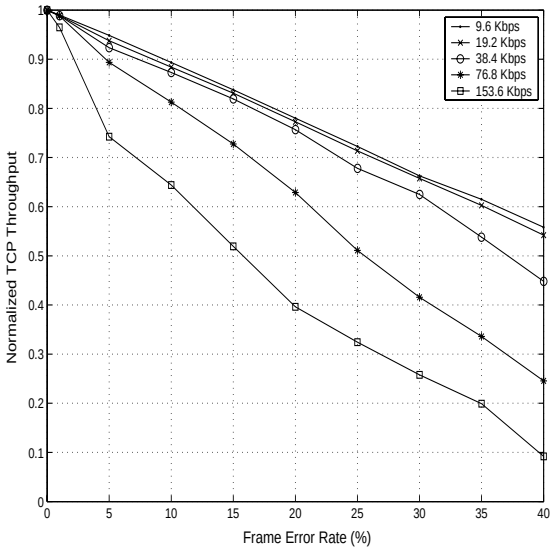
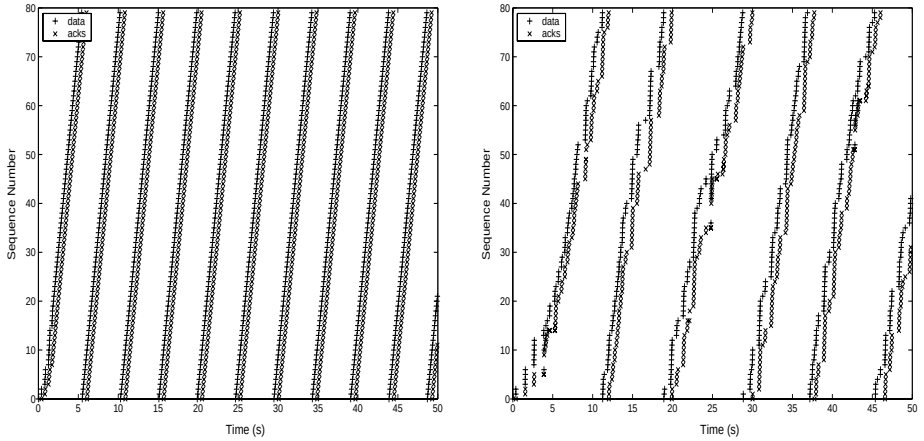


Figure 4.5: TCP throughputs for various values of raw link rates and physical layer FERs (i.i.d.)

4.3 Supplemental Channel Allocation Decisions

It has been shown earlier that a high data rate SCH assigned to a mobile in poor wireless conditions significantly reduces the throughputs. So, the BSC should be able to make a decision as to how much resources could be allocated to the mobiles so that the scarce resources are not unnecessarily wasted for a mobile in poor radio conditions and the overall system performance is not seriously degraded. A straightforward way of making this decision would be to look at FERs, however the thresholds for FER in making these decisions ought to have some logical connection with upper layer protocols. We discuss one possible way of doing this.

As an example, consider a conservative design with stringent QoS constraints wherein the system has to be designed in a manner for smooth operation as in Figure 4.4(b) such that, like Figure 4.3(a), the maximum normalized delay of any constituent RLP frame of an IP packet, d_{tot}, is within transmission time for next IP packet, i.e. a maximum normalized delay of 2. Now using our simulator, we can calculate d_{tot} for different FER structures, retransmission settings and latencies,



(a) $\epsilon = 0$

(b) i.i.d. FER = 0.20

(c) Correlated FER = 0.20,$p = 0.05$,$q = 0.20$

(d) i.i.d. FER = 0.35

Figure 4.6: TCP behavior for $R_{phy} = 153.6$ kb/s (sequence numbers are in *modulo*-80 fashion).

and based on that, a maximum level of sustainable SCH allocation can easily be calculated for this case by noting that the boundary condition for smooth operation is when last RLP frame of current IP packet is recovered just before the transmission of last frame of next IP packet, i.e., $d_{tot} < N_{rlp}(R_{phy}, S_{ip})$. This inequality together with equation 2.2 can be used for calculating the maximum allowable SCH allocation i.e. R_{phy} for a given FER structure and thus BSC channel allocation decisions can be made for the case under consideration. Similar arguments can be used to develop a system for a more liberal design with a wider

delay spread and a range of FER structures, radio link latency and retransmission settings.

In this chapter we examined the impact of wireless errors on performance of upper layer protocols and found that the impact is not very significant for a reasonably good radio channel. In the next chapter, we will focus on the other issue of dynamic assignment of supplemental channels in cdma2000 networks.

Chapter 5

Congestion due to Rate Variations

5.1 Introduction

A typical buffering scheme for cdma2000 network is shown in figure 5.1. Radio link rate variations are quite frequent in wireless networks[18], [1]. For instance in cdma2000 networks, on top of fundamental channels (FCH), supplemental channels (SCH) of one of the rates as shown in table 2.1 could be assigned to the mobile stations for specific duration in the range 20ms to 5.12 seconds. These allocations can be done in a variety of schemes, and one such scheme is *finite-burst* mode[19]. Clearly, sudden allocation of a supplemental channel means an equivalent swing in the total bandwidth of the radio link which might cause problems at intermediate shared buffers as discussed earlier.

Queue Management (QM), especially Active Queue Management (AQM) has been an active area for research ever since the introduction of RED [20] but all the work in this area focused on mainly wired networks with nearly fixed bandwidths and delay constraints. Network designers for these networks, after identifying the bottleneck, based on their experience of some a priori estimates on mean delay and bandwidths were able to comfortably come up with parameters for router settings. The problem of queue management in wired domains was simple in the sense that

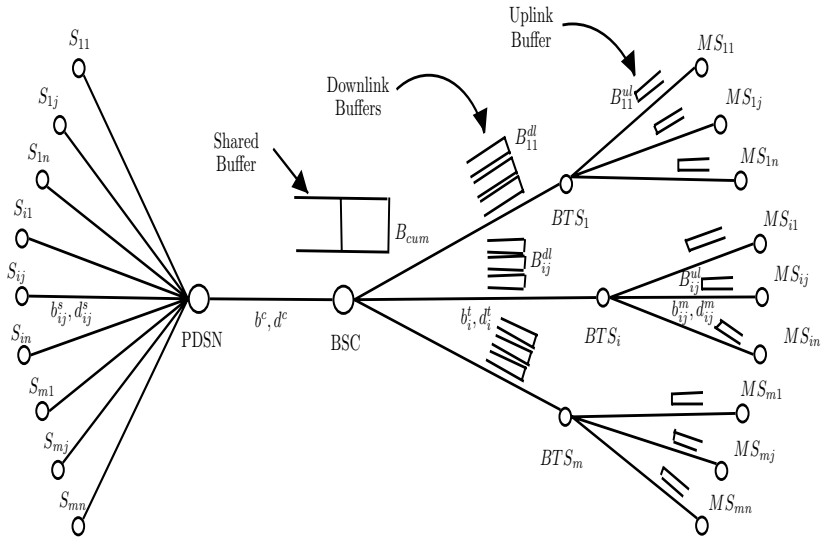


Figure 5.1: Buffering scheme in cdma2000 data services (The symbols used are described in table 5.2).

only a few parameters had to be tuned for a well-understood and well-behaved wired network.

However such assumptions are not true in wireless domains where bandwidths oscillate and delays can vary sharply. This poses problems for both queue management techniques and transport protocols' design. Whereas a sharp rise in bandwidth poses problems for the former due to a sudden enormous burst of packets delivered by the TCP agents due to ACK compression. On the other hand, the latter is confronted with the possibility of spurious timeouts caused by sudden deallocation of high bandwidth [16]. This problem is discussed at length in [21]. However it can be shown that the latter problem is rare in occurrence and occurs for particular setups of TCP window settings and the authors of [16] report that window sizes somewhat larger than delay-bandwidth product of the network

can eliminate the problem altogether. However queue-management problem is one that demands greater attention due to its complex nature and the fact that conventional wisdom of "large window sizes help wireless" means greater losses in cases of congestion at intermediate routers, as we will see later.

The problem of queue management is very complicated due to the fact the location of bottleneck changes frequently based on current state of resource allocation. With the proposed introduction of higher data rate techniques like CDMA 1X-EVDO that aim at providing higher data rates to the tune of 2 Mb/s, a global network view of the congestion and QM issues that does not assume wireless links as the only bottleneck would be more suitable. A review paper[33] on current ongoing research on wireless links considers a study of QM issues in shifting bottleneck scenarios as highly desirable, and our work, for the first time, presents an extensive study based on analysis and simulations. It differs from earlier works in the sense that it does not consider the end wireless links as the only bottlenecks. As an example to illustrate our point, figure 5.2 shows the TCP traces (obtained using *ns2* simulator[9]) for a mobile whose rate is switched from base FCH rate of 9.6 kb/s to 163.2 kb/s (FCH + 16X SCH) at $100s$. The intermediate buffer is assumed to have a service rate of mere 40 kb/s and a buffer capacity of 5 packets for this flow. The traces of dropping at this buffer are also shown. It is clear that sudden allocation of higher bandwidth leads to ACK compression and resultant prolonged timeout-based recovery.

In this reference, it would be worthwhile to mention that the QM problem of wireless domain is twofold - individual link QM and collective QM for other network nodes. The heart of this argument lies in the fact that not only the individual link buffers, B_{ij}^{dl}s, can overflow, instead situations may be there when other collective buffers like B_{cum} feeding these individual link buffers can also overflow. It has to be understood that these two problems are of entirely different natures

and have to be addressed differently. Table 6.2 compares these two. Notably enough, some research efforts have been done towards the study of problem of link QM [22] and solutions have been proposed but no studies in our knowledge have been performed on the issue of possibility of other network bottlenecks that impact the overall performance in a significant manner. We would also like to mention that link queues are tiny queues and any kind of QM technique employed for links only would not make appreciable overall difference. On the other hand, our analysis of QM that encompasses other network nodes together with wireless link, would be instrumental in improving the performance. It may eventually turn out that these two QM techniques may need to complement each other instead of being completely disjoint problems. In a nutshell, we would like to say that a unified study of QM techniques involving wired links together with wireless links is essential for developing an effective approach.

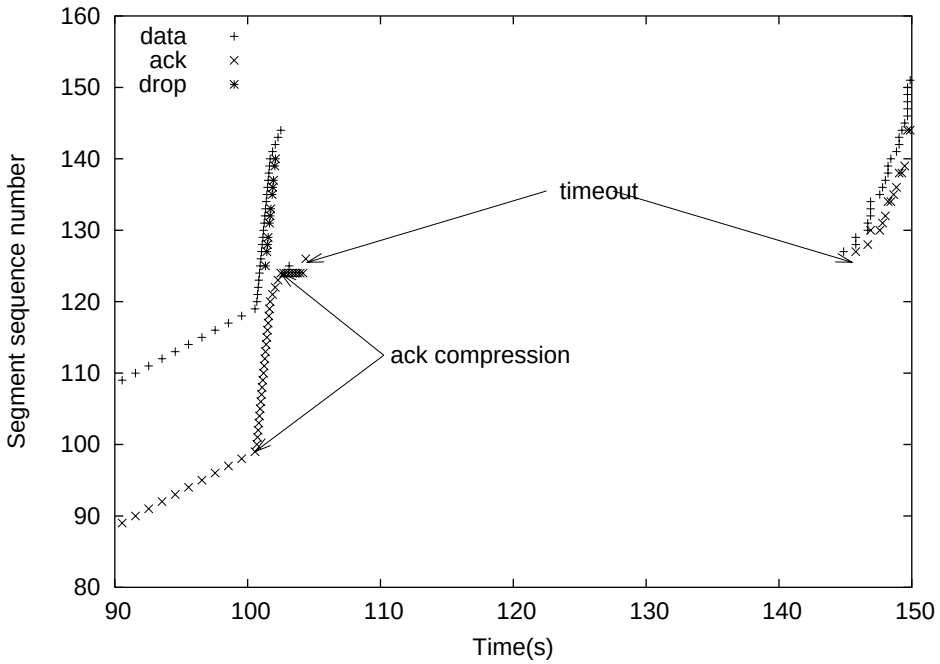


Figure 5.2: TCP trace for SCH allocation.

Link QM	**Network QM**
QM scheme for link buffers e.g. B_{ij}^{dl}, B_{ij}^{ul}.	Scheme for other network buffers that may overflow, like B_{cum} in figure 5.1.
Stores packets for/from a mobile only.	Stores packets for/from many mobiles.
Usually handles just 1, at most 3-4 TCP flows.	Handles lots of TCP flows, several hundreds of them.
Very low statistical multiplexing.	High statistical multiplexing.
Deterministic QM schemes like PDPC[22] preferable.	Probabilistic QM schemes like RED preferable, however possibly with some modifications

Table 5.1: Comparison of Link QM and Network QM

5.2 Modeling Bandwidth Oscillations

To develop an intuitive understanding of the congestion problem, we try to model the impact of rate variations by an analytical model for losses based on continuous flow approximation as used in some previous works [23, 24, 25, 26]. Figure 5.3 shows the model we are considering for analysis of a single mobile's flow which is assumed to have a fixed share of resources at the shared buffer, i.e. an individual service rate of b_b, and an available buffer space B. Note that we are assuming this to be reserved for the particular mobile under consideration. The downlink buffer, B_{ij}^{dl}, is the radio link buffer that stores the frames destined for the mobile. The service rate of this link buffer is determined by the aggregated rates of channels allocated to the mobile and in a *dynamic* environment, it keeps on switching between a lower value, $R_{ij} = b_{ll}$ corresponding to the base FCH rate, say 9.6 kb/s and the higher rate of b_{lh} equal to sum of rates of an FCH and a SCH, say $R_{ij} = FCH_{1X} + SCH_{16X} = (9.6 + 153.6)kb/s = 163.2kb/s$. Shared buffers are normally designed for nominal loads and we are specifically interested in scenarios when b_b falls in between these two values (say a rate of mere $SCH_{4X} = 38.4$

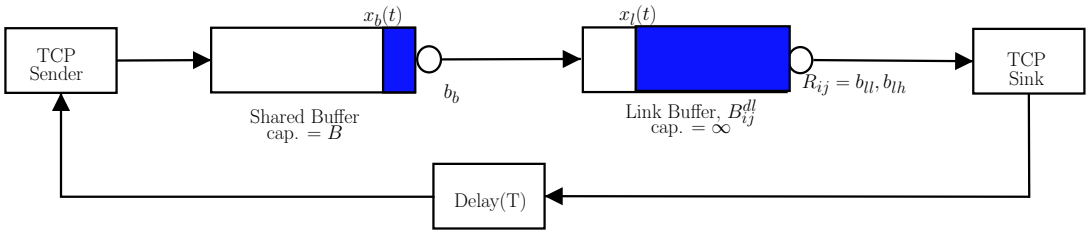


Figure 5.3: Model for losses due to rate change in a radio link.

kb/s for these values of b_{ll} and b_{lh}) as this configuration causes the *oscillation* of *queued workloads* between shared buffer and link buffer as R_{ij} fluctuates. Link buffers usually have very large capacities and most implementations would involve assigning a supplemental channel when data backlog is too much, so drop events in link buffers are rare. On the other hand, shared buffers, if not designed properly to absorb the oscillations, might cause excessive losses as shown next. The other conventions that we use are the contents of link buffer and shared buffers, $x_l(t)$ and $x_b(t)$, respectively and the round-trip propagation delay, T, lumped between the TCP sender and the sink. To illustrate the impact of a single switching, let us assume that the mobile was operating with the lower link rate of b_{ll} for a long duration prior to time t_0. Assuming that the window size of TCP sender is fixed at W, the contents of link and shared buffers for long-term steady state prior to t_0 are $x_l(t) = W - b_{ll} \cdot T$ and $x_b(t) = 0$. Now if at instant t_0, the link rate is switched to the higher value of b_{lh}, the queued workload in link buffer begins to move to the shared buffer since link buffer empties. This is due to the fact that ACKs are generated at a faster rate and TCP sender sends more traffic to the shared buffer. To calculate the losses at shared buffer, we are specifically interested in the starting and ending times, t_{start} and t_{end}, when the packets come to shared buffer at the higher rate of b_{lh}, as it is during this period the shared buffer begins to fill and possibly overflows. Clearly, leading edge of packets at the higher rate

reaches shared buffer at $t_{start} = t_0 + T$. Now for t_{end}, one of the following three possible cases have to be considered:

Case 1: The link buffer runs empty before the leading edge of packet flow at higher rate reaches shared buffer. Clearly for this case, after the bandwidth switch at $t = t_0$, $\frac{dx_l(t)}{dt} = b_{ll} - b_{lh}$. The time required for link queue to become empty is $\frac{0 - x_l(t_0)}{b_{ll} - b_{lh}}$ $= \frac{W - b_{ll} \cdot T}{b_{lh} - b_{ll}}$. So, the governing condition for this case to happen is $T \geq \frac{W - b_{ll} \cdot T}{b_{lh} - b_{ll}}$, or $b_{lh} \geq \frac{W}{T}$. Clearly, this change in state of link queue from being non-empty to empty will reach TCP sender after a delay T, when it stops sending the packets to shared buffer at the higher rate b_{lh}. Thus,

$$t^*_{end} = t_0 + \frac{W - b_{ll} \cdot T}{b_{lh} - b_{ll}} + T \quad \text{if } b_{lh} \geq \frac{W}{T} \tag{5.1}$$

Case 2: The leading edge of packet flow at higher rate reaches the shared buffer before the link buffer runs empty. In this case, for the first T seconds after t_0, the shared buffer's inflow rate is b_{ll}, for $t_0 \leq t \leq (t_0 + T)$, afterwards it becomes b_{lh}, for $t_0 + T \leq t$ and while $x_l(t - T) \geq 0$. Consequently outflow rate of shared buffer is b_{ll} till $t < t_0 + T$ and b_b, afterwards. The draining rates of link buffer are, $\frac{dx_l(t)}{dt} = b_{lh} - b_{ll}$ for $t_0 < t < t_0 + T$ and $(b_{lh} - b_b)$ afterwards while $x_l(t) \geq 0$. Thus at $t = t_0 + T$, the content of link buffer is, $x_l(t_0 + T) = x_l(t_0) - (b_{lh} - b_{ll}) \cdot T = W - b_{lh} \cdot T$. So the total time after t_0, required for the link buffer to run empty is $T + \frac{x_l(t_0 + T)}{b_{lh} - b_b} = \frac{W - b_b \cdot T}{b_{lh} - b_b}$. Since there is a delay of T seconds between link buffer and TCP agents, shared buffer will stop receiving packets at high rate at,

$$t^{**}_{end} = t_0 + \frac{W - b_b \cdot T}{b_{lh} - b_b} + T \quad \text{if } b_{lh} < \frac{W}{T}. \tag{5.2}$$

Case 3: The leading edge of packet flow at higher rate reaches the shared buffer before the link buffer runs empty and the TCP sender becomes aware of a drop event in shared buffer before it stops receiving packets at higher rate. Clearly, as described earlier, if a TCP sender receives first trail of packet flow after a drop event, it stops sending any further packets irrespective of arrival rate of ACKs. In

practice, it waits till it gets three duplicate ACKs, after which it retransmits the lost segment. Now for our model, the shared buffer , for $t > t_0 + T$, begins to fill at a rate $\frac{dx_b(t)}{dt} = b_{lh} - b_b$. It gets filled after another $\frac{B}{b_{lh}-b_b}$ seconds and the buffered packets in shared buffer at this instant take another $\frac{B}{b_b}$ seconds to clear the shared buffer after which the trail reaches the link buffer. The remaining content of link buffer at this instant is $W - b_{ll} \cdot T - (b_{lh} - b_{ll}) \cdot T - (b_{lh} - b_b) \cdot (\frac{B}{b_{lh}-b_b} + \frac{B}{b_b})$. This remaining data in the link buffer ahead of the trail behind first dropped segment drains at rate b_{lh} and after another T seconds the trail behind first drop event reaches the TCP sender and at that point TCP sender stops transmission. So, for this case:

$$\begin{aligned} t^{***}_{end} = \quad & t_0 + T + \frac{B}{b_{lh}-b_b} + \frac{B}{b_b} \\ & + \frac{W-(b_{ll}\cdot T)-((b_{lh}-b_{ll})\cdot T)-((b_{lh}-b_b)\cdot(\frac{B}{b_{lh}-b_b}+\frac{B}{b_b}))}{b_{lh}} \\ & + T \\ & = t_0 + T + \frac{B}{b_{lh}-b_b} + \frac{W}{b_{lh}} \end{aligned} \tag{5.3}$$

The governing condition for this case can be obtained by using the values of t_{end} in equation 5.2 and 5.3 and noting that the value in latter should be smaller than the one in former to let the TCP sender become aware of first drop before it stops receiving packet flow at higher rate. This leads to a condition,

$$t^{***}_{end} \leq t^{**}_{end} \Rightarrow b_{lh} \leq \frac{W}{T + \frac{B}{b_b}} \tag{5.4}$$

Based on equations 5.1-5.4, the time at which the shared buffer stops receiving packets at higher rate of b_{lh} either due to link buffer becoming empty or TCP sender receiving information about first drop event is given by:

$$t_{end} = \begin{cases} t_0 + T + \frac{B}{b_{lh}-b_b} + \frac{W}{b_{lh}} & \text{if } b_{lh} \leq \frac{W}{T+\frac{B}{b_b}} \\ t_0 + T + \frac{W-b_b\cdot T}{b_{lh}-b_b} & \text{if } \frac{W}{T+\frac{B}{b_b}} < b_{lh} < \frac{W}{T} \\ t_0 + T + \frac{W-b_{ll}\cdot T}{b_{lh}-b_{ll}} & \text{if } b_{lh} \geq \frac{W}{T} \end{cases} \tag{5.5}$$

To calculate the volume of packets dropped from the shared buffer we note that no losses occur while the buffer gets filled and losses occur afterwards. So from time t_{start} to $t_{start} + \frac{B}{b_{lh}-b_b}$, no losses occur and afterwards losses occur at a rate

$(b_{lh} - b_b)$ until time t_{end}. These concepts together with equation 5.5 can be used to calculate the loss volume (L_V) that comes out as:

$$L_V(b_{ll}, b_b, b_{lh}, W, B, T) =$$

$$\begin{cases} \frac{W \cdot (b_{lh} - b_b)}{b_{lh}} & \text{if } b_{lh} \leq \frac{W}{T + \frac{B}{b_b}} \\ max[(W - B - b_b \cdot T), 0] & \text{if } \frac{W}{T + \frac{B}{b_b}} < b_{lh} < \frac{W}{T} \\ max[(\frac{(W - b_{ll} \cdot T)(b_{lh} - b_b)}{(b_{lh} - b_{ll})} - B), 0] & \text{if } b_{lh} \geq \frac{W}{T} \end{cases} \tag{5.6}$$

5.3 Simulation Model

For the purposes of simulations, we used an implementation of cdma2000's link layer protocol, RLP[1], in *ns2*[9] simulator. This RLP module[10] can be used to model the SCH allocations and wireless losses. For analytical simplification, we assumed error-free wireless links. A long FTP session is assumed to be conducted between fixed hosts in external internet and mobile stations. The setup is exactly as shown in figure 5.1 and other parameters that will be used (unless otherwise specified) are presented in table 5.2.

5.4 Results and Analysis

In this section, we will use the two models that were presented in last section and will try to characterize the queue management problems due to bandwidth oscillations.

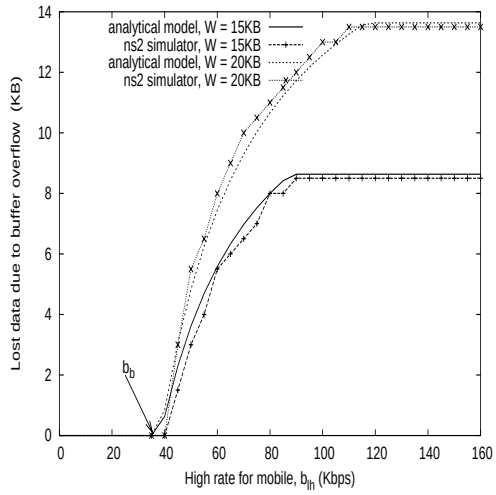
5.4.1 Dependence on window size, bandwidth swing and round-trip delay

Equation 5.6 shows the relation of losses due to bandwidth switching for a single mobile under the assumptions of fixed resources for it at the shared buffer. We simulate the similar scenario in the simulation environment by using a droptail buffer of fixed size and a fixed service rate for packets destined for the mobile.

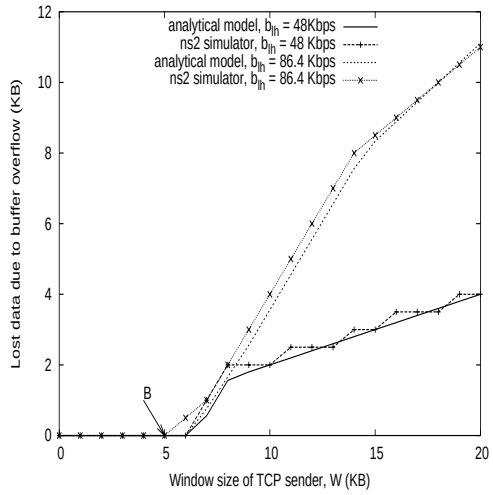
Table 5.2: Simulation parameters

Description	**Symbol**	**Value(s)**
Application		FTP
TCP version		Reno
TCP Segment Size		460, 960 Bytes
Receiver's buffer size		32KB
TCP Segments per ACK		1
TCP Minimum RTO		$1s$
IP Packet Size		500, 1000 bytes
Number of BTS	m	1, 10
Number of MSs in each BTS	n	1, 10
Mobile's Rate	R_{ij}	$9.6 - 163.2$ kb/s
Bandwidths	 b^s_{ij} b^c b^t_i b^m_{ij}	 100 Mb/s, for all i, j 100 Mb/s 100 Mb/s, for all i R_{ij} kb/s, for all i, j
Delays	d^s_{ij}, d^m_{ij}, d^t_i, d^c	37.5ms each
BSC's Buffer	B_{cum}	Service rate = $m \cdot n \cdot 38.4kb/s$ Droptail(cap.= 5KB) RED(cap. = 300KB, $min_{th} = 20KB$, $max_{th} = 60KB$).
Link Buffer	B^{dl}_{ij}	service rate = R_{ij} cap. = ∞

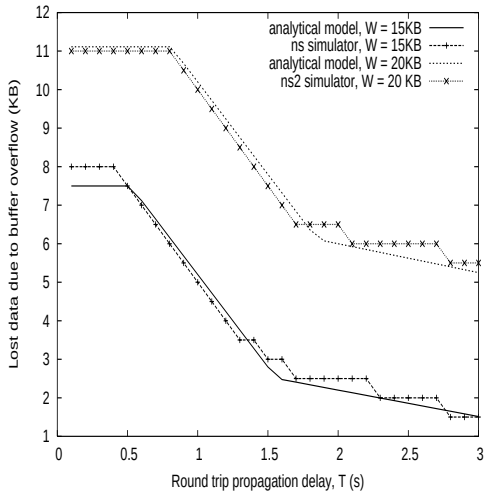
The parameters are shown in table 5.2. Figure 5.4 shows the comparison between the two models. It can be clearly seen that, except for the slight inaccuracy due to continuous flow approximation, the two models are in close agreement. Figure 5.4(a) shows that, for fixed W, T, b_b, b_{ll} and B, loss volume, L_V, first increases and then remains nearly constant. The limiting value of losses for increasing b_{lh} will be, $\lim_{b_{lh}\to\infty} L_V(b_{ll}, b_b, b_{lh}, W, B, T) = (W - B - b_{ll} \cdot T)$. So, for very sharp bandwidth swing, the losses can be as high as total buffered data at low bandwidth, $W - b_{ll}T$, minus the buffer space available at shared buffer, B. Figure 5.4(b) shows the loss variation with window sizes. It can be seen that losses increase linearly



(a) $T = 0.3\text{s}, W = 15$ kB, 20 kB



(b) $T = 0.3$s, $b_{lh} = 48$ kb/s, 86.4 kb/s



(c) $b_{lh} = 86.4$ kb/s, $W = 15$ kB, 20 kB

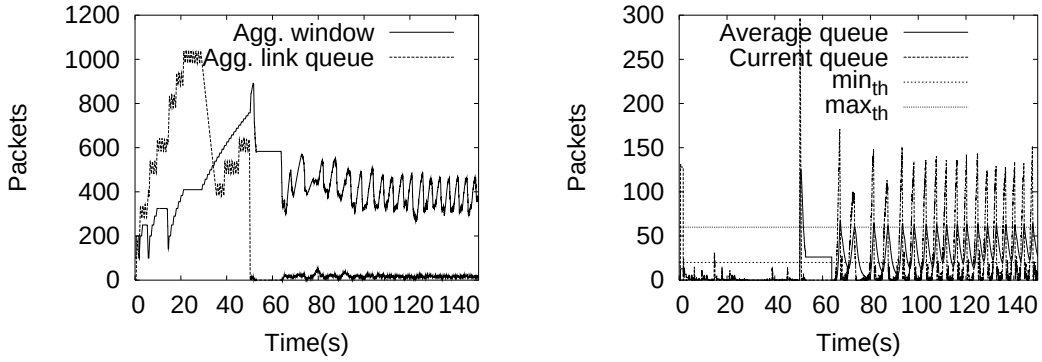
Figure 5.4: Variation of losses due to bandwidth change with, (a) mobile's higher switched rate, b_{lh} , (b) TCP sender's window size, W, and, (c) round trip propagation delay, T. (Other parameters are kept constant at $b_{ll} = 9.6$ kb/s, $b_b = 38.4$ kb/s, B= 5 kB, packet size for *ns2* simulations = 500 bytes)

with window size and the slope is governed by the conditions in equation 5.6. Loss variation with round-trip delay is shown in figure 5.4(c), which shows that, keeping other parameters fixed, losses decrease with increasing propagation delay.

5.4.2 A worst-case scenario

Next we focus on scenarios with multiple mobile users and simulate a worst-case scenario where all the mobiles under a BSC simultaneously change their rates, so that the aggregate service rate of link buffers becomes greater than the service rate of shared buffer B_{cum} instantly. This scenario translates to shifting of bottleneck from wireless links to the shared buffer. Figure 5.5 shows this case when the rates of all the 100 mobiles are switched from 9.6 kb/s to 163.2 kb/s due to a 16X (153.6 kb/s) SCH allocation at 50s. This means a change in aggregate link bandwidth from 0.96 Mb/s to 16.32 Mb/s. In this scenario, all the queues of link buffers are shifted to shared buffer, which has insufficient buffer space and hence results in excessive dropping of packets. Our example with a shared buffer following RED discipline shows that RED algorithm is unable to check the sharp growth of queues and drops packets like an ordinary droptail gateway after hitting its buffer limit of 300 packets. The low pass filter characteristics employed in RED algorithm create lot of inertia in it so that it takes a long time for it to raise its average queue length above the threshold to perform any useful congestion indication action. Even when it begins to do so, it takes almost a round trip time for the congestion indication to reach the senders so that they can resort to any reduction in sending rates by window-halving mechanisms and by this time, lots of packets are lost in a droptail-like fashion. Packet traces show that the total dropped packets, because of this droptail like behavior, are 729, close to the aggregate window size of 915 packets of TCP senders. This is due to the fact that the shared buffer has very little space of 300 packets (set as per recommendations in [20], which does not account for transient bursts due to bandwidth changes), and on top of that, being a RED gateway, it begins dropping all incoming packets as soon as average queue reaches the max_{th} value of 60 packets.

It is also notable that subsequent timeout based recovery is also very prolonged.



(a) Aggregate window, aggregate link queue.

(b) Queue at shared buffer, B_{cum}

Figure 5.5: Plots for simultaneous $16X$ SCH allocation to all 100 mobiles at $50s$.

The reason for this being that before 50s, all the mobiles were operating at a lower bandwidth of 9.6 kb/s and larger link queues and hence their retransmission timeout(RTO) will be highly inflated (RTT for such low bandwidth networks is highly dominated by current bandwidth, e.g. each 1000 byte packet in link queue of base service rate of $9.6kb/s$ adds 1.1 seconds to RTT) and after multiple packets are dropped, the TCP sender will wait for a long time for the lost packets before it begins to retransmit. This leads to extensive under-utilization of the overall system, in the sense that, owing to high rate allocation, the link buffer drains fast and soon runs empty and during the timeout based recovery after multiple losses at the shared buffer, the latter also remains empty as almost all the TCP senders keep on waiting for their outstanding packets that have been dropped by the shared buffer. This is exemplified in figure 5.5(b), where between 51s and 62s, both the shared buffer and the link queues are empty. Since both potential bottlenecks are empty, this is a period of under-utilization of the entire system. We call such periods as *dead-periods* and use them as a performance metric in our typical load scenarios later. Note that the average queue of RED remains constant at nearly 21 packets during 51s-62s even though actual queue is zero and no packets arrive. This is due to its implementation of not changing the average

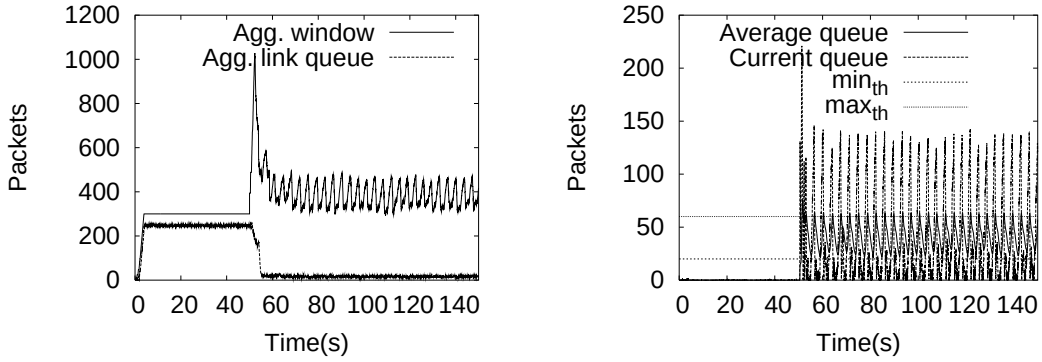
queue value when no packet arrives and changing it only when the first packet arrives, by an exponential decay mechanism based on link rate[20].

5.4.3 Comparison with a scenario with variable number of users

To establish our claim of bandwidth changes as a severe source of congestion, we benchmark the variable bandwidth scenario against the cases when number of users changes as in [27], keeping the aggregate link bandwidth and shared buffer's service rate constant. To simulate an aggregate bandwidth swing from 0.96 Mb/s to 16.32 Mb/s, we change the number of active users from 6 to 100 at 50s, keeping the link rate for all the mobiles at 163.2 kb/s throughout. The plots for this scenario are shown in figure 5.6. The new incoming users begin in slow start phase with an initial window of size one and quickly learn the allowable rates at shared bottleneck link. The mobile stations that were switched on from the start at 0s and were operating at larger windows are penalized by the RED algorithm and are forced to conform to the new changed network dynamics. All this happens very fast due to probabilistic dropping by RED wherein all the senders are not shut simultaneously as in previous case of variable bandwidth and flow of data across the shared buffer keeps on going at all times. Comparing this with figure 5.5, it is very clear that RED's random dropping mechanism is capable to handle congestion due to change in number of users.

5.4.4 Impact of rise time for aggregate bandwidth switching

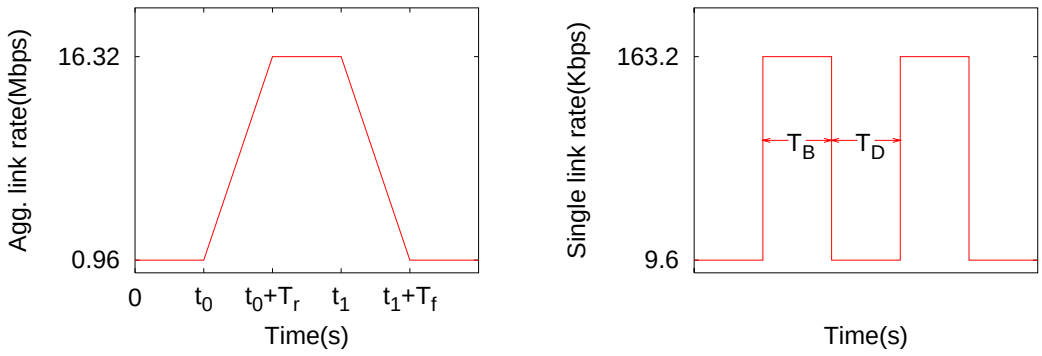
So far we considered a worst-case simultaneous allocation of SCH to all the mobile stations under a BSC. In a realistic scenario, the bandwidth allocations are usually *on demand* basis with little or no regulation. To simulate more gradual increases and to analyze the impact of rate of rise of aggregate bandwidth, we



(a) Aggregate window, aggregate link queue. (b) Queue at shared buffer, B_{cum}

Figure 5.6: Plots for more users joining simultaneously at 50s.

create a simulation setup as shown in figure 5.7(a). In this configuration, mobiles are switched to higher rate sequentially over a time span of T_r seconds, so that time between switching to higher rate of two mobiles is $T_r/(m \cdot n)$. This scheme allows for aggregate output links' bandwidth, as seen by BSC, to change from 0.96 Mb/s to 16.32 Mb/s over T_r seconds when each of the mobile's rate is changed from 9.6 kb/s to 163.2 kb/s one after another in a sequential manner. In the down-switching, the rates are decreased sequentially over a span of fall time of T_f seconds.

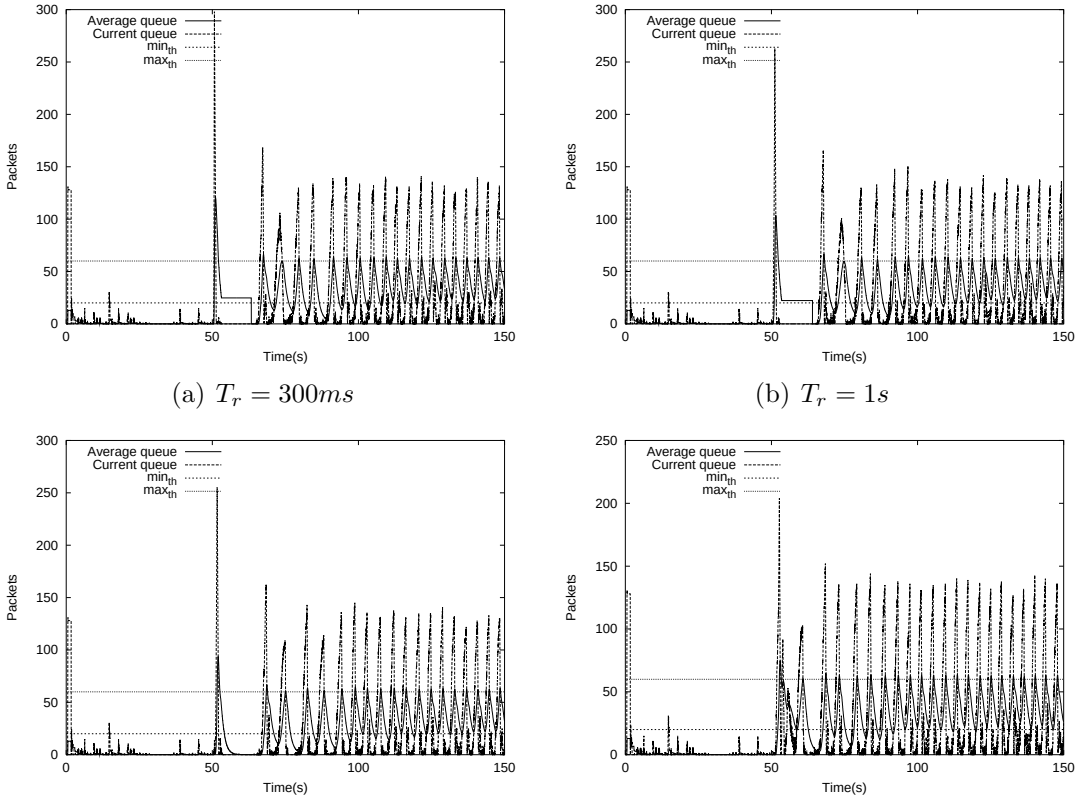


(a) Rise and fall times (b) Burst and delay durations

Figure 5.7: Illustration of rise and fall times, burst and delay durations.

Figure 5.8 shows the queueing behavior for BSC's buffer for an up-switching of the system beginning at 50s. The round-trip propagation delay is 300ms, as before.

We plot for T_r values of 0.3ms, 1s, 2s and 5s. It can be clearly seen that in all the first three cases, severe under-utilization of system resources occurs as all the TCP senders have suffered multiple losses and are waiting to timeout. For $T_r = 5s$, the packet flow is not halted after bandwidth allocations but even then the instantaneous queue at BSC's shared buffer is often zero, leading to under-utilization. This indicates that the extent of degradation due to congestion phenomenon after bandwidth change is larger for rise-times that are not significantly larger than round-trip times. In most cases, changes in aggregate bandwidths lead to empty queues at shared buffers and system under-utilization.



(a) $T_r = 300ms$ (b) $T_r = 1s$

(c) $T_r = 2s$ (d) $T_r = 5s$

Figure 5.8: Impact of rise time of aggregate bandwidth on queue behavior at shared buffer.

5.4.5 Typical load scenario

A typical scheme for assigning higher data rate to mobiles is to allocate and de-allocate SCH in a *finite-burst* mode[16]. This mode of operation is shown in figure 5.7(b), where a SCH of 153.6 kb/s is assigned for duration T_B seconds at inter-allocation spacing of T_D seconds. In the final part of this section, we look at the performance of RED for various values of burst-delay durations in this finite-burst mode of operation for the mobiles. We have chosen the rise and fall times for the 100 mobiles to be 10% of the burst and delay durations respectively, i.e. in each burst, the aggregate link rate reaches its maximum $0.1T_B$ seconds after the burst was given to the first mobile and vice versa for down-switching. First, we present a performance metric that is used for analyzing the load scenario that captures the underutilization of the whole wireless system. A good metric for QM behavior could have been the aggregate throughput for the mobiles, but reductions in throughputs can not be benchmarked against a standard as in [16]. There, the authors measured througputs against average channel rate. However in our study where the wireless channels are not the only bottleneck at all times, benchmarking against average channel rate makes little sense. Utilization of BSC's shared buffer could have been the other option but since BSC's queue is also not the bottleneck at all the times and when mobiles' aggregate rate is less than BSC's service rate, the shared buffer at BSC remaining empty is very natural. Essentially we would like to get a metric that simultaneously captures underutilization of both the wireless channels and BSC's input buffer. Since these are the only two possible locations of bottleneck in scenario we are considering, it suffices to analyze the utilization of these two. However, instead of looking at utilization of these two, we instead look at periods during which both are not utilized. These periods are representative of the underutilization of the whole system as both the potential bottlenecks are not being utilized during these periods. We call these times as

	T_B(secs)												
T_D(secs)	0.02	0.1	0.5	1	2	3	4	5	6	7	8	9	10
0.02	0.26	0.66	0.66	0.54	0.71	0.70	0.62	0.62	0.70	0.62	0.62	0.59	0.59
0.1	0.0	0.59	3.62	2.26	1.24	1.34	0.95	1.04	1.22	0.94	0.90	0.97	0.97
0.5	0.0	0.0	4.5	4.78	2.80	3.02	2.68	2.70	1.52	1.76	1.90	1.56	1.93
1	0.0	0.0	1.9	1.7	2.3	2.46	2.16	2.85	7.74	1.36	1.67	1.72	1.74
2	0.0	0.0	0.76	0.38	1.90	0.92	0.52	1.78	6.56	0.57	1.5	0.78	0.83
3	0.0	0.04	0.78	0.59	0.97	0.82	0.56	3.85	9.43	2.0	1.48	1.34	1.58
4	0.0	0.14	0.38	0.38	1.10	0.5	0.59	6.25	12.04	14.88	1.38	1.52	2.48
5	0.0	0.0	0.38	0.28	1.15	0.35	0.82	7.10	13.9	16.18	18.84	3.56	3.39
6	0.0	0.0	0.70	0.14	0.70	0.95	1.8	7.02	10.6	8.98	9.0	2.43	3.52
7	0.0	0.0	0.41	0.17	1.48	0.88	2.28	6.70	8.72	11.6	11.84	2.83	3.56
8	0.0	0.0	0.10	0.52	1.7	0.64	2.16	6.41	9.09	8.90	15.1	2.30	2.66
9	0.0	0.0	0.34	1.10	1.84	0.47	3.04	8.30	9.33	10.16	14.16	8.91	3.2
10	0.0	0.0	0.23	1.28	0.22	0.20	3.62	9.25	15.3	16.2	17.4	2.93	4.90

Table 5.3: Percentage dead-periods (100% underutilization definition) versus burst-delay durations.

dead-periods.

So, based on this idea, the fractional *dead-period* of 100% system underutilization, ρ_{dead}, could be defined as the ratio of total time, T_{dead}, during which the queue for BSC's input buffer, link queue of all mobiles - are all empty and the total simulation time T_{sim}. Clearly this metric quantifies the fraction of time during which the TCP senders are waiting to timeout after their packets are being dropped by BSC's input queue on sudden rate increase. During these periods, all the link queues as well as the BSC's input queue remain empty meaning waste of resources due to shared buffer's droptail-like behavior. The length of fractional dead-period versus various values of burst, delay durations is shown in Table. 5.3.

As can be seen in table 5.3, for smaller values of T_B and T_D, the fractional *dead-periods* are negligible. This is because TCP does not sense bandwidth changes at high frequency of allocation/de-allocation (i.e. $1/(T_B + T_D)$), ACK compression does not occur for long and hence TCP senders do not send data packets very fast to cause any havoc. However, at larger values, dead-periods are longer because now TCP senses this bandwidth change and puts in data at an excessive rate for BSC's shared buffer leading to loss of many packets and subsequent inefficient timeout-based recovery. For very large values of T_B and T_D, since not many

bandwidth changes occur during simulation time, this effect is less pronounced.

In figure 5.9, a relaxed definition of *dead-period* is used wherein fractional time when the BSC's input queue is empty and 90% of link queues are empty. Clearly a scenario with BSC's queue being empty and only 10 percent of links in use is also a strong indication of under-utilization of total available resources. It can be clearly seen that for certain values, fractional dead-period can be as large as nearly 25 percent. This is a waste of resources as during one-fourth of the time the whole wireless systems is not being used in the sense that both BSC's input queue and all link queues are empty and translates to a dismal value of overall system utilization at mere 75%.

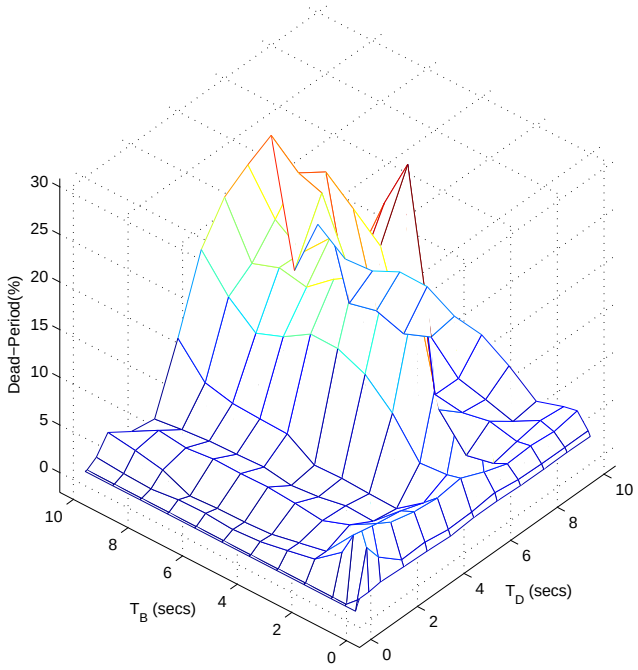


Figure 5.9: Fractional dead-period (90% underutilization definition) for various values of burst-delay duration.

So far we have seen how a rise in aggregate bandwidth poses problems for buffers at intermediate nodes. We have also seen that only a rise in a time period not significantly larger than round-trip times (RTTs) produces erratic behavior.

A RED gateway is effectively able to handle a slow, gradual rise in bandwidth. Also, in the long-run, RED is able to handle the extra bandwidth and under-utilization periods are only after the sudden allocation of bandwidth. So we can say that target problem's domain is limited to only short-term under-utilization after a sharp rise in aggregate bandwidth in a period not very large than RTT. For long-term buffer management with constant aggregate link rates or cases with very large rise times, RED performs reasonably well.

Nonetheless the significance of these short-term under-utilizations can not be ruled out. The reasons for it are twofold. First, the dead-periods of under-utilization are not really that short. In our simulations we have shown that these periods are roughly 10-12s which is by no means a small duration to be neglected that easily. Second, the allocations/deallocations are done quite frequently and during a typical load period this will be done several times, as a consequence of which, each allocation will create a dead-period of underutilization. These multiple dead-periods will have pronounced impacts on long-term aggregate throughput for mobiles in a typical load scenario.

5.5 Stable Operation of RED Mechanism

It was shown in previous sections that TCP behavior in wireless data networks is more bursty due to rate variations in radio links. This is due to *ack compression* after sudden allocation of higher data rate supplemental channels (SCHs) to mobiles resulting in sudden jumps in sending rates of TCP senders. Clearly, TCP connections over variable rate radio links would be more bursty than those over wired networks where propagation delays and link are constant throughout the path of the connection. Since the purpose of a buffer is to absorb burstiness and yield higher utilization, a natural consequence of burstier nature of wireless data networks would be greater buffer requirements to enhance utilization. Designing a

buffer involves choosing an appropriate size for effective link utilization and avoiding excessive queueing delays based on the behavior of the flows traversing the router. The most widely used rule-of-thumb states that a buffer size equal to the delay-bandwidth product (DBP) of the link is a good estimate for achieving high utilization [28]. This choice is based on the principle that on each window-halving of a TCP sender after a packet drop due to congestion, the bottleneck link's queue should not run empty. The other issue in buffer design is to employ some active congestion indication mechanism like Random Early Drop (RED) so that during times of congestion, all the flows are well multiplexed to yield higher throughputs and lower queueing delays. Many studies have been conducted on DBP-based rule-of-thumb and RED algorithm with varying results [29, 30, 31]. It is our belief that, instead of following a common method of parameter tuning, the design of buffers should be tailored according to the specific behavior of flows traversing the router. In this section, we attempt to find out the boundary for stable operation for particular setups. We derive these thresholds for one parameter at a time keeping others constant.

We are using the same setup as in previous sections. The settings for BSC's input buffer are as described in previous works, (target queue size for RED, $q_T = DBP$; minimum threshold for RED, $min_{th} = q_T/2$; maximum threshold for RED, $max_{th} = (3 \cdot q_T)/2$, queue limit for BSC's buffer, $limit = 3 \cdot q_T$). Figure 5.10 shows the scheme for sequentially increasing the aggregate rate of outgoing links. Each of the N(=100) mobiles are sequentially switched from the lower rate, b_{ll} kb/s, to the higher rate, b_{lh} kb/s, so that the average rate for the N mobiles changes over a period of t_r seconds and exceeds over average service rate, b_b, available at BSC.

5.5.1 Factors contributing to queue overshoots at BSC

Earlier it was shown that rise time has major impacts on the performance after bandwidth changes. The other parameters which impact the performance that are introduced in this section are the fractional rate overload, $\delta(= \frac{b_{lh}-b_b}{b_b})$, and queue overload in link buffers prior to increase in aggregate link bandwidth, β. The latter can be quantified as ratio of aggregate link queue length prior to onset of channel allocations to the queue limit of BSC's shared input buffer. We discuss each of them:

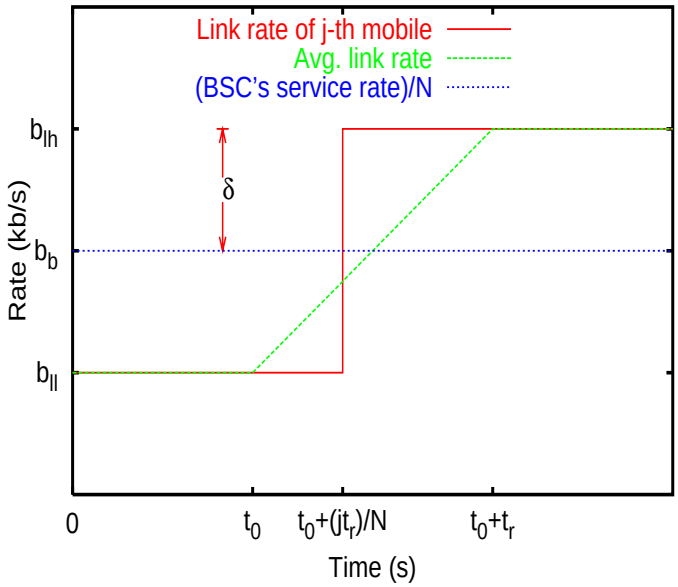


Figure 5.10: Scheme for increasing the radio links' rates.

Queueing overload: As discussed previously, channel allocations are also often based on data backlog in link queues. It is important to note that if excessively larger size of queue is allowed to build up in link buffer, it will result in longer duration of ack compression and greater losses. This factor is dependent on channel allocation policy, e.g. the inter-burst separation in finite burst mode[5]. A larger spacing between two consecutive SCH allocations results in a large queue in the link buffer and it can be alleviated by means of more frequent channel allocations

based on fast signaling. However, signalling can not be done at very high frequency due to system limitations. Even though cdma2000 systems support faster signaling than IS-95 and IS-95B by means of mini-messages and 5ms frames, still signalling can not be fast enough to achieve any desired level of parity between aggregate link queues and BSC's *limit*. Since, the present practice is to set BSC's buffer size based on DBP and does not account for burstiness due to channel allocations, this is often insufficient to absorb the sudden transfer of workload from links to BSC's input buffer after channel allocation. Still, it is possible to choose a suitable design so that the mismatch of queueing workload between shared buffer and link queues is not too large. We identify the thresholds for stable operation of traditional DBP based sizing of shared buffer, and beyond the threshold, buffers of sizes larger than those governed by DBP principle are necessary.

Rate overload: If the aggregate rate at which links can drain is greater than the service rate of input buffer of BSC, this leads to transfer of entire queued workload in links to the shared input buffer. If the value of aggregate bandwidth after channel allocations is very large than BSC's service rate, b_{ll}, it leads to faster filling of shared buffer and greater losses afterwards. As seen before, by means of continuous flow approximations, we had shown that for very sharp bandwidth switching, amount of data lost can be as high as the difference of queued workload in link queues prior to channel allocations, $W - b_{ll} \cdot T$, (W is aggregate window size of senders and T is round trip propagation delay) and the buffer space available at shared buffer, B. Rate overload can be mitigated by reducing the mismatch between shared buffer's service rate and aggregate link rate, but is often limited by processing powers available at BSC (considering that lots of processing of IP packets into RLP frames and vice versa needs to be done at the BSC).

Rise times: These are the times during which the aggregate link rate switches to higher value. As shown previously, rise times not significantly larger than round-

trip propagation delay lead to queueing problems in the shared buffer. We note that since channel allocations are done in a totally unregulated manner, rise-time is one factor that can not be tuned in the design process but can only be obtained empirically from real systems.

In determining the thresholds for stable operation of queueing behavior we use the following rule. The threshold for a variable, keeping other parameters constant, is the minimum value of the variable for which either of the following two things occur: (1) the queue length of shared buffer just hits its maximum value equal to the buffer space , $limit$, or (2) the EWMA (Exponential Weighted Moving Average) averaged queue length of RED algorithm just hits the maximum threshold parameter, max_{th}. We argue that these two conditions define the threshold by observing that a stable queue management would be one that preserves its probabilistic dropping to maintain good statistical multiplexing even during congestion. Since, conditions (1) and (2) are the ones that trigger undesirable droptail behavior, the prime target of a stable design would involve avoiding these two as much as possible. Although some researchers have proposed the gentle mode of RED beyond max_{th}, still we would like to maintain that average queues greater than max_{th} are the regions that should be avoided and aim at operating the average queue in a linear probabilistic dropping mode between min_{th} and max_{th}.

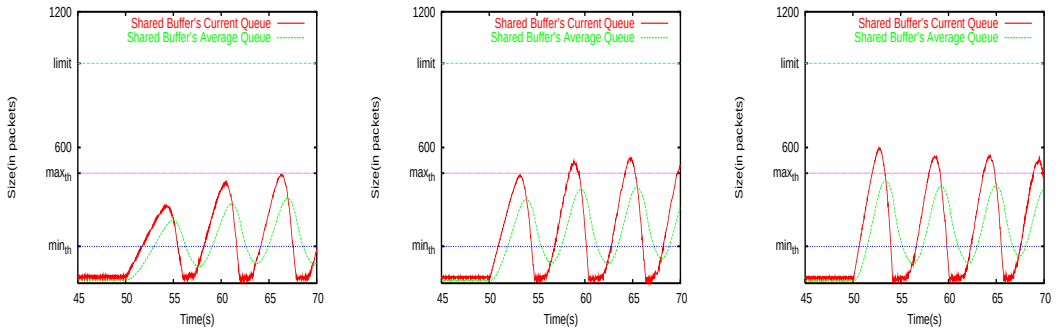
5.5.2 Results

We simulated the congestion scenario for a variety of parameter settings. Figure 5.11 shows the impact of varying the rate overload, δ, keeping the values of queueing overload constant at $\beta = 30\%$ and rise time at $t_r = 1s$. It can be clearly seen that transfer of workload from links to shared buffer is smooth for lower values of δ and is very sharp at higher values leading to prolonged timeouts and drops. For each of the overloads, the number of packets drops in the 5 seconds after the

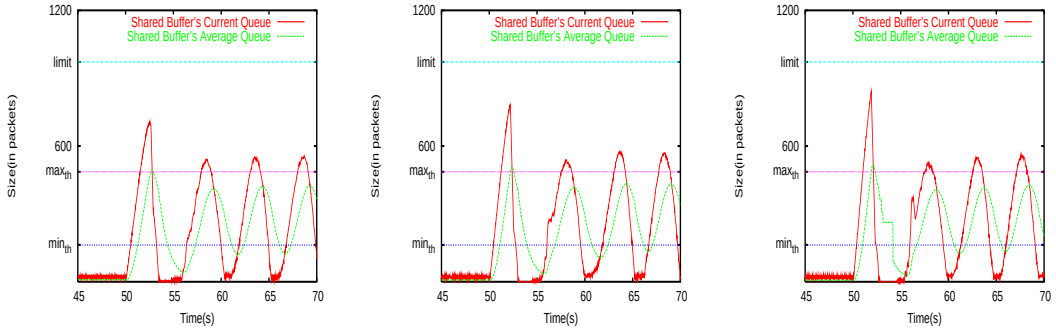
channel allocation are also shown. It is during this period that most of drops are due to channel allocations and not because of long term probabilistic dropping of RED mechanism.

We now proceed on to identifying the threshold for stable operation for this setup. A closer examination of the figure reveals that the threshold condition (1)of previous section occurs for some value between 40% and 45% and condition (2) first occurs for some value between 15% and 20%. Clearly, in this case, average queue hits max_{th} value before queue size reaches the buffer limit and hence condition (2) governs the threshold value of δ. But this is not always true as, for some simulations, it was observed that condition (1) is the determining factor. After running several simulations in the range 15% to 20%, it was observed that 18.5% is the minimum value at which condition (2) just occurs and hence this seems to be the threshold value for stable operation. This may seem a too low value for stable operation, but this is expected for a small value of rise time of 1s.

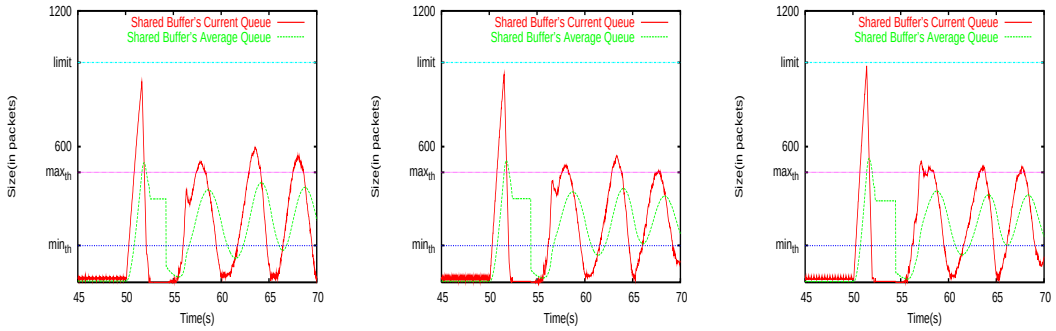
We generalize this concept of obtaining the values of rate overload threshold, δ^*, for various values of queue overloads and rise times as shown in figure 5.12. The system operates smoothly for values of overload rates below the threshold and beyond that performance will be degraded due to channel allocation induced congestions. A lower value of queueing overload and a higher value of rise time yield a broader range of operation. Thus, by means of simulations, range of stable operation can easily be found and used to enhance the performance. If the setup parameters can not be compromised for strict adherence to stable operation, larger buffers are the only option to mitigate the problem. This might come as a little surprise because larger buffers would mean greater queueing delays, especially since some latest work [29] claim that even buffer sizes much smaller than DBP yield same performance, but our suggestions are based on specific needs of wireless networks to absorb the excessive burstiness due to rate variations.



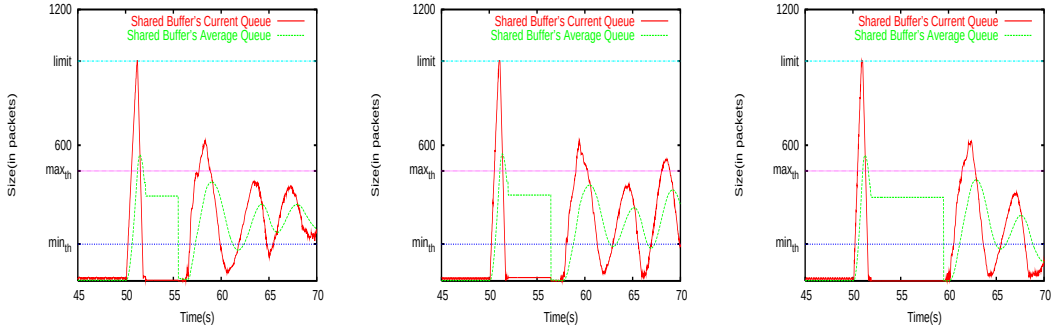
(a) $\delta = 5\%$, drops = 12 (b) $\delta = 10\%$, drops = 18 (c) $\delta = 15\%$, drops = 33



(d) $\delta = 20\%$, drops = 55 (e) $\delta = 25\%$, drops = 84 (f) $\delta = 30\%$, drops = 99

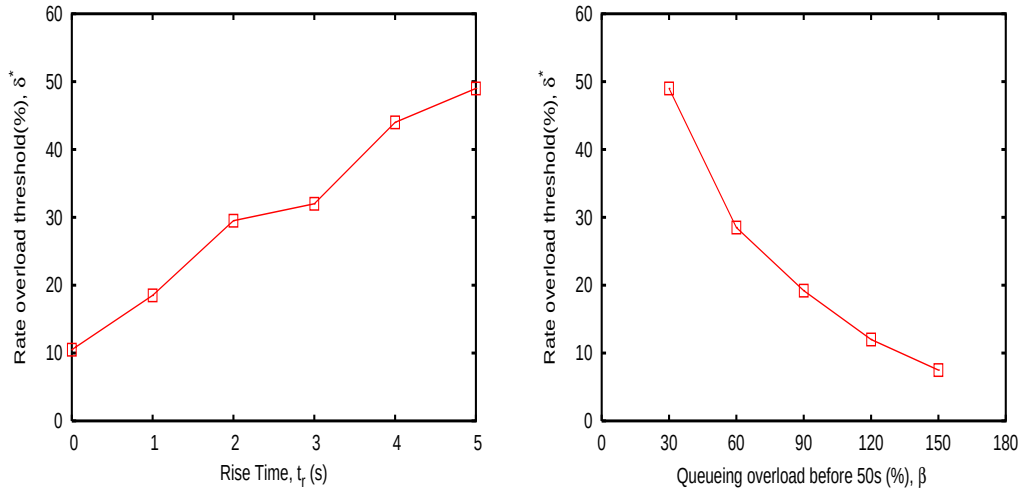


(g) $\delta = 35\%$, drops = 134 (h) $\delta = 40\%$, drops = 167 (i) $\delta = 45\%$, drops = 192



(j) $\delta = 55\%$, drops = 213 (k) $\delta = 65\%$, drops = 244 (l) $\delta = 75\%$, drops = 291

Figure 5.11: Queueing behavior in shared buffer and link queues with varying levels of overload. For each of the figures, the number of packets dropped in 5 seconds after the channel allocations (i.e. between 50-55s) are also shown. The values of t_r and β are constant at 1s and 30% respectively.



(a) Constant queueing overload, $\beta = 30\%$ (b) Constant rise time, $t_r = 5$s

Figure 5.12: Threshold for stable operation with default settings

Chapter 6

Future Work and Conclusions

In previous chapters the performance of cdma2000 networks in variable rate and delay conditions was analyzed. We summarize our key results and present some future working directions for this study.

6.1 Wireless Errors

In chapters 3 and 4, we presented an accurate model for link layer in cdma2000 data networks. Fading-induced effects have been examined at various levels and an exact analysis for interaction of link layer protocols and upper layers has been presented based on our simulation tool. We illustrated the conditions under which link layer retransmissions produce the undesirable effects such as delay spikes, residual frame losses and reduced data rates.

With present technology, it is not possible to reduce wireless link loss rates comparable to those in wired networks. So, any protocol like TCP that works with assumptions of nearly error-free links, if used over wireless links would result in poor performance. However, end-to-end protocols like TCP are key to providing desired user-perceived Quality of Service (QoS) (in line with end-to-end principle[34]), and any QoS mechanism employed over just wireless links would be insufficient. Thus, a mapping between end-to-end QoS objectives and wireless link design parameters is necessary for developing a better wireless data system. Since

our model provides an accurate relationship between the two, for future works, the simulation model presented in this thesis can be used for developing an elaborate scheme for channel allocation based on specific QoS requirements.

6.2 Rate Variations

A better understanding of queue management problem due to bandwidth oscillations was developed in chapter 5. We now try to explore the solution space for the problem of queue management in cellular networks at various levels and we also recognize the possible pitfalls in terms of applicability of these approaches.

6.2.1 Modifications to queue management

If somehow RED algorithm is able to track the sharp enormous growth in queue size on sudden allocation of bandwidth and distinguish it from usual transient spikes and additionally inform the senders without significant dropping in a droptail fashion, it could mitigate this problem. However, these are competing demands and it might be hard to find the decision boundary for them. On one hand, filtering out transients helps in allowing short-lived bursts but on the other hand this leads to sluggish response of router to bandwidth-induced sudden incipient congestion leading to loss of many packets in a droptail-like fashion. These contrasting demands make it difficult to understand how much a low-pass filter (LPF) behavior is desirable in calculating average queue size. However, if there were a way by which BSC had an idea of current aggregate bandwidth, it could make decision based on them so that it knows when it needs to react fast and when not to. It can alternatively allow for larger queue sizes when aggregate bandwidth is high and vice versa as our ultimate QM target is bounding the packet delays and not the queue sizes. In one set of experiments conducted as part of this thesis we examined the latter approach.

A case for dynamic settings of RED To argue a case for dynamic settings of RED algorithm a small, simplistic setup as used in figure 6 of [20] is used. The setup is shown in 6.1. Two values of bottleneck bandwidths of 45 Mb/s and 90 Mb/s are used in the link connecting node 5 to node 6. Our initial target is to get a power curve (throughput versus average queue size) as in figure 5 of [20]. The power curve plots for the two bandwidths are shown in figure 6.2.

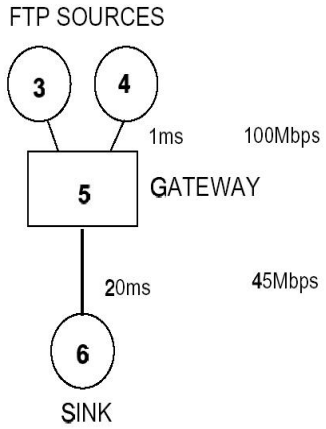


Figure 6.1: Setup for RED(from [20])

The network power offered by an AQM scheme is given by,

$$NetworkPower = (AggregateThroughput)/(AverageQueueingDelay) \quad (6.1)$$

and it has been recognized as an important metric in analyzing the efficacy of an AQM scheme. RED claims to keep the throughput high and average delays low and hence yields an overall high network power. In this reference, it would be important to mention that, for a network with fixed value of bottleneck bandwidth, the average queueing delay is directly proportional to average queue size and hence it can be effectively used as an estimator of queueing delay and RED parameters settings can be done based on that. However, in cases where the bottleneck bandwidth varies with time, average queue size no longer remains a good estimator

for average queueing delay. Remember, our core target for an AQM scheme is to keep a target value of average queueing delay, not an average size of queue. If the queue can be drained out at a faster rate, a liberal approach allowing greater value of average queue size keeping the average queueing delay constant needs to be followed.

First observe in figure 6.2 that the power curves for the two values of bandwidths are roughly the same. We now develop our analysis based on this curve. Figure 6.3 shows the power curve for some arbitrary setup. Let us first assume that the bottleneck link was operating at bandwidth B_1. Also, assume that the RED parameters are selected in such a manner to keep the average queueing delay at roughly around D^* by keeping the average queue length at q_1 so that the average queueing delay is $D^* = q_1/B_1$. So, the operating point in this case becomes $P_1(T_1, q_1/B_1)$, with aggregate average throughput at T_1 and average queuing delay at q_1/B_1. Now let us assume that at some instant the bandwidth becomes B_2, where $B_2 > B_1$. RED will show minor variation and finally settle down to same level of average queue at q_1 (original RED may show minor change in average queue length, but ARED[27] due to its very aggressive approach keeps the average queue length tied to fixed value of $1/2(max_{th} + min_{th})$). Now we will elaborate upon why this operating point $P_1(T_1, q_1/B_2)$ is not an optimal one for bandwidth B_2. Our target was to keep the average queueing delay around D^*, however RED because of its current queue-based algorithm went for an overkill and brought the average queueing delay at a far lower level of q_1/B_2. Instead, it should have settled down at point $P_2(T_2, q_2/B_2)$ (where $q_2 = (B_2 * D^*)$) thereby achieving a significantly higher level of throughput, T_2 and same level of average queueing delay,$D^*(= q_2/B_2)$.

So, based on this discussion, we can see that RED is incapable of achieving the maximum level of throughput for a certain delay constraint. The problem,

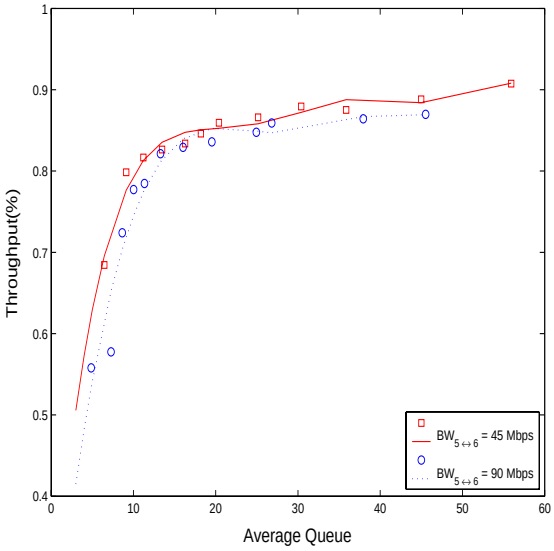


Figure 6.2: Throughput versus Average Queue Size

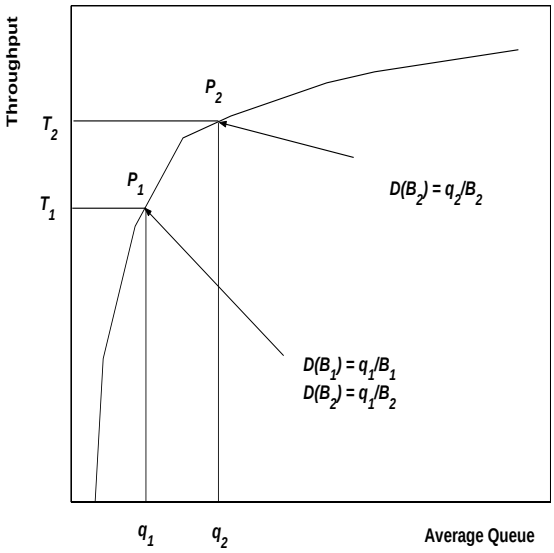


Figure 6.3: Proof of non-optimal operation of RED

as we saw with RED is because of its queue-length based control action which should have been based on queueing delay. So the claim is that, provided RED provides optimal solutions for fixed bandwidth bottlenecks, a minor variation of making queueing delay as the control parameter instead of queue length in RED's

operation should make it more robust for scenarios where bandwidths change and an *a priori* estimate on long-term average bandwidth can not be made. The next section proposes the modified RED algorithm that can possibly address this issue.

Dynamic RED for variable bandwidth scenarios The basis on which the following modifications to RED are proposed are based on assumption that the bottleneck node is aware of the changes in its output links' bandwidths. This can be true in CDMA networks' Base Station Controller, that changes the bandwidths of mobiles based on user demands. Since BSC has unified view of total available bandwidth in outgoing links, it should try to keep the queue length in such a manner that the total queueing delay in shared queue and individual link queue is at a desired level thereby keeping the throughput high. It should not keep the same level of average queue in shared buffer even if it output links can drain out at significantly faster rate, instead it should change shared buffer's queue length dynamically to keep the average delay suffered by a packet in the two queues roughly constant.

Let us now define the control parameter, *qdel*, the current level of queueing delay as, $qdel = q/b$, where q and b are current values of queue size and bandwidths respectively. The gateway is aware of the value of b based on above assumption. In the modified version of RED, *qdel* is the parameter used for desired control action. It can be verified that in a fixed bandwidth case, *qdel* becomes directly proportional to current queue, q, and the modified version of RED reduces to original RED. The modified version of original RED algorithm [20] to be used for variable bandwidth cases is shown in Figure 6.4.

Although this modification is proposed on basis of original RED [20], similar modifications can be made to ARED and *gentle* versions of RED. Note that the average queueing delay, avg_{del}, is calculated in a manner similar to original RED using EWMA low-pass filter with only modification of calculating average queueing

for each packet arrival
    calculate the average queueing delay avg_{del}
    if $min_{th} \leq avg_{del} < max_{th}$
      calculate probability p_a
      with probability p_a:
            mark the arriving packet
    else if $max_{th} \leq avg_{del}$
          mark the arriving packet
 where,
          $avg_{del} \leftarrow (1 - w_q)avg_{del} + w_q(qdel)$
          $qdel = q/b$
          q = current queue,
          b = current bandwidth.

Figure 6.4: Modified RED algorithm

delay based on current and past samples of queueing delays while original RED calculates average queue size based on samples of queue sizes. This approach guards the RED against transient changes in bandwidths and settles it to work with a long-term average value of bandwidth. Also note that, in modified RED, min_{th} and max_{th} are thresholds on queueing delays. It is believed that these modifications should tune RED to a queue level corresponding to a long-term averaged out bandwidth thereby providing nearly same delays as RED with higher aggregate throughputs than that.

Another approach to address rate variability would be to regulate the flow of returning ACKs at the intermediate shared buffer in some fashion so that TCP senders do not suddenly start pushing data at an excessive rate on seeing ACKs returning at a very fast rate due to high bandwidth allocation. This means that on sudden allocation, the BSC delays the returning ACKs for a while so that RED routers have time to respond to it and multiple drops of packets do not happen. In the long run, ACKs will be forwarded without any delay, so that we

are not unnecessarily increasing the RTTs. Basically in this scheme, on sudden allocation, we simply delay the ACKs for a while so that ACK compression does not happen and a sharp bandwidth change reflects in a smooth change in TCP sender's sending rate. Such an approach is presented in [35]. Therein the authors use a ACK-regulator mechanism by which TCP is adjusted for variable bandwidth, variable delay scenarios.

6.2.2 Fixing TCP

Another way of looking at the problem is to somehow fix the TCP in some manner so that its window mechanism responds smoothly to fastly arriving ACKs. In the long-run it should work with the fast rate but for periods just after allocation it should not change its sending rate abruptly. Such an approach would be difficult in practice for two reasons: first all the TCP senders can not be enforced to conform to this new approach, and second this assumes that rate of returning ACKs is totally dependent on wireless conditions. By not allowing TCP to increase its sending rate sharply for many ACKs in a short duration, we might lead to fewer packets at bottleneck resulting in underutilization.

The other working direction is to fasten the recovery process after multiple packet losses. Fortunately, a soultion to this exists in form of Newreno TCP's modified fast recovery algorithm to react to multiple packet losses. Basically, a Newreno TCP sender retransmits packets based on partial ACKs that arrive after detection of losses and subsequent partial recovery. This mechanism is very robust in handling multiple losses and resumes the normal flow of data in a shorter span of time. Figure 6.5 shows that the recovery time after multiple losses has been reduced to a mere 16s which, in case of a Reno sender in figure 5.2, was 50s due to the latter's dependence on timeout mechanism after multiple losses. Not all the TCP implementation in existing Internet use Newreno modification, and our

results supply another reason for enforcing the vendors to conform to Newreno modifications.

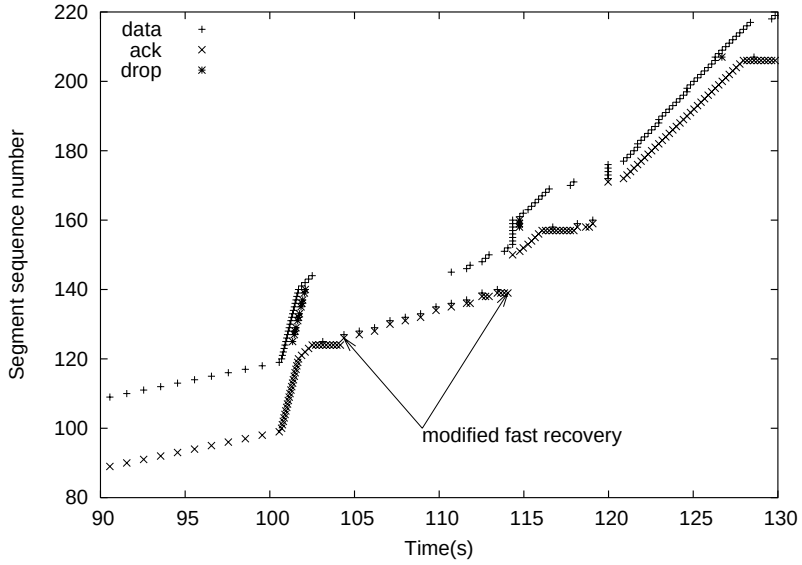


Figure 6.5: Enhancement offered by Newreno TCP's fast recovery algorithm as against Reno TCP in figure 5.2.

6.2.3 Scheduling bandwidth allocation/deallocation

Another approach that will solve most of the problems but lacks greater applicability is scheduling the allocation/deallocation of higher-bandwidth supplemental channels. If the aggregate links' bandwidth remains roughly the same, much of the QM problems will not occur. One way of doing this is by using some scheduling mechanism at BSC to do the allocation/dealloaction uniformly spread over the time. For instance, in our setups with burst-delay durations of T_B, T_D, whereby we give SCH to all mobiles at say t_0, remove it at $t_0 + T_B$ and again allocate it at $t_0 + (T_B + T_D)$, which results in oscillation of aggregate bandwidth between (mnb_{ll}) and (mnb_{lh}). If the BSC allotted these bandwidths in a sequential fashion like $k - th$ mobile is assigned SCH at time $t_0 + ((k-1)/(mn))(T_B + T_D)$, removed at $t_0 + T_B + ((k-1)/(mn))(T_B + T_D)$

and reallocated at $t_0 + (T_B + T_D) + ((k-1)/(mn))(T_B + T_D)$ for $1 \leq k \leq (mn)$, such an approach, while keeping the burst-delay durations of T_B and T_D for a mobile constant, also keeps the cumulative bandwidth at all times tied down to a value of $mn(b_{ll}(T_D/(T_B + T_D)) + b_{lh}(T_B/(T_B + T_D)))$. Under this scheme, individual bandwidths rise and fall abruptly but aggregate bandwidth remains constant and it can be shown that this poses no problem in terms of queue management. However since channel assignment is done in a completely unregulated fashion, such a smooth behavior will be difficult to be enforced at BSC. Instead, what could be done is to implement an algorithm at BSC by which, based on a value of maximum tolerable rise rate of aggregate bandwidth, other mobiles requesting additional bandwidth, when aggregate bandwidth rise rate has hit its maximum, are denied the SCH for a while so that queue management can be done smoothly. Such an approach requires only an estimate of current aggregate bandwidth at BSC and pre-determined thresholds of rise-rate for SCH denial at current juncture and of delay period after which this SCH can be allocated. This can be an effective approach if BSC is completely aware of current status of channels allocated to all mobiles under it.

6.3 Conclusions

In this work, an accurate model for link layer retransmission mechanisms is presented and it is shown by means of a simulation tool as to how it impacts the network performance. We further introduced a new problem of congestion due to variable rate links in wireless data networks. By means of analysis and simulations, we identified and characterized this problem. We analyzed the system performance under variety of configurations and based on them, we provide extensive results and possible working directions to stimulate further research to develop a solution to this problem.

Appendix

The Class RlpAgent

`RlpAgent` is a C++ class defined as follows:

```
class RlpAgent : public Agent { public :
        RlpAgent();                      //constructor
        void recv(Packet* pkt, Handler*);
        void send(Packet* pkt, Handler*);
        void timeout(int);
protected:
        int command(int argc, const char*const* argv);
        int lvn;                //
        int lvr;                // Seq. numbers
        int lvs;                //
        double rlp_rate_;          // Channel's Phy. Rate
        ErrorModel* em;         // Error Model
        Rlp_Timer rlp_timer_;
        NsObject* dst;          //Attached PPP Agent
        NakList nak_list_;      // Nak list for lost frames
};
```

This class needs to instantiated for each endpoint of the radio link. During a data transfer, the agent receives IP packets from its attached `PppAgent` object,

segments it into RLP frames and transmits these frames to the peer RLP object. The receiver RLP agent receives these frames and assembles them into IP packets and forwards it to its destination. Additionally, RLP agents handle the ARQ mechanism by few retransmissions. The receiver RLP maintains a `nak_list_` of missing RLP frames and tries to retrieve them using `NAK` control frames. This arrangement is shown in figure 3.1

Configuring an RLP Agent

The following Tcl commands can be used to configure various options for simulating a cdma2000 network:

```
//Setting up agents and parameters
set ppp0 [new Agent/Ppp]    //Creating a PPP agent
$ns attach $node0 $ppp0     //Attaching ppp0 to node0
set rlp0 [new Agent/Rlp]    //Creating a RLP agent
$ns attach $node0 $rlp0     //Attaching rlp0 to node0
$rlp0 set rawrate_ 9.6     // Phy. Chan. rate in kb/s
$rlp0 set rlp_delay 13     //frame count estimate for
                           //round-trip radio link delay
$rlp0 set hdrlen_ 5        // RLP frame header length in bytes
$rlp0 set slot_ 20          // Frame slot duration in ms

//Attaching a faded channel model with p = 0.05, q = 0.20
set good [ new ErrorModel 0 pkt]
set bad [ new ErrorModel 1 pkt]
set states [list $good $bad]
set periods [0.02 0.02]
set transmx [{0.95 0.05} {0.2 0.8}]
```

```
set trunit pkt set ttype time
set nstates 2 set start $good
set em_ [new ErrorModel/Multistsate $states $periods\
        $transmx $trunit $ttype $nstates $good]
$rlp0 errmodel_ $em_

//Connecting the RLP, PPP agents
$ppp0 dst $rlp0            // Attaching ppp0 to rlp0 $rlp0 dst $ppp0
$ns connect $rlp0 $rlp1    //Connecting two RLP agents $ns connect
$ppp0 $tcp0    //Connecting PPP agent to a TCP agent
```

Bibliography

[1] TIA/EIA/IS-707-A-1, "Data services options for spread spectrum systems - Radio Link Protocol type 3", June 2004, http://www.3gpp2.org/Public_html/specs/C.S0017-010-A_v1.0_040617.pdf.

[2] M. Chatterjee, G. Mandyam, and S. K. Das, "Fast ARQ in high speed downlink packet access for WCDMA systems", *Proc. of European Wireless Conference*, pages 451–457, May 2002.

[3] "Proposed 1XTREME physical layer delta specification", *Source: Nokia, Motorola, LSI Logic, Texas Instruments and Dot Wireless, Contribution to 3GPP2*, August 2000.

[4] A. Gurtov and S. Floyd, "Modeling wireless links for transport protocols", *ACM Computer Communications Review (CCR), to appear.*

[5] F. Khafizov and M. Yavuz, "Analytical model of RLP in IS- 2000 CDMA networks", *Proc. of IEEE Vehicular Technology Conference (VTC'02)*, vol. 1, pages 487–491, September 2002.

[6] F. Khan, S. Kumar, K. Medepalli, and S. Nanda, "TCP performance over cdma2000 RLP", *Proc. of IEEE Vehicular Technology Conference (VTC'00)*, pages 41–45, May 2000.

[7] A. Gurtov. NS2 simulation tests for modeling wireless links. In *ns2 simulator's tcl/ex/wireless-scripts*, December 2003.

[8] Y. Bai, P. Zhu, A. Rudrapatna, and A. T. Ogielski, "Performance of TCP/IP over IS-2000 based CDMA radio links", *Proc. of Vehicular Technology Conference (VTC'00)*, pages 222–236, September 2000.

[9] http://www.isi.edu/nsnam/ns/.

[10] http://www.sce.carleton.ca/∼vpaliwal/dist/.

[11] W.R.Stevens, *TCP/IP Illustrated*, Addison Wesley, 1993.

[12] TIA/EIA/IS-2000.1, "Introduction to cdma2000 standards for spread spectrum systems", March 1999.

[13] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi,"CDMA/HDR: A Bandwidth Efficient High Speed Wireless Data Service for Nomadic Users", *IEEE Communications Magazine*, vol. 38, no. 7, July 2000, pp. 70-77.

[14] http://www.opnet.com/products/modeler/home.html.

[15] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels", *IEEE Trans. Commun.*, 46(11):1468–1477, November 1998.

[16] M. Yavuz and F. Khafizov,"TCP over wireless links with variable bandwidth", *in Proc. of Vehicular Technology Conference(VTC'02)*, Sept. 2002, vol.3, pp. 1322 - 1327.

[17] H. Lin and S. K. Das, "An adaptive radio link protocol to improve TCP performance over correlated fading wireless channels", *Proc. of Personal Wireless Communications*, pages 24–28, September 2000.

[18] 3GPP TS 03.60:Digital cellular telecommunications system (Phase 2+), "General Packet Radio Service (GPRS): Service description, Stage 2".

[19] M. Yavuz and F. Khafizov,"Running TCP over IS-2000", *in Proc. of Intl. Conf. on Comm.(ICC'02)*, May. 2002, vol.2, pp. 3444 - 3448.

[20] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", *ACM Trans. on Networking(ToN)*, vol.1 no.4, August 1993, pp. 397-413.

[21] H. Inamura *et al*, "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks", RFC 3481, *Internet Engineering Task Force*, February 2003.

[22] M. Sagfors, R. Ludwig, M. Meyer and J. Peisa. "Queue management for TCP traffic over 3G links", *in Proc. of IEEE Wireless Communications and Networking Conference(WCNC'03)*, Mar. 2003.

[23] Y. Wardi and B. Melamed, "Loss volume in continuous flow models: fast simulation and sensitivity snalysis", *in Proc. of IEEE MED-2000*, Patras, Greece, July 17-19, 2000.

[24] B. Liu, D. Figueiredo, Y. Guo, J. Kurose and D. Towsley, "A study of networks simulation efficiency: fluid simulation vs. packet-level simulation", *in Proc. of IEEE Infocom 2001.*

[25] V. Misra, W. Gong and D. Towsley, "A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED", *in Proc. of ACM SIGCOMM'00*, Stockholm, Sweden, September 2000.

[26] B. Melamed, S. Pan and Y. Wardi, "Hybrid discrete-continuous flow simulation", in *Proc. of the SPIE Intl. Symp. on Information Technologies and Communication (ITCOM'01)*, Aug. 2001.

[27] S. Floyd, R. Gummadi and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", *under submission*, August 1, 2001.

[28] Curtis Villamizar and Cheng Song, "High performance tcp in ansnet," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 45–60, 1994.

[29] Guido Appenzeller, Isaac Keslassy, and Nick McKeown, "Sizing router buffers," in *Proc. of ACM Sigcomm*, August 2004.

[30] Jeffrey Semke, Jamshid Mahdavi, and Matthew Mathis, "Automatic tcp buffer tuning," in *Proceedings of the ACM SIGCOMM '98 conference*. 1998, pp. 315–323, ACM Press.

[31] Wu-Chang Feng, Dilip Kandlur, Debanjan Saha, and Kang G. Shin, "Blue: an alternative approach to active queue management," in *Proceedings of NOSSDAV*. 2001, pp. 41–50, ACM Press.

[32] S. Floyd and T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 2582, *Internet Engineering Task Force*, April 1999.

[33] A. Gurtov and S. Floyd, "Modeling Wireless Links for Transport Protocols", *in ACM Comp. Commm. Review(CCR)*, to appear.

[34] J. Saltzer, D. Reed and D. Clark, " End-to-End Arguments in System Design", ACM Transactions in Computer Systems, November 1984.

[35] M. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation", *in Proc. of 8th annual international conference on Mobile computing and networking(Mobicom'02)*, Atlanta, Georgia, USA, 2002.