Differential Radio Link Protocol: An Improvement To Tcp Over Wireless Networks

2005

Jaideep Sarkar
University of Central Florida

Differential Radio Link Protocol: An Improvement to TCP over Wireless Networks

by

Jaideep Sarkar
B.E, Electrical and Electronics Engineering
Birla Institute of Technology, India, 2002

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Engineering
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2005

Abstract

New generations of wireless cellular networks, including 3G and 4G technologies, are envisaged to support more mobile users and a variety of wireless multimedia services. With an increasing demand for wireless multimedia services, the performance of TCP becomes a bottleneck as it cannot differentiate between the losses due to the nature of air as a medium and high data load on the network that leads to congestion. This misinterpretation by TCP leads to a reduction in the congestion window size thereby resulting in reduced throughput of the system. To overcome this scenario Radio Link Protocols are used at a lower layer which hides from TCP the channel related losses and effectively increases the throughput. This thesis proposes enhancements to the radio link protocol that works underneath TCP by identifying decisive frames and categorizing them as *crucial* and *non-crucial*. The fact that initial frames from the same upper layer segment can afford a few trials of retransmissions and the later frames cannot, motivates this work. The frames are treated differentially with respect to FEC coding and ARQ schemes. Specific cases of FEC and ARQ strategies are then considered and it is shown qualitatively as how the differential treatment of frames can improve the performance of the RLP and in effect that of TCP over wireless networks.

To my teachers, family and friends....

Acknowledgments

I would like to express my deepest gratitude to all those who have helped in the successful completion of my thesis. First of all I would like to thank my advisor, Dr. Mainak Chatterjee, for his valuable help and guidance and of course his brilliant ideas without which the completion of this work would not have been possible. I would like to thank Dr. Damla Turgut and Dr. Taskin Kocak for their willingness to serve in my thesis committee and for their suggestions to make this work better. My sincere thanks to the UCF Thesis and Dissertation Office especially the editors Katie Grigg and Joanne Muratori for their valuable suggestions towards the final presentation of the thesis. My thanks to the department for helping me whenever it was needed. Finally, I would like to thank my friends and family who made things easier for me during difficult times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

DATA OVER WIRELESS

1.1 Introduction

The explosive growth of wide-area cellular systems and local-area wireless networks and the emergence of home area radio networks and personal area body networks are just the beginning of *-the wireless revolution.* The ultimate goal-uncompromised connectivity and performance for mobile computing devices- requires that we meet the challenge of creating fully integrated, seamless, fault-tolerant and heterogeneous networks composed of fully distributed, energy efficient, and ubiquitous mobile computing platforms. The realization that wireless connectivity profoundly affects the way we compute, communicate, and interact, motivates us to better comprehend all the aspects of the underlying systems and the interactions between them. Making truly tetherless computing possible, demands that we carefully evaluate, enhance, and perhaps re-design our networks, systems, algorithms, and applications. Mobile networks and their wireless links are fundamentally different from conventional stationary, wired computer networks. Mobile connectivity frees communication from the location constraints of the stationary wireline infrastructure. It will allow users to access information anytime, anywhere. Mobile users would like to use the same applications over the wireless link and with the same quality of service (QoS) they are getting over a wired link. TCP/IP is very popular in wired networks. Many researchers (Hasegawa *et al* [27], Faber *et al* [23], Bruyeron *et al.* [13] , Heidemann [28], Mathis *et al.* [40]) studied the performance of TCP and suggested improvements (Rhee *et al.* [46]; Pazos *et al.*[45]). Many others were interested in the modeling and analysis of TCP (Yang [54] , Mo *et al* [41] , Paxon [44]). There are a wide variety of TCP versions used on the Internet. Mobile wireless is one of the most challenging environments for the Internet protocols and for TCP in particular. One approach to supporting the wireless environment is to use a transport protocol, not TCP, which is specifically adapted to the wireless world. These protocols can account for problems associated with the nature of the mobile wireless environment. WTCP by Sinha *et al.* [49] is an example of such a protocol. WTCP is specifically designed for wireless wide area networks. It is a rate based protocol rather than window-based like TCP. It uses a ratio of the average interpacket delay observed at the receiver to the interpacket delay at the sender as the primary metric for rate control, rather than the packet loss and retransmit timeouts used by TCP. Many researchers proposed solutions to improve the performance of TCP over mobile wireless networks. This thesis is an attempt in the same direction to provide seamless interworking between the wired and wireless worlds.

1.2 Wireless Data Services: Associated Problems

To support end-to-end services to wireless and mobile hosts in current and future generation cellular systems, it is necessary that transport layer protocols such as TCP be supported over the wireless links. This is because most of the networks are IP based and TCP still remains the most dominant inter-networking protocol providing reliable end-to-end transmission [19]. However, the design of TCP has been done in such a way that it performs well in wireline networks where the channel error rates are extremely low. Due to the lossy nature of the wireless channels, there are frequent packet losses which are misinterpreted by TCP as congestion related losses and it unnecessarily reduces its transmission window size resulting in reduced throughput. Several schemes have been proposed to alleviate the effects of non-congestion related losses over wireless links [5, 8, 14]. These schemes widely vary in their implementations and depend on the nature of the wireless network, i.e, local area or wide area. Radio link protocol (RLP) [5] is one such mechanism that is particularly meant for cellular networks and has been incorporated into the 2^{nd} and 3^{rd} (2G/3G) systems.

1.3 Motivation

A number of design incompatibilities between the Internet and wireless communication systems have begun to emerge as the Internet connectivity reaches out to the mobile users of cellular systems. We know that the Internet is constructed on the basis of wired communication networks having high reliability (low bit errors) and high transmission capacity (transmission rate is up to gigabits per second). However a cellular system has an unreliable link due to various kinds of interference, noise, fading and low communication capacity due to limited resource of frequency spectrum. To provide seamless Internet services, one of the efforts is to configure a new architecture to support wireless Internet access scenario. The seven layer ISO-OSI hierarchy protocol stack is the basis of design and implementation of the Internet. With the architecture, protocols are designed independently for different layers and it thus simplifies the implementation of protocols that support communications within the same layer. Applying this hierarchy protocol stack to the wireless Internet scenario without any modification is not fully appropriate, as we will find, due to two major characteristics of wireless communications, mobility and wireless access. The proposition of Mobile IP to modify the IP protocol solves the mobility problem. RLP, wireless medium access control (MAC) protocol, and wireless physical equipment are considered for the lower layers to solve the wireless access problem. A growing trend of personal communications indicates the popularity of *data* services over Wireless networks. TCP was designed for traditional wired networks where congestion contributes to most of the packet loss and unusual delay. The protocol responds to packet loss by reducing its transmission window size, activating congestion control algorithm and backing off its retransmission timer [19]. The performance of TCP over wireless links could degrade due to handoff, high bit error rate and long round trip delay on the air interface. The congestion control measures developed for wired networks would cause an unnecessary reduction in network throughput. Several schemes have been proposed to alleviate the effects of non congestion related losses over wireless links. These schemes include radio link protocols (RLPs) [5], fast retransmission [14] and split-TCP connection [8]. the RLP approach where RLP stays below TCP and above the physical layer, has been adopted and implemented for several wireless standards, one of which is cdma 2000 which explained in section 3.1. The throughput of RLP's depends on the rate of channel coding and the rate of transmission due to error [35]. High coding rate and low transmission rate contribute to high throughput. However, the higher the coding rate, the less the capability to correct transmission errors and the higher the rate of retransmissions. It has been shown that a reliable link layer protocol can provide very good TCP performance. This work has been motivated by the fact there is a further scope of improvement at the link layer which is explained in detail in the following sections. An improvement in performance at the link layer will directly affect the performance of the higher layer (TCP) thereby addressing the main issue of implementing high bandwidth data services in wireless networks.

1.4 Contributions Of This Thesis

This thesis demonstrates how the *relative position* or the sequence number of the frames plays an important role in the overall delay performance of the RLP which in turn impacts TCP throughput. We show that the timely delivery of a fraction of the frames are more vital than others, and hence categorize them into *crucial* and *non-crucial*. The crucial frames are those that have greater impact on the delay performance of the RLP. The fraction and the sequence number at which the frames become crucial from non-crucial is found. Differential treatment of the crucial frames with respect to FEC coding and ARQ schemes is then proposed. Both sequential and parallel transmission of frames (i.e., considering single channel and multiple channels) which result in preemptive and non-preemptive transmission respectively is considered for frame transmission. Specific examples of FEC and ARQ schemes are then considered to qualitatively analyze the failure probability, delay and goodput as achieved by the RLP. TCP throughput is shown to improve with the proposed RLP. The improvement is significant when the channel error rates are high. The proposed RLP allows the TCP applications to tune the desired levels of FEC and ARQ so as to obtain a certain level of performance from the RLP.

1.5 Organization Of This Thesis

The rest of the thesis is organized as follows. Background work and some of the important protocols that aim towards improving TCP for wireless channels are explained in detail in chapter 2. The Radio Link Protocol is explained in chapter 3. The basis of our framework is presented in chapter 4 and the existence of crucial frames is shown. Considering both cases of retransmissions, i.e., preemptive and non-preemptive, the fraction of the frames that are crucial is found. In chapter 5, it is qualitatively demonstrated with the help of specific examples of FEC and ARQ how differential treatment of RLP frames can enhance the performance of RLP with the help of three performance metrics- RLP failure probability, delay and goodput. Differential FEC is first considered to show qualitative improvement for each of the metrics. The same principle is then applied for differential ARQ, and then FEC and ARQ are both applied to the crucial frames to show the combined effect. Finally, the effect of fragmentation and the differential treatment of RLP frames on the TCP throughput is shown. Finally, conclusions are drawn in chapter 6.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 TCP For Wired Networks

TCP is a connection-oriented protocol. It has been tuned to perform well for networks composed of wired links and stationary hosts. It is a reliable transport protocol that adapts to the network requirements. It regulates the number of packets it sends by inflating and deflating a window. To do that the TCP sender uses the cumulative acknowledgements (ACKs) sent by the receiver.

TCP also adapts to problems on the wired link. The main problem is the delay caused by packet losses due to congestion. The congestion control scheme in regular (Tahoe) TCP [31] implementation has three main parts:

1. Slow-start
2. Congestion avoidance
3. Fast Retransmit

The Slow-start algorithm works as follows: the TCP sender starts with a congestion window (*cwnd*) that is equal to 1. For each received ACK, TCP exponentially increases the window until it is equal to a threshold (*ssthresh*), then it enters the congestion avoidance phase where it continues to increase its *cwnd* linearly until it reaches the receivers maximum advertised window.

TCP continually measures how long acknowledgements take to return to determine which packets have reached the receiver, and provides reliability by retransmitting lost packets. TCP also assumes that the packet was lost if the sender receives a number of duplicate acknowledgements (usually three).

TCP reacts to any packet lost by:

1. Dropping *ssthresh* into half the current window or 2 (whichever is larger) to reduce the amount of data.

2. Resetting its transmission (congestion) window size to 1, thus activating the slow-start algorithm to restrict the rate at which the window grows to previous levels.

3. Resetting the retransmission timer to a backoff interval that doubles with each consecutive timeout according to Karns exponential timer backoff algorithm [33]. This also results in the reduction of the traffic load at the intermediate links and therefore controls the congestion in the network.

Details of the TCP protocol are provided in [50] and [19] and the TCP congestion control mechanism is explained in detail in Appendix B.

2.2 Wireless Networks

A host is mobile if it is allowed to move freely around a local or wide area network. This allows users to access electronic data and services anywhere and anytime. A

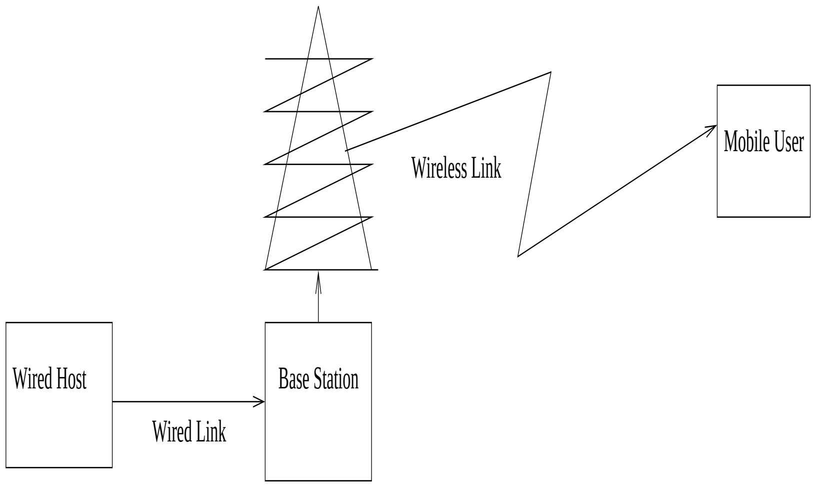

Figure 2.1: Network Topology

user should not be able to differentiate the operation and performance between a mobile and a fixed host (FH). Generally, mobile networks are composed of a wired backbone network and a wireless network. A cellular network infrastructure is used to connect mobile users to the Internet. The wireless network is geographically divided into cells, each of which contains a base station (BS) that provides a connection end-point for roaming mobiles. The base stations are connected to the wired infrastructure. They provide a gateway for communication between the wireless network and the backbone interconnect. As a mobile host (MH) travels between wireless cells, the task of forwarding data between the wired network and MH must be transferred to the new cells BS. This is called handoff. Figure 2.1 illustrates a typical mobile network topology. It is necessary to implement TCP protocols for mobile environments that will provide mobile hosts with the same services that are offered to fixed hosts. Before implementing mobile TCP protocols, we need to know what problems mobile hosts and their wireless links introduce.

2.3 Wireless Networks: Associated Problems

The successful use of mobile computing entails several challenges as mentioned below:

1. *High bit error rates*: wireless links are susceptible to high bit error rates. This leads to the loss of data packets or acknowledgements.

2. *Disconnections*: disconnections can happen due to several reasons:

    - When a mobile moves from one cell into another, the new base station takes over, this is called handoff. During handoff, there is a brief disconnection period.
    - When a mobile host moves out of reach of other transceivers.
    - When radio signals are blocked by buildings and other similar objects.
    - When a cell contains a large number of users and the bandwidth is not enough to satisfy their needs.

3. *Limited and variable bandwidth*: the available bandwidth depends on the location and the number of users in the cell.

4. *Cell size*: determining the suitable cell size requires careful design. Small cell sizes provide high-bandwidth connections, but result in small cell residence time, which leads to frequent disconnections.

5. *Power scarcity*: mobile computers are battery-operated and as a result the power resource is limited. Therefore, it will be helpful if the transmitting and receiving time is minimized.

6. *Dynamic network topology*: the topology of the network changes rapidly due to the movement of mobile hosts

To implement a mobile TCP, we have to consider the above problems. We also have to keep in mind the following:

1. *Non-congestion delay*: Long delays and lost packets in mobile environments are not necessarily due to congestion. Congestion control algorithms need to be used only in the event of genuine network congestion. Note that Jacobson [31] assumes that packet loss due to damage in transit is rare, hence most probably packets get lost due to network congestion and not due to damage. In Jacobson [31], it has been stated that the congestion control scheme is insensitive to damage loss. High loss rates due to damage of one packet per window (e.g., 12-15% for an 8 packet window) degrades TCP throughput by 60%. The additional degradation from the congestion avoidance window shrinking escalates the problem.

2. *Serial timeouts*: Frequent disconnections cause a condition called serial timeouts at the TCP sender. This happens when the retransmission timer at the sender is doubled with each unsuccessful retransmission attempt, in order to reduce the transmission rate. Thus, when the mobile is reconnected, TCP will take a long time to recover from such a reduction and data will not be transmitted for a period of time.

3. *Packet size variation*: packet size over wireless links is typically much smaller than the packet size over wired links. As a result, each packet on the wired networks gets fragmented when transmitted over the wireless link. Therefore, finding the optimal packet size on the wireless link is a key issue for performance.

Since regular TCP was not initially designed with mobile hosts in mind we cannot expect it to perform well in wireless networks. ElAarag [20] found that the TCP sender never recovers its big congestion window size due to the continuous halving of the *ssthresh*. More experiments can be found in ElAarag [21] and ElAarag and Bassiouni [22].

2.4 Protocols That Improve TCP Performance In Wireless Networks

The increasing interest in mobile computers caused researchers like ElAarag and Bassiouni [22], Chandran *et al.* [16], Goff *et al.* [25], Ludwig and Rathonyi [37], Xylomenos and Polzos [53], Chan *et al.* [15], Wang and Tripathi [52], Samaraweera and Fairhurst [47] to be interested in the performance and the improvement of TCP in wireless environments. Inamura *et al.* [30] suggested mitigations to improve the performance of TCP over 2.5G and 3G wireless networks.

The following section summarizes some of the protocols that have been proposed to improve the performance of TCP over wireless networks. These protocols laid the foundation for all subsequent research in this area. We can classify the different proposed protocols into three categories: link layer protocols, end-to-end protocols, and split connection protocols. Figure 2.2 shows the protocols that are employed to improve the TCP performance in the air medium.

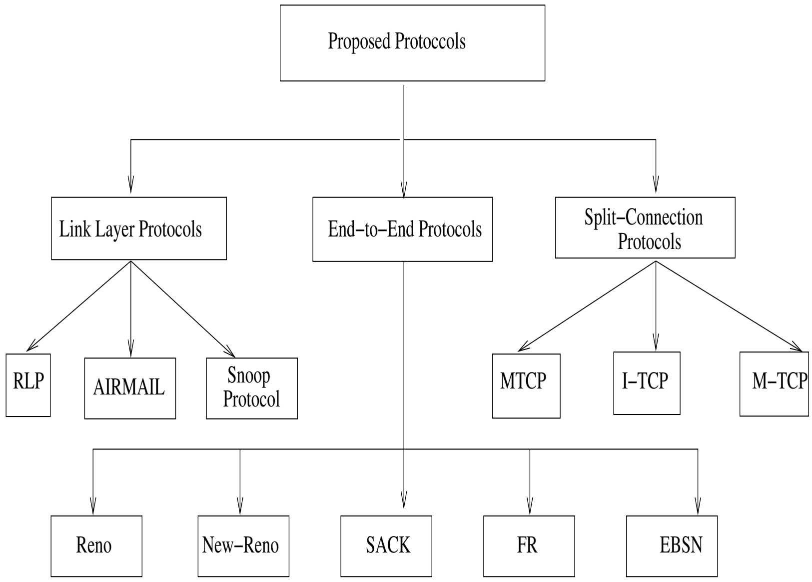

Figure 2.2: Protocols that improve TCP over Wireless Networks.

2.4.1 Link Layer Protocols

This approach tries to increase the quality of the lossy wireless link. Thus, it hides the characteristics of the wireless link from the transport layer and tries to solve the problem at the link layer. The intuition behind link layer protocols is that the problem is local, and hence should be solved locally. They use techniques like forward error correction (FEC) and automatic repeat request (ARQ). There have been several proposals for link-layer protocols. Radio Link Protocol (RLP) was proposed by Nanda *et al.* [42], AIRMAIL by Ayanoglu *et al.* [7] and Snoop by Balakrishnan *et al.* [10]. In this section we shall discuss these three link layer protocols.

- *Radio Link Protocol (RLP)*:Nanda *et al.* [42] proposed a point-to-point automatic repeat request (ARQ) for radio channels. Their protocol exploits inorder delivery of link-layer packets over the radio link. In their basic protocol, a packet is retransmitted only if the transmitter is sure that it was not received. This makes the protocol very efficient in the sense that the receiver gets no more than one copy of any packet. Feedback packets from the receiver together with sequence number of packets and a send sequence number at the transmitter are used to determine whether the packet was received or not. In the basic protocol, the channel may be forced to be idle during periods when all retransmissions have been completed. An enhanced version of their protocol preemptively retransmits unacknowledged packets during this time. This enhancement results in higher throughput and lower delays. Also, the basic protocol requires frequent full receiver state feedback, which is inefficient if user data is to be carried in the reverse direction. So, the enhancement protocol piggybacks partial receiver state in the reverse channel on user data packets. This thesis attempts to improve the performance of RLP by differential treatment of its frames as we will see in the following chapters. Chapter 3 explains in detail the evolution of RLP for 3G systems and its working.

- *AIRMAIL*: Ayanoglu *et al.* in [7] proposed a protocol named AIRMAIL (AsymmetrIc Reliable Mobile Access In Link-layer). The protocol is asymmetric in the sense that the base station is the side responsible for making decisions, whether it is transmitting or receiving. This is because the mobile host has limited battery power and smaller processing capability. Thus the asymmetry places the bulk of the intelligence at the base station with the goal of reducing the processing load at the mobile. The reliability is established by using a combination of automatic repeat request (ARQ) and forward error correction (FEC). The protocol requires the base station to send periodic status messages, while allowing the mobile to combine several acknowledgements into a single one to conserve power. The mobile acknowledgement, unlike the base station, is event driven to reduce the processing load on the mobile. There are three possible levels of FEC: bit-level which is achieved in hardware at the physical layer, byte-level which is done by a per-packet cyclic redundancy check (CRC), and packet-level which is done by allocating some packets for correction which are used for recovery of lost packets without retransmission. Ayanoglu *et al.* showed that a different level of FEC is needed depending on the characteristics of the mobile channel. Therefore, they designed an algorithm that adaptively uses the three levels of channel coding. Thus the bandwidth expansion due to FEC is minimized. The authors also handle handoff by window management and state transfer.

- *The Snoop Protocol*: The Snoop Protocol: Balakrishnan *et al.* [10] aimed to achieve the goal of improving TCP performance without changing the existing TCP implementation in the fixed network. They introduced a module, called Snoop, at the base station that monitors every packet that passes through the connection in either direction. The Snoop module maintains a cache of TCP packets sent from the fixed host that have not yet been acknowledged by the mobile host.Apacket loss is detected either by the arrival of duplicate acknowledgment or by a local timeout. To implement the local timeout, the module has its own retransmission timer. The Snoop module retransmits the lost packet if it has it in the cache. Thus, the base station hides the packet loss from the fixed host, hence avoiding its invocation of an unnecessary congestion control mechanism. The authors improved the performance of the Snoop module by adding selective retransmissions from the base station to the mobile host

2.4.2 End-To-End Protocols

In the end-to-end approach, the TCP sender attempts to handle the losses in a way that improves the performance over regular TCP. Therefore this category maintains the end-to-end semantics of TCP. Reno [31, 50], New- Reno [29] and SACK [40, 24] are different TCP implementations that were initially designed to improve the performance of regular TCP in wired networks. Some researchers for example Balakrishnan *et al.* [10] considered them for wireless networks. Allman *et al.* [6] considered them for networks with satellite channels, which share some of the problems of networks with mobile wireless links. If these implementations were to be used on wireless networks, they will have the advantage that no re-compilation of new software will be needed at the fixed hosts. Comparisons of Reno, New-Reno and SACK in wired networks can be found in Fall and Floyd [24], and in wireless networks can be found in ElAarag and Bassiouni [22] and ElAarag [21]. Caceres and Iftode [14] were among the first to address the impact of mobility and wireless networks on the performance of TCP. Bakshi *et al.*[9] studied the effect of local error recovery and explicit feedback by the base station. However the discussion of these protocols are beyond the scope of this thesis and so have not been included.

2.4.3 Split-Connection Protocols

The main idea behind the split connection approaches is to isolate mobility and wireless related problems from the existing network protocols. This is done by splitting the TCP connection between the mobile host and the fixed host into two separate connections: a wired connection between the fixed host and the base station, and a wireless connection between the base station and the mobile host. In this way the wired connection does not need any changes in existing software on the fixed hosts, and the wireless connection can use a mobile protocol specialized to provide better performance. In what follows, we briefly discuss the following split connection protocols: MTCP, proposed by Yavatkar and Bhagawat [55], I-TCP, by Bakre and Badrinath [8] and M-TCP, by Brown and Singh [12]

- *MTCP*: The basic idea behind MTCP is to protect the long connection over the wired network from the impact of the erratic behavior of the short connection over the wireless link and also recover quickly from errors over the wireless link. Therefore, Yavatkar and Bhagawat [55] introduced a session layer protocol called MHP (Mobile Host Protocol), at the base station and the mobile host. The session layer is above TCP and below the socket. They designed this layer such that it compensates for the unreliability and unpredictability of the wireless link using its knowledge about host migration and wireless links characteristics. They proposed two implementations for the session layer. One uses TCP over the wireless link, and the second uses a selective repeat protocol (SRP) over the wireless link. SRP is designed to recover quickly from high and bursty packet losses. The receiver returns a selective ACK (SACK) when an out of sequence packet is received specifying the missing packet. Then, the sender in turn retransmits the missing packet. SRP also can recover more than one packet in one round trip time.

- *I-TCP*: The same idea was used by Bakre and Badrinath [8]. If a mobile host need to communicate to a fixed host using I-TCP, a request is sent to the current base station to open a TCP connection with the fixed host on behalf of the mobile host. The mobile host communicates with its base station on a separate connection using a variation of TCP that is tuned for wireless links and is aware of mobility. The I-TCP software consists of two components, one on the mobile host and the other on the base station. The component on the mobile host consists of special library calls that are similar in functionality and interface to the socket calls made by an application using regular TCP. This library makes the communication needed with the base station, transparent to the mobile host. The second component consists of a user level Unix process pumping data from one part of the connection into the other. It also handles handoff support for I-TCP.

- *M-TCP*: Brown and Singh [12] focused on the effects of frequent long disconnections and low variable bandwidth on TCP throughput. They also considered the power scarcity of the mobile devices. Thus, they designed a protocol that dynamically assigns a fixed amount of bandwidth to each mobile node based on their changing needs. It performs local error recovery to solve problems resulting from the lossy wireless link. It reduces the power consumption at the mobile node by ensuring that the number of duplicate packets are kept small. The protocol also ensures efficient handoff and deals with the problems caused by long or frequent disconnections. They used a three-layer hierarchical architecture. At the lowest level, there are the mobile nodes and the base station in each cell. Several base stations are controlled by a machine called the supervisor host at a second level of the hierarchy. The supervisor hosts are connected to the wired network at the top level of the hierarchy. The supervisor host handles the routing and maintains connections for mobile users. This protocol is a split connection protocol where the connection between the fixed host and the mobile host is split at the supervisor host. Regular TCP is used on the fixed network (between the fixed host and the supervisor host), while a special version of TCP is used over the wireless link. The TCP client at the supervisor host is called SH-TCP while that on the mobile host is called M-TCP. When the SH-TCP receives data from the sender, it passes it to the M-TCP client, which replies by an acknowledgment. The SHTCP passes this acknowledgement to the sender. To ensure that the sender does not go into congestion control when the ACK does not arrive because the mobile host temporarily got disconnected, the SH-TCP does not forward the ACK of the last byte to the sender until it knows that the mobile host has disconnected. This forces TCP to go into persist mode by setting the window size to zero. Therefore, TCP will not suffer from retransmit timeouts, nor will it close its congestion window. When the mobile host reconnects, the TCP sender is ready to transmit at full speed. If the mobile host did not disconnect but has little available bandwidth. The SH-TCP shrinks the sender's window before it exponentially backs off its retransmission timer. At the M-TCP client, when the mobile host disconnects, it freezes all its timers to ensure that the congestion control is not invoked. When it reconnects, it unfreezes all the timers and resumes normal operation, as if it did not lose any data during disconnection.

CHAPTER 3

RADIO LINK PROTOCOL(RLP)

3.1 Simplified Protocol Stack Of *cdma2000*

We will mainly focus on the link layer of the protocol stack which is outlined in TIA/EIA/IS-2000 [5]. The link layer provides protocols to support and control data transport services. It is divided into two sublayers, the link access control (LAC) and the media access control (MAC) as shown in Figure 3.1. The LAC sublayer provides an interface for transporting data over the air between peer upper layer entities. The LAC employs a number of different protocols to match the quality of service requirements of each upper layer entity to the characteristics of the MAC sublayer in order to provide scalable transmission reliability capabilities. It utilizes various end to end reliable ARQ protocols that use sequence numbering, acknowledgements, and retransmission of lost or damaged packets to provide reliable services. The MAC sublayer provides a control function that manages resources supplied by the physical layer and coordinates the usage of them by various LAC service entities. The MAC sublayer also provides multiplexing and quality-of-service (QoS) control. This can be done by prioritizing requests fairly, and resolving conflict messages. This QoS control mechanism can help to balance the varying QoS requirements of multiple concurrent services. With the ever growing demand of data applications, the RLPs have gone through a series of modifications primarily to address the latency constraints of these applications. The evolution of RLPs and their performance with respect to the CDMA systems is discussed next.

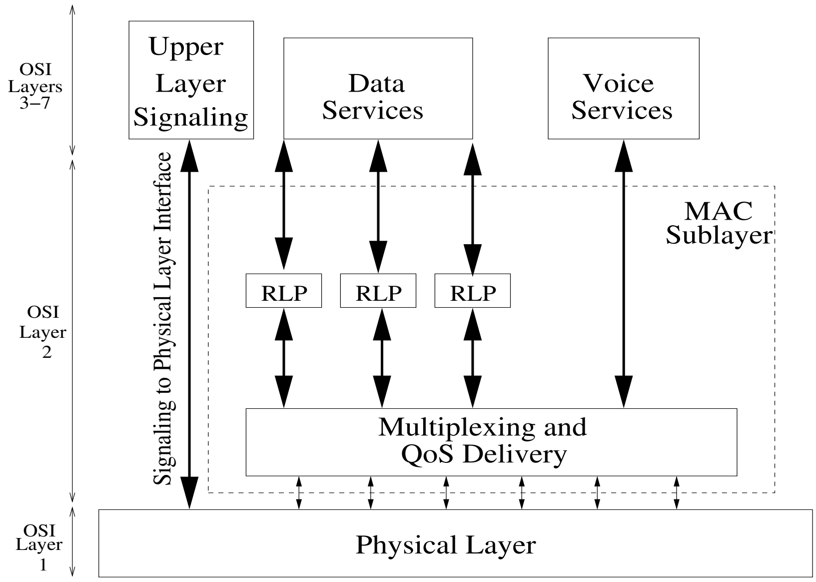

Figure 3.1: cdma2000 protocol stack

3.2 Evolution Of RLP For CDMA Systems

The performance of radio link protocols for various CDMA systems has been studied over the years as the standard evolved. The performance issues related to TCP and RLP interaction in the CDMA protocol stack have been investigated in [11]. The impact of TCP source activity on the call admission control for the cellular CDMA standard IS-95 was studied in [48]. The support of data services over the IS-95 physical channels using RLP was proposed in [26]. For IS-99 (the first IS-95 data standard), the performance evaluation of TCP over RLP was shown in [32] and the performance for circuit mode data services was shown in [18]. Several studies have also been made for the cdma2000 (one of the 3G standards) system [2]. The performance of TCP over the cdma2000 RLP was shown in [34]. A negative acknowledgment based hybrid ARQ scheme was proposed in [51]. In all these standards, RLP has been the *only* layer below TCP to shield the losses by triggering retransmissions, and hence there were some performance limitations. To deal with interactive services or those with stricter delay requirements, it is necessary to incorporate a *fast* retransmission mechanism below the RLP. This was achieved through an ARQ mechanism at the MAC layer, thus providing two layers of retransmission reliability [17]. In [36], the performance of TCP using link and MAC layer retransmissions was evaluated in the presence of correlated fading channels. The benefit of MAC layer ARQ is that retransmissions can be done very quickly without notifying the upper RLP layer.

3.3 The Radio Link Protocol

Radio link protocols are generally employed in the Logical Link Control (LLC) layer, between the physical layer and the TCP layer, to conceal the channel related losses from TCP by quickly recovering the dropped packets by means of local retransmissions. A complete explanation of TCP and its congestion control algorithm has been explained in 2.1. RLP fragments the segments received from TCP into equal sized RLP frames and adds a header to each frame before transmitting over the physical channel. There are various factors which determine the size or the number of RLP frames to be generated from the TCP segment. If the size of the RLP frames is too small, which implies that the number of the RLP frames is large then the overhead due to the header will also be large.

However, if some frames are lost during transmission then only a small portion of the data is lost. On the other hand, when the size of the RLP frames is large (or, the number of frames is small) the overhead due to the header is small. But, the loss of such frames will mean that a considerable portion of the data is lost. The underlying physical layer also imposes restrictions on the size of the RLP frames. The RLP frames cannot be larger than what the physical layer frame can accommodate. Thus an optimal solution is chosen regarding the number of frames that need to be created from a TCP segment.

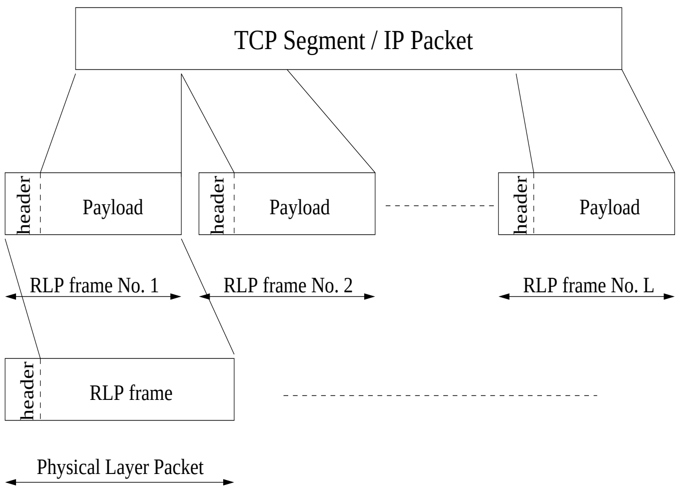

Figure 3.2: Fragmentation of TCP Segment to RLP frames

During the data transfer phase, RLP maintains the sending sequence number count $L_{V(S)}$ and two sequence numbers for receiving, $L_{V(R)}$ and $L_{V(N)}$. All operations are carried out modulo arithmetic. $L_{V(S)}$ is incremented every time a non-zero frame is sent out(it is the sequence number of the next new frame to be sent). $L_{V(R)}$ is the sequence number of the next frame expected to be received and $L_{V(N)}$ is the oldest sequence number of the missing frames. Let j be the sequence number(SEQ) of the newly received frame, the RLP transmission procedure can be described as follows-

1. If $SEQ < L_{V(N)}$, or if the frame is already stored in the resequencing buffer, discard.

2. If $SEQ = L_{V(N)}$, update $L_{V(N)}$ to the next oldest missing frame sequence number. Pass received frames up to $L_{V(N)} - 1$ to the upper layer

3. If $L_{V(N)} < SEQ < L_{V(R)}$, store frame SEQ in the resequencing buffer if it is missing.

4. If $SEQ = L_{V(N)} = L_{V(R)}$, pass all frames received upto $L_{V(R)}$ to the upper layer.

5. If $SEQ = L_{V(R)}$ ($L_{V(N)}$ or $SEQ > L_{V(R)}$), increment $L_{V(R)}$ and store frame SEQ into resequencing buffer.

6. Update the NAK list

7. For all cases, send NAKs of missing frames if their retransmission timers are not yet set or expired.

In case of a RLP frame loss during transmission, the RLP uses an Automatic Repeat reQuest (ARQ) mechanism to recover the lost or damaged frames. RLP uses a timer function for invoking the retransmissions in case of a loss. This timer value is much smaller than the TCP timeout and allows the RLP to quickly recover dropped or erroneous frames before the TCP timer expires. The RLP is allowed a finite number of retransmissions for the same frame. The RLP aborts the frame recovery process once the allowed number of retransmissions is exhausted. If RLP fails to recover a frame, it hands over the segment (with missing frames) to the upper layer, i.e., TCP, which then starts its own retransmission scheme to recover the damaged segment.

The RLP can use a number of retransmission schemes like (1,1,1,1,1), (1,2,3), (1,1,2,3), etc [18], depending on the channel conditions and the performance required for the session it is supporting. For example, the (1,2,3) scheme uses three trials of retransmission with one copy of the frame being retransmitted in the first trial, two copies in the second trial and three copies in the third and final trial.

3.4 HARQ: Hybrid ARQ

Oftentimes, hybrid ARQ's [39] are also used to enhance the performance of RLPs. Hybrid ARQs incorporate certain forward error correcting (FEC) schemes through which it ensures that there is a higher probability of the packets reaching the receiver end. The transmitter on the receipt of a NACK or a time-out will trigger a retransmission. It might so happen that a packet which has been successfully received had the ACK damaged. In that case, the transmitter will time-out and re-send the packet resulting in duplication of the packet at the receiver buffer. Now the question arises about the effect of the retransmission, *whether the retransmitted packet is on-time at the re-sequencing buffer for it to be passed on to the higher layers.* This can be only possible if the round trip time (RTT) is sufficiently low and the packet can be accommodated in the re-sequencing buffer. If the RTT is high and retransmission is not feasible, then the RLP frames can be made more robust by adopting *forward error correction* (FEC) schemes.

CHAPTER 4

IDENTIFYING CRUCIAL FRAMES

4.1 Introduction

The main motivation of our proposition is due to the fact that existing RLPs do not differentiate the frames obtained from the *same* TCP segment. This leads us to provide differential treatment to the RLP frames. We believe that the importance of each RLP frame from the same TCP segment is different and hence deserves differential treatment. Our claim is based on the fact that the reassembly of the RLP frames can only be done when *all* the frames belonging to the same TCP segment are correctly received by the receiver. The basic philosophy is that the *last* frame of a particular segment will decide the time of delivery of the reassembled TCP segment to the upper layer. Note that the last frame need not be the last one in terms of the sequence number, but the last frame to be *received correctly* (among the frames from the same TCP segment). However, under ideal channel conditions (i.e., no frame loss) the last frame in terms of the sequence number will arrive last simply because of the sequential nature of the transmission as shown in Figure 4.1.

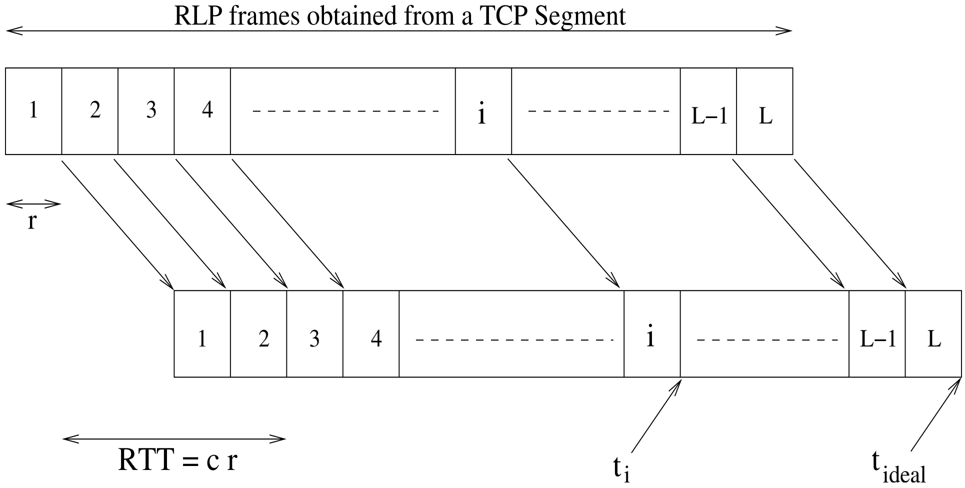

Figure 4.1: Error-free transmission of RLP frames

4.2 Obtaining Index Of Cruciality

Suppose a TCP segment under consideration is fragmented into L frames. It can be seen from figure 4.1 how the L RLP frames obtained from a TCP segment are transmitted and received, the last (Lth) frame arriving at time t_{ideal}. At this time, all the L frames are ready to be reassembled to form the TCP segment. But in a realistic situation, frames will get dropped and due to retransmissions of the erroneous frames, the i^{th} RLP frame might successfully arrive last, where $1 \leq i \leq L$. If i happens to be one of the initial frames of the TCP segment then few trials of retransmission are possible because the retransmissions would be complete before the last RLP frame successfully arrives at the receiver. But retransmissions of the later frames might delay the delivery of all the reassembled TCP segment to the upper layer. It is these later frames which are more important in deciding the total delay for the reassembly of all the RLP frames to form the TCP segment. Hence we call these frames as *crucial*, and the others as *non-crucial*. Next, we identify the fraction of total frames that are crucial.

Table 4.1: Notations

T_{rtt}	Round trip time
r	Transmission time of a RLP frame
p	RLP frame error rate
L	No. of RLP frames form one TCP segment
$t_e(i)$	Expected time of arrival for ith frame
c	$\lceil T_{rtt}/r \rceil$
k	index of last non-crucial frame

We consider two cases of retransmission: *non-preemptive* and *preemptive*. In the non-preemptive case, we assume the use of multiple transmitting channels, where the retransmission of a given frame does not delay the transmission of another frame. In the preemptive case, we assume the use of a single channel where retransmissions are given highest priority and preempts the transmission of the next frame. Let us consider these two cases in detail.

4.3 Crucial Frames: Non-Preemptive Transmissions

Let us assume that the size of a TCP segment be T bytes and it is fragmented into equal sized RLP frames. The number of RLP frames obtained from a TCP segment would be $L = \lceil \frac{T}{R_p} \rceil$, where R_p is the payload of each RLP frame. The actual size of the RLP frame would be R_p plus some header information and some redundancy checks. (Frequently used notations in the paper are shown in Table 4.1) As shown in Figure 4.1, the time at which all the L frames are received at the receiver under ideal channel conditions (i.e., no frame loss) is denoted by

t_{ideal}. Considering successive and pipelined transmission we obtain

$$t_{ideal} = Lr + T_{rtt}/2 \tag{4.1}$$

where T_{rtt} is the round trip time of each RLP frame and r is the transmission time of each RLP frame. Thus, it is the ideal (best possible) time of arrival of an entire segment. T_{ideal} consists of two components. First, the transmission time of all the frames back to back. Second, the propagation time for the last frame (or, half the round trip time). The acknowledgement from the receiver is not considered as we know for sure that the frames will go through.

However, when frames get dropped or corrupted with a probability p, the correct reception of frames will get delayed due to retransmissions. If we do not restrict the number of allowed retransmissions then the expected delay, D, of a lost frame would be given by

$$\begin{aligned} D &= (1-p)\frac{T_{rtt}}{2} + p(1-p)\frac{3T_{rtt}}{2} + p^2(1-p)\frac{5T_{rtt}}{2} + \cdots \\ &= \frac{T_{rtt}(1+p)}{2(1-p)} \end{aligned} \tag{4.2}$$

In reality, the allowed number of retransmission is finite (usually 3). The inclusion of the higher terms would have negligible effect since p is much smaller than 1. The exact value for D can always be obtained with the series truncated.

This value of D is used to identify crucial and non-crucial frames. The frames which have D greater than t_{ideal} are defined as the crucial frames since these frames have profound effect in increasing the total re-assembly delay. Otherwise, the frames are non-crucial.

Certain protocols like High speed packet data access (HSDPA) [4] used in WCDMA [1] systems have defined their own ARQ mechanisms. In order to reduce receiver buffering requirements, the ARQ scheme is based on a N-channel stop-and-wait protocol. In this scheme, a transmission can use upto N channels to transmit the data. For each channel, there will be a separate ARQ process. So the total segment transmission time will depend upon how the channels are allocated to a user.

Let us denote the expected time of arrival of the ith frame as $t_e(i)$. The expected time of arrival of the first frame at the receiver is

$$t_e(1) = D \tag{4.3}$$

The expected time of arrival of the second frame with respect to the transmission start time of the first frame will be

$$t_e(2) = D + r \tag{4.4}$$

where r is the transmission time of each RLP frame. Similarly, the expected time of arrival of the ith frame with respect to the first frame is

$$t_e(i) = D + (i-1)r \tag{4.5}$$

We can thus find the frames whose expected time of arrival at the receiver will be more than t_{ideal}. The time required to reach the receiver can be found by equating t_{ideal} and $t_e(i)$. Thus,

$$D + (i-1)r = Lr + T_{rtt}/2 \tag{4.6}$$

Substituting for D and using $c = \lceil T_{rtt}/r \rceil$, we get

$$i = \lceil L + c/2 - \frac{c(1+p)}{2(1-p)} + 1 \rceil \tag{4.7}$$

Thus, for different values of k, p and L, we can find the frame number i. Frame number i gives the starting index for the crucial frames. Of course, the fraction of crucial frames will vary under different conditions. The fraction of crucial frames is simply defined as the ratio of the number of crucial frames to the total number of frames, i.e., $\frac{L-(i-1)}{L}$.

4.3.1 Fraction Of Crucial Frames

The fraction of crucial frames are shown in Figures 4.2 and 4.3 for $T_{rtt} = 20$ and 30 respectively. The number of RLP frames from each TCP segment L, varies from 0 to 100. The frame error rate p is assumed to be 10%, 20% and 30%. We observe that as the number of frames, L increase, the fraction of crucial frames decreases, suggesting that most of the frames will be recovered even if they suffer many retransmissions before the RLP timer expires. The actual problem arises when L is small resulting in a larger fraction of crucial frames. As expected, it is observed that as the channel error condition improves the fraction of crucial frames decreases. For a smaller value of c (Figure 4.2) the fraction of crucial frames is smaller compared to larger value of c (Figure 4.3).

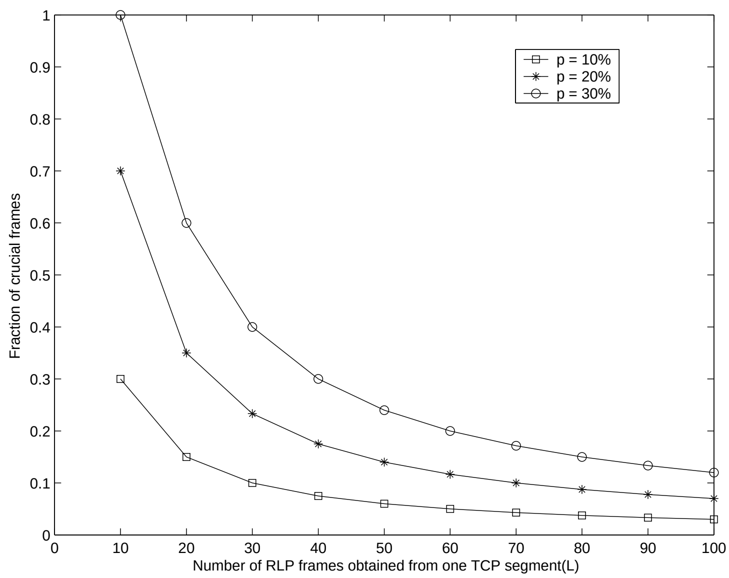

Figure 4.2: Fraction of crucial frames for $T_{rtt} = 20$

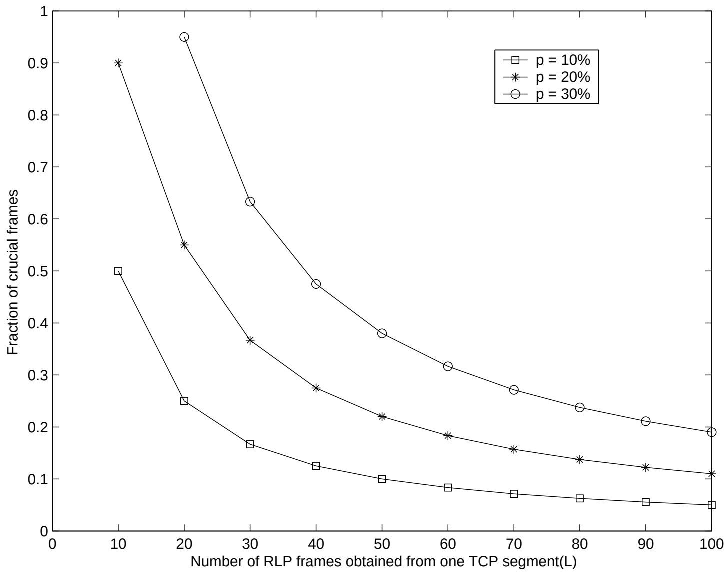

Figure 4.3: Fraction of crucial frames for $T_{rtt} = 30$

4.4 Crucial Frames: Preemptive Transmissions

So far we have considered that original and retransmitted frames use different channels and hence there is no inter-dependency of transmission sequence. However, if original and retransmitted frames use the same channel the retransmitted frames are given higher priority and delay the transmission of original frames. For example, in Figure 4.4, we observe that frame number 4 is delayed due to the retransmission of frame number 1, but frames 2 and 3 are transmitted before the NACK for frame 1 can arrive.

We consider that time is slotted where each slot corresponds to the transmission time of one RLP frame. Let $c = \lceil T_{rtt}/r \rceil$ denote the slots after which retransmission occurs once the original transmission is corrupted. Note that $c = 3$ in Figure 4.4. In a realistic case, each retransmission of a given RLP frame delays the delivery time of future RLP frames. Let us consider the ith RLP frame. The

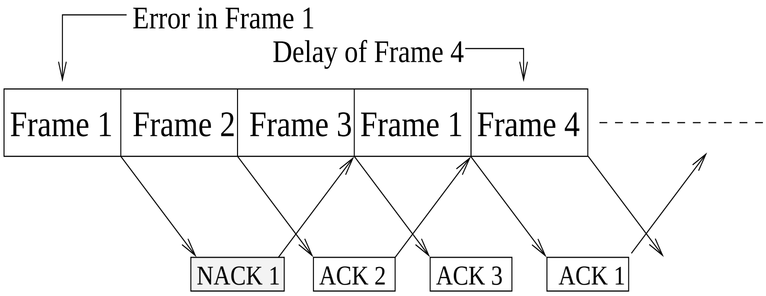

Figure 4.4: Preemptive scheduling

total expected delivery time for this frame can be expressed from the linearity of expectations as

$$t_e(i) = t_e^d(i) + t_e^r(i) \tag{4.8}$$

where $t_e^d(i)$ denotes the expected delay in sending the original ith RLP frame and $t_e^r(i)$ denotes the delay of the ith frame due to retransmissions.

In order to calculate $t_e^r(i)$ we first note that a retransmitted frame is never delayed more than single slot, as the frame is retransmitted immediately at the next slot and no more than one NACK can arrive together. When there is no restriction in the number of retransmissions, we get

$$t_e^r(i) = \frac{T_{rtt}(1+p)}{2(1-p)} \tag{4.9}$$

It can be noted that each retransmission leads to an increase in $(c+1)$ slots.

Now let us consider the expression for $t_e^d(i)$. The minimum time t_{min} taken to transmit the ith RLP frame corresponds to the situation where all previous RLP frames $(1 : i-1)$ are transmitted successfully. Therefore, the minimum time taken to transmit the ith RLP frame is $(i-1)r$. Also, due to the failures and the subsequent retransmissions from the $(1 : i-1)$ frames, there would be additional delay. Let the delay be represented by Δ and the expected delay by $E(\Delta)$.

Therefore, we can write

$$t_e^d(i) = (i-1)r + E(\Delta) \tag{4.10}$$

However not all retransmissions of previous RLP frames add to the delay. For example, if $c = 4$, retransmissions requests for RLP frames $(i-1 : i-3)$ can arrive only after $(i-1)r$. Now if all previous RLP frames $(1 : i-4)$ were transmitted successfully, Δ will be zero independent of what happens to RLP frames $(i-1 : i-4)$. Based on this observation, we can find the probability of Δ being 0, i.e.,

$$Pr(\Delta = 0) = (1-p)^{i-1-c}. \tag{4.11}$$

In the above equation, if $i-1-c \leq 0$, $Pr(\Delta) = 1$. For $\Delta = m$ implies that there exist m retransmissions. One can always consider the failure event for original and retransmitted event as independent. This suggest that failure event has happened from RLP frames (original or retransmitted) transmitted in $1 : i+m-c$ slots. So we express

$$Pr(\Delta = m) = p^m(1-p)^{i-c-1}. \tag{4.12}$$

Therefore,

$$\begin{aligned} E(\Delta) &= r\sum_m mp^m(1-p)^{i-c-1} \\ &= r(1/(1-p)2)(1-p)^{i-c-1} \\ &= r(1-p)^{i-c-3} \end{aligned} \tag{4.13}$$

Thus equation (4.10) reduces to

$$t_e^d(i) = (i-1)r + r(1-p)^{i-c-3} \tag{4.14}$$

Adding equations (4.14) and (4.9), we get

$$t_e(i) = (i-1)r + r(1-p)^{i-c-3} + \frac{T_{rtt}(1+p)}{2(1-p)} \tag{4.15}$$

Due to the transcendental nature of this equality, i cannot be obtained directly. Therefore, we use numerical methods to solve for i.

4.4.1 Fraction Of Crucial Frames

We solve for i by performing a binary search, since the right side of equation (4.15) is monotonically increasing with i. The solution for i is shown in Figures 4.5 and 4.6 for $T_{rtt} = 20$ and 30 respectively. We observe from the plots that there exists slight inflection at a certain value of L for every curve. This inflection point is due to the value of c. The fraction of crucial frames before the inflection point follows the pattern as in the case of parallel transmission. After the inflection point the fraction of crucial frame is increased because of additional delay due to sequential transmission.

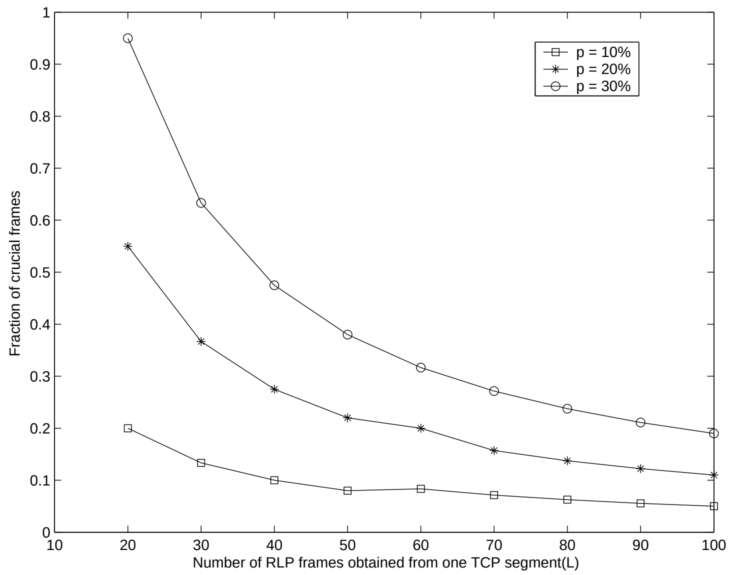

Figure 4.5: Fraction of crucial frames for $T_{rtt} = 20$

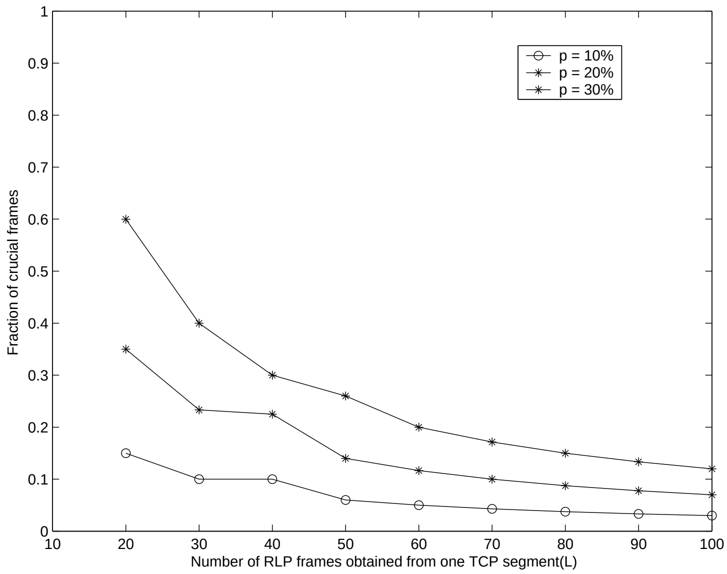

Figure 4.6: Fraction of crucial frames for $T_{rtt} = 30$

In the subsequent presentation of the paper, we will just use the value of i as obtained from either Equation (4.7) or (4.15) as per the situation.

CHAPTER 5

DIFFERENTIAL TREATMENT OF FRAMES

5.1 Introduction

So far we have identified a fraction of frames emanating from the same TCP segment that are more crucial than others, in determining the performance of the RLP. Now, we would like to treat these crucial frames differently compared to the non-crucial ones. Currently, the existing RLPs use the same channel coding and ARQ mechanism for all the frames. In this paper, we do not propose any new channel coding scheme or ARQ mechanism but rather show comparatively how the performance of the RLP could be improved if we were to use different channel coding schemes and ARQ mechanisms for different frames. We propose differential FEC and ARQ treatment for the RLP frames. Both the FEC and the ARQ schemes to be applied to a frame will depend on the index of that frame.

In obtaining equation (4.2), we assumed that the number of retransmissions allowed was infinite, and therefore all packets were eventually recovered. However, in reality this is not true. In most cases, the maximum number of retransmissions allowed is three, but the manner (i.e., the number of copies transmitted in each trial) in which these three trials are done is different. Due to the finite number of retransmissions trials there is no guarantee that a frame will be recovered by the RLP; and therefore, the RLP will fail with a certain probability. Let us formally define RLP failure probability along with the two other metrics of interest - delay and goodput.

- *RLP failure probability:* RLP failure probability is defined as the probability of the RLP failing to deliver *all* the L frames within its allowed number of retransmissions as a result of which the recovery mechanism will be handed over to the upper layer (e.g., TCP).

- *Delay:* Delay for a frame is defined as the time taken for that frame to be received correctly at the receiver with respect to the *first* frame's transmission time i.e, the time the RLP started transmitting the first frame from a particular TCP segment.

- *Goodput:* Goodput is defined as the ratio of the actual number of information bits decoded correctly at the receiver to the total number of bits transmitted.

For the ease of understanding, let us consider specific cases for the qualitative analysis of the above metrics. However, they can easily be extended to a more generalized scenario. It can be noted that for all the analysis that follows, a certain FEC and ARQ scheme was assumed. We show how the differential treatment of the frames affects the performance of the RLP. Let us now discuss the differential FEC and differential ARQ schemes independently before we discuss the combined mechanism.

5.2 Providing Differential FEC To The Crucial Frames

Redundant bits in the form of FEC codes are usually added to the payload of the RLP frame for detection and possible correction of transmission errors. The correction capability of these codes will depend on the kind of codes and the length of the code used. Since this paper does not deal with FEC codes, the simplest simplest of codes- *block codes* will be used. In block codes, M redundancy bits are added to the information bearing N bits. (Note that the extra M bits are generated using a generator matrix operating on the N bits.) If we consider a RLP frame of $N + M$ bits, then the resulting bit loss probability is given by [38]

$$b = \sum_{j=M+1}^{N+M} \binom{N+M}{j} (1 - b_{pl})^{N+M-j} b_{pl}^{j} \frac{j}{N+M} \tag{5.1}$$

where b_{pl} is the bit loss probability before decoding.

The idea behind this expression is that even if it is not possible for the decoder to receive all the bits in a block, it can still deliver the received bits to the application. Since j is the number of lost bits and $N + M$ is the number of sent bits, the last factor gives the bit loss probability without which the expression would simply give the probability of having more loss than can be recovered by the block. Of course, different FEC schemes will yield different loss probabilities. From this equality (or any such relation between M and b) we can calculate the number of redundant bits to be added to achieve a desired loss probability. As discussed earlier, the initial (non-crucial) frames can afford a few trials of retransmission without adding substantial delay to the RLP reassembly. Therefore, The FEC coding need not be very robust for these non-crucial frames. This will also reduce the overhead since the redundancy bits will be less. On the other hand the later packets are more crucial and retransmission should be avoided or minimized. One way to avoid or minimize loss of the crucial frames is to use stronger FEC codes. We will not deal with the specifics of different codes but will assume simple block codes with varying redundancy to achieve the desired degree of robustness against errors.

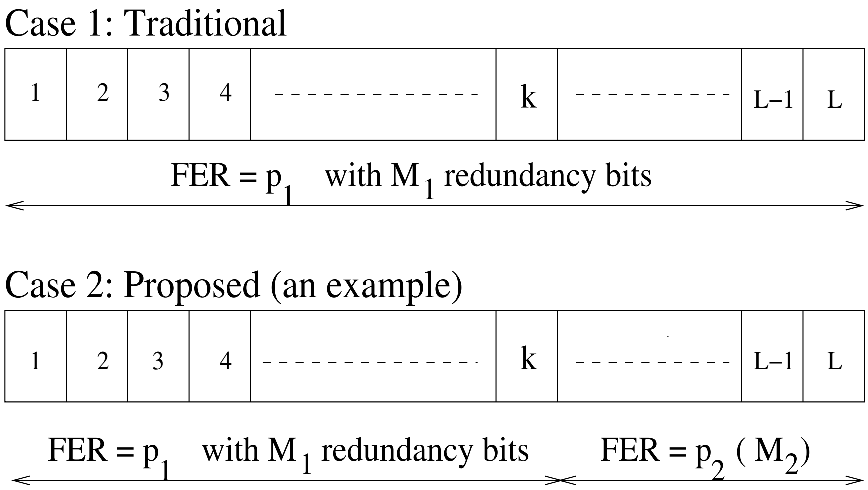

Figure 5.1: Different FEC schemes

Let us assume that a traditional RLP uses M_1 bits to code each frame as shown in Case 1 of Figure 5.1. It can be noted that each of the L frames is coded with the same number of bits, i.e., M_1, because of which the FER observed is p_1. The RLP is made aware of the crucial frames and it encodes each of the crucial frames using M_2 bits, where $M_2 > M_1$, as shown in Case 2. This usage of more redundancy bits will result in FER $= p_2$, where $p_2 < p_1$. The exact reduction in the FER will depend on the values of M_1, M_2, N, and the kind of coding used. We assume that the ARQ scheme used is (1,1,1).

5.2.1 RLP Failure Probability

We need to calculate the probability that all the L frames will not be correctly received at the receiver. For Case 1, where the FER is p_1, the RLP failure probability (F_1) is simply given by

$$F_1 = 1 - \left(1 - {p_1}^3\right)^L \tag{5.2}$$

For the example in Case 2, where the first k frames experience a FER of p_1 and the last $L - k$ frames experience a FER of p_2, the RLP failure probability (F_2) is given by

$$F_2 = 1 - \left(\left(1 - {p_1}^3\right)^k \times \left(1 - {p_2}^3\right)^{L-k}\right) \tag{5.3}$$

Due to stronger FEC in Case 2, the RLP is able to recover more frames than in Case 1.

5.2.2 Delay

Recall that the expected delay of the ith frame is $D+(i-1)r$, where $D = \frac{T_{rtt}(1+p)}{2(1-p)}$. For Case 1, the delay at the RLP is

$$D_1 = \frac{T_{rtt}(1 + p_1)}{2(1 - p_1)} + (L - 1)r \tag{5.4}$$

For Case 2, we treat the crucial and non-crucial frames separately, because the frames would experience different loss rates. Irrespective of the losses, successive frames are always transmitted, i.e., crucial frames would not be prevented from transmission even if all the non-crucial frames are not received correctly. The last non-crucial (kth) frame is expected to arrive correctly after a delay of

$$D_{nc} = \frac{T_{rtt}(1 + p_1)}{2(1 - p_1)} + (k - 1)r \tag{5.5}$$

Similarly, the last crucial (L) frame is expected to arrive correctly after a delay of

$$D_c = kr + \frac{T_{rtt}(1+p_2)}{2(1-p_2)} + (L-k-1)r \tag{5.6}$$

The term kr is the time it takes to transmit the non-crucial frames (not necessarily correctly) before the crucial frames are transmitted. It is not known which frames would arrive later because the exact relation between p_1 and p_2 is not known. Therefore for Case 2, the delay at the RLP is

$$D_2 = \max(D_{nc}, D_c) \tag{5.7}$$

Although, the error performance of the transmission is improved by adding the redundancy bits, the goodput is compromised as discussed next.

5.2.3 Goodput

It is to be noted that the goodput is $(1-p)\frac{N}{N+M}$ when the frame reaches in its first transmission. However, the goodput obtained after jth retransmission trial, $1 \leq j \leq 3$, will depend on the probability of previous failures and the total number of frames transmitted to eventually recover that frame. Therefore, for the jth retransmission, the goodput will be $\frac{1}{j+1}p^j(1-p)\frac{N}{N+M}$. Thus, the goodput due to the original transmission and three retransmissions in Case 1 would be

$$G_1 = (1-p_1)\frac{N}{N+M_1}.$$

$$G_1 = \sum_{j=0}^{j=3} \frac{1}{j+1} p_1^j (1-p_1) \frac{N}{N+M_1} \tag{5.8}$$

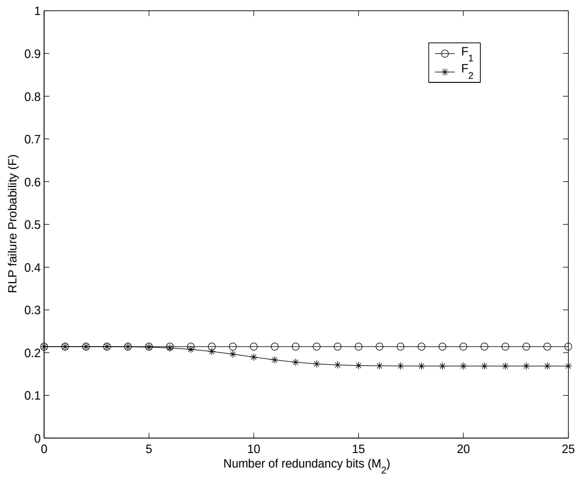

Figure 5.2: RLP failure probability vs. Number of redundancy bits

We also define *throughput* as the fraction of the information bits correctly transmitted to the Similarly, the goodput for Case 2 is

$$G_2 = (\frac{k}{L})G_1 + \sum_{j=0}^{j=3} \frac{1}{j+1} p_2^j (1-p_2) \frac{N}{N+M_2} \tag{5.9}$$

As expected, it can easily be verified that there is a reduction in goodput in Case 2, i.e., $G_2 < G_1$.

5.3 Numerical Results For Differential FEC

Let us discuss the performance of the RLP with respect to the three metrics when the proposed differential FEC is applied to the crucial frames. We assume $L = 30$, T_{rtt}=20 and p=20% in calculating the results. It is also assumed that the number of information bits per frame (N) is 50 and the number of redundancy bits M_2 is varied from 0 to 25. M_1 was maintained at 0, implying that no FEC was applied to the non-crucial frames.

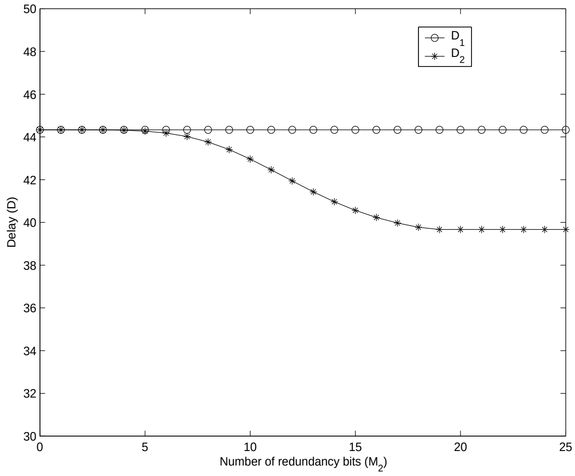

Figure 5.3: Delay vs. Number of redundancy bits

As expected, we observe that there is an improvement in the RLP failure probability with the increase in the redundancy bits as shown in Figure 5.2. This ensures that the RLP is more effective in recovering the lost frames and thereby preventing the information of losses propagating to TCP. Of course, the improvement saturates beyond $M = 20$ which confirms that the error correcting capabilities of

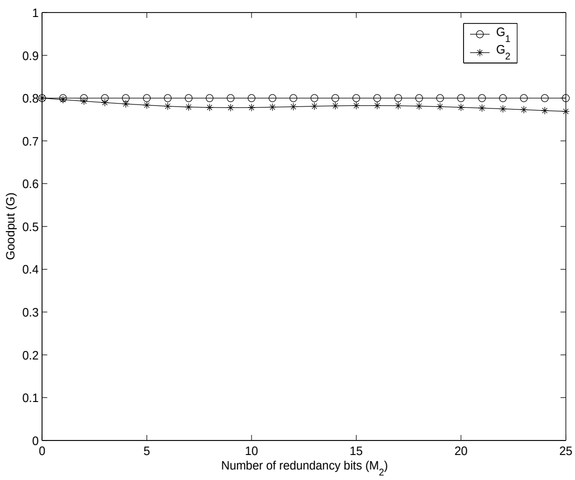

Figure 5.4: Goodput vs. Number of redundancy bits

FECs are bounded. We observe from Figure 5.3 how the delay performance of the RLP improves because of adding more robustness to the crucial frames. We also observe that there is a variation in delay with the increase in redundancy bits of the FEC scheme used. From the goodput perspective it is advisable that the redundancy is not made arbitrarily large which we see from Figure 5.4. There is slight reduction in the goodput with the increase in redundancy. This is the trade-off for better RLP failure probability and delay performance. It is interesting to note the oscillatory nature of the goodput curve G_2. This is because, initially the goodput decreases due to the increase in redundancy bits, however when the redundancy bits are further increased, then the recovery capability of the lost frames are enhanced resulting in increased goodput. The goodput at any M depends on which factor dominates– the goodput or the error recovering capability.

5.4 Applying Differential ARQ To The Crucial Frames

In the differential FEC case (Section 5.2) we considered that the underlying ARQ scheme was (1, 1, 1) but with varying number of redundancy bits. Now, we consider two different ARQ schemes– (1, 1, 1) for Case 1 and (1, 2, 3) for Case 2 as shown in Figure 5.5. We also assume that the FEC codes used is uniform across all the frames (say M bits per frame), therefore all frames would experience a FER of p (say). Of course, these can be generalized with the only condition that the crucial frames in Case 2 must have a stronger ARQ than the non-crucial ones.

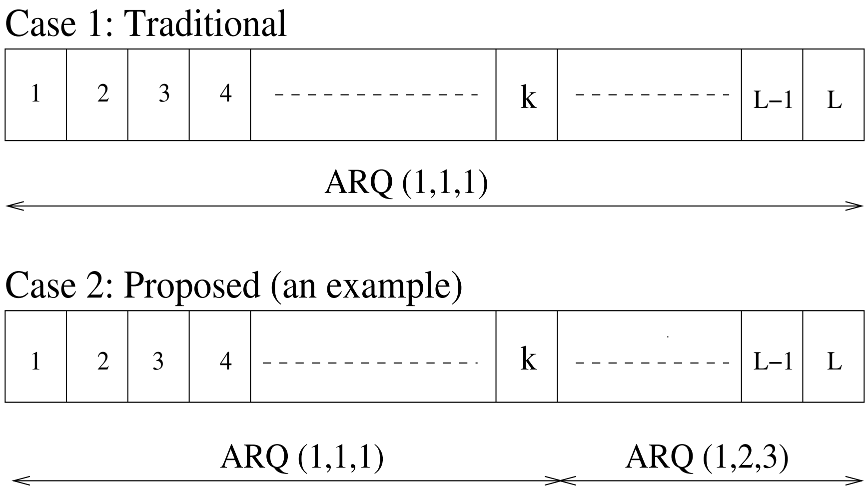

Figure 5.5: Different ARQ schemes

5.4.1 RLP Failure Probability

The RLP failure probability (F_1) for the example in Case 1 is obtained as

$$F_1 = 1 - (1 - p^3)^L \tag{5.10}$$

For the example in Case 2, where the probability of correctly receiving a non-crucial frame is $(1-p^3)$ and a crucial frame is $(1-p^6)$, the RLP failure probability (F_2) is

$$F_2 = 1 - \left((1 - p^3)^k \times (1 - p^6)^{L-k}\right) \tag{5.11}$$

As expected, we observe that $F_2 < F_1$.

5.4.2 Delay

We calculate the delay for the frames which are ultimately decoded correctly at the receiver. The frames which do not, are accounted for in the RLP failure probability. The expected delay for each non-crucial frame will be due to the delay contributions from the original transmissions and the three trials of retransmission. For Case 1, where the ARQ scheme is (1,1,1), the expected delay for any frame is the expected delay due to the initial transmission and possibly the three retransmissions. The total expected delay (D_1) in getting all the L frames in Case 1, is dictated by the last frame to arrive at the receiver. If we calculate the expected time of arrival for all the L frames then we find that the expected delay for the Lth frame equals the delay for the first frame plus the delay in transmitting the Lth frame. Thus, we get

$$D_1 = (1-p)\frac{T_{rtt}}{2} + p(1-p)\frac{3T_{rtt}}{2} + \\ p^2(1-p)\frac{5T_{rtt}}{2} + p^3(1-p)\frac{7T_{rtt}}{2} + (L-1)r \tag{5.12}$$

For Case 2, we obtain the expected delay for the non-crucial frames (1 through k) in the same manner as Case 1. The expected delay (D_{nc}) for the non-crucial frames is

$$D_{nc} = (1-p)\frac{T_{rtt}}{2} + p(1-p)\frac{3T_{rtt}}{2} + \\ p^2(1-p)\frac{5T_{rtt}}{2} + p^3(1-p)\frac{7T_{rtt}}{2} + (k-1)r \tag{5.13}$$

The calculation for the crucial frames ($k+1$ through L) will be a little different because of the ARQ scheme. The expected delay for any crucial frame would be $(1-p)\frac{T_{rtt}}{2} + p(1-p)\frac{3T_{rtt}}{2} + p^2(1-p2)\frac{5T_{rtt}}{2} + p^4(1-p3)\frac{7T_{rtt}}{2}$. Therefore, expected delay (D_c) from crucial frames would be

$$D_c = (1-p)\frac{T_{rtt}}{2} + p(1-p)\frac{3T_{rtt}}{2} + \\ p^2(1-p^2)\frac{5T_{rtt}}{2} + p^4(1-p^3)\frac{7T_{rtt}}{2} + (L-1)r \tag{5.14}$$

The term $(L-1)r$ appears because we are calculating the delays with respect to the time of transmission of the first frame. Since, we do not know the values of the variables used, we cannot determine for sure whether the non-crucial or the crucial frames arrive later. Physically it means, the arrival of the frames would depend on the observed FER and also on the ARQ used. Therefore, the overall delay for Case 2 is simply determined by finding the greater of D_{nc} and D_c as the time of reassembly will depend on the last frame that arrive at the receiver. Therefore,

$$D_2 = max(D_{nc}, D_c). \tag{5.15}$$

5.4.3 Goodput

Following the logic from the earlier goodput calculation, the goodput for Case 1 would be

$$G_1 = \sum_{j=0}^{j=3} \frac{1}{j+1} p^j (1-p) \frac{N}{N+M} \tag{5.16}$$

For Case 2, we consider (1,1,1) and (1,2,3) schemes for the non-crucial and crucial frames respectively. We observe that the goodput for the first k frames would be the same as in Case 1. The goodput for the crucial frames will again depend on the probability of previous failures and the total number of frames transmitted to eventually recover that frame. For the original transmission the goodput will be $(1-p)\frac{N}{N+M}$. For the three retransmissions the goodput will be $\frac{1}{2}p(1-p)\frac{N}{N+M}$, $\frac{1}{4}p^2(1-p^2)\frac{N}{N+M}$ and $\frac{1}{7}p^3(1-p^3)\frac{N}{N+M}$ respectively. So, the total goodput for all the L frames in Case 2 would be

$$G_2 = \frac{k}{L} G_1 + \frac{L-k}{L}\Big((1-p)\frac{N}{N+M} + \sum_{j=1}^{j=3} \frac{1}{\sum j + 1} p^j (1-p^j)\Big) \tag{5.17}$$

Note that this expression for G_2 is for the ARQ scheme considered, i.e., (1,2,3). The expression can be made general for any ARQ. For ease of demonstration, the specific scheme (1,2,3) has been worked out.

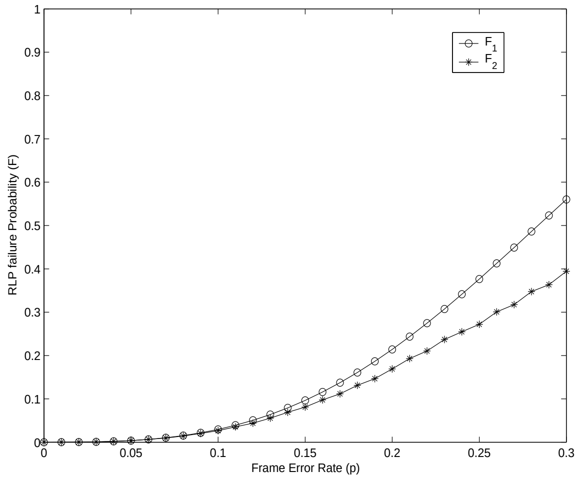

Figure 5.6: RLP failure probability vs. Frame error rate

5.5 Numerical Results For Differential ARQ

Just to focus on the ARQ performance, we assume that there are no redundancy bits added to any RLP frame, thus $M_1 = 0$. The frame error rate p is varied from 0 to 0.3. The plots for equations (5.10) and (5.11) are shown in Figure 5.6 which suggest how the RLP failure probability is lowered when differential ARQ is applied. However, from Figure 5.7 we do not see appreciable gain in the delay with the better ARQ. This is due to the fact that the ARQ scheme (1,2,3) successfully recovers more frames in a given time than the (1,1,1) scheme which results in an additional delay. This fact is further illustrated in Table 5.1 where we show the fraction of recovered frames after the initial transmission (Tx) and after every retransmission (Ret) for the two schemes. It is clear that from the second retransmission onwards the (1,2,3) scheme starts recovering more frames than the (1,1,1) scheme. and thereby these additional frames that have to be sent by the (1,2,3) scheme will contribute towards the delay. Last, from Figure 5.8 we observe that there is hardly any degradation in the goodput for scheme (1,2,3).

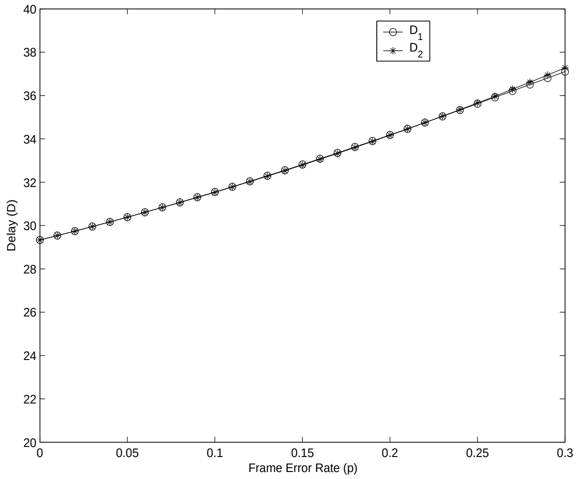

Figure 5.7: Delay vs. Frame error rate

This is because, the loss in goodput due to the transmission of duplicate frames in (1,2,3) scheme is compensated by the recovery of more frames. This is evident from the last row of Table 5.1 where the loss with scheme (1,2,3) is significantly lesser than that of scheme (1,1,1).

5.6 Applying Differential FEC+ARQ To The Crucial Frames

In sections 5.2 and 5.4, we have shown how differential RLP would perform if only FEC or ARQ was applied. In this section we apply both differential FEC and ARQ. Similar to the previous sections, we consider 2 cases as shown in figure 5.9.

In Case 1, the frames are subjected to an ARQ scheme of (1,1,1) and each frame is coded with M_1 bits. In Case 2, the non-crucial frames are treated as the frames

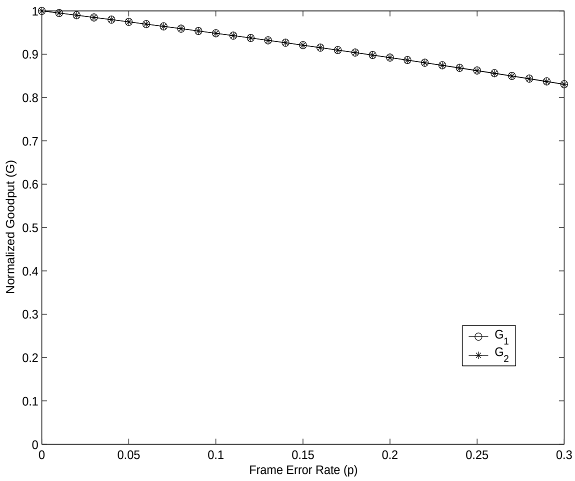

Figure 5.8: Goodput vs. Frame error rate

in Case 1 but the crucial frames are subjected to an ARQ scheme of (1,2,3) with M_2 redundancy bits per frame.

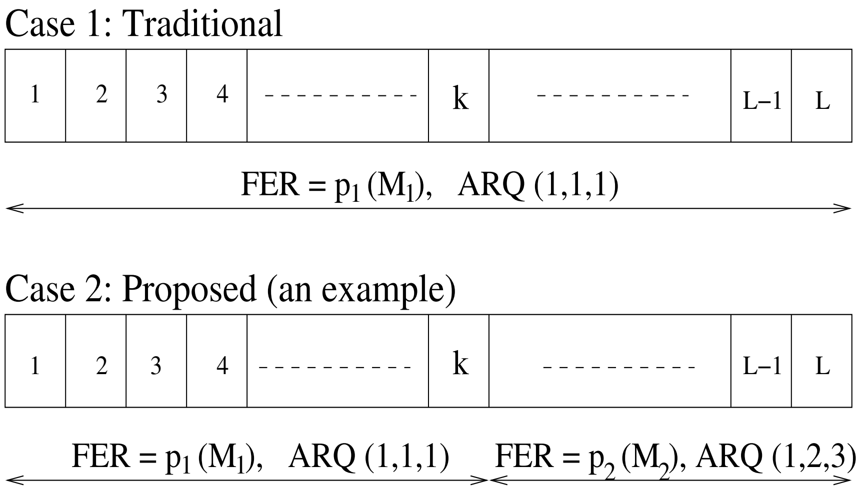

Figure 5.9: Different (FEC+ARQ) schemes

Table 5.1: ARQ Performance Comparison for $p = 0.2$

	ARQ (1,1,1)	ARQ (1,2,3)
Tx	0.8	0.8
Ret 1	0.16	0.16
Ret 2	0.032	0.0384
Ret 3	0.0064	0.0015872
Loss	0.0016	0.0000128

5.6.1 RLP Failure Probability

The RLP failure probability for Case 1 is obtained as

$$F_1 = 1 - (1 - p_1^3)^L \tag{5.18}$$

Similarly the RLP failure probability for Case 2 is obtained as

$$F_2 = 1 - \left((1 - p_1^3)^k \times (1 - p_2 6)^{L-k}\right) \tag{5.19}$$

5.6.2 Delay

The delay in Case 1 will be the same as that of D_1 in the differential ARQ case but p would be replaced by p_1 in the expression for D_1. Thus,

$$\begin{aligned} D_1 = (1 - p_1)\frac{T_{rtt}}{2} + p_1(1 - p_1)\frac{3T_{rtt}}{2} + \\ p_1^2(1 - p_1)\frac{5T_{rtt}}{2} + p_1^3(1 - p_1)\frac{7T_{rtt}}{2} + (L - 1)r \end{aligned} \tag{5.20}$$

Similarly for Case 2, the delay for the non-crucial frames, D_{nc}, would be the same as the equation for delay for the non-crucial frames in the ARQ case with

p replaced by p_1. Hence the delay for the non-crucial frames would be

$$D_{nc} = (1-p_1)\frac{T_{rtt}}{2} + p_1(1-p_1)\frac{3T_{rtt}}{2} + \\ p_1^2(1-p_1)\frac{5T_{rtt}}{2} + p_1^3(1-p_1)\frac{7T_{rtt}}{2} + (k-1)r \tag{5.21}$$

The delay for the crucial frames ($k+1$ through L) will now have the combined effect of both FEC and ARQ. The expected delay for any crucial frame would be $(1-p_2)\frac{T_{rtt}}{2} + p_2(1-p_2)\frac{3T_{rtt}}{2} + p_2^2(1-p_2^2)\frac{5T_{rtt}}{2} + p_2^4(1-p_2^3)\frac{7T_{rtt}}{2}$. The expected delay ($D_c$) from the crucial frames would be

$$D_c = (1-p_2)\frac{T_{rtt}}{2} + p_2(1-p_2)\frac{3T_{rtt}}{2} + \\ p_2^2(1-p_2^2)\frac{5T_{rtt}}{2} + p_2^4(1-p_2^3)\frac{7T_{rtt}}{2} + (L-1)r \tag{5.22}$$

The term $(L-1)r$ appears because the delay is calculated with respect to the first frame. So, the overall delay for Case 2 is again obtained by the greater of the two– D_{nc} and D_c. Thus,

$$D_2 = max(D_{nc}, D_c). \tag{5.23}$$

5.6.3 Goodput

The goodput due to the original transmission and three retransmissions in Case 1 would be

$$G_1 = \sum_{j=0}^{j=3} \frac{1}{j+1} p_1^j (1-p_1) \frac{N}{N+M_1} \tag{5.24}$$

Similarly for Case 2, where the non-crucial frames are subject to an ARQ scheme of (1,1,1) and redundancy bits of M_1 and the crucial frames with an ARQ scheme of (1,2,3) with redundancy bits M_2 the goodput is given as

$$G_2 = \frac{k}{L}G_1 + \frac{L-k}{L}\Big((1-p_2)\frac{N}{N+M_2} + \sum_{j=1}^{j=3} \frac{1}{\sum j+1} p_2^j (1-p_2^j)\Big) \tag{5.25}$$

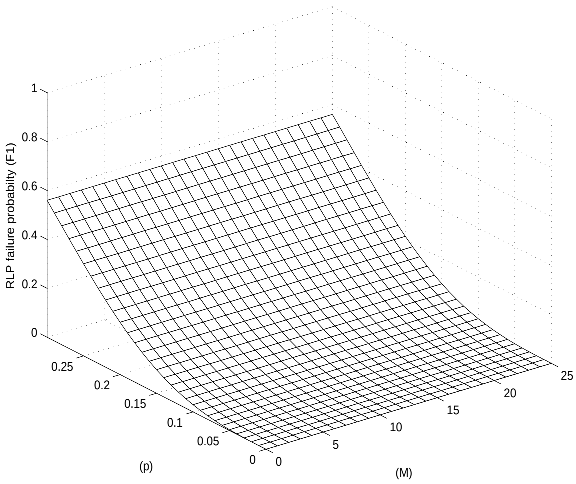

Figure 5.10: RLP failure probability vs. Number of redundancy bits and FER (Case 1)

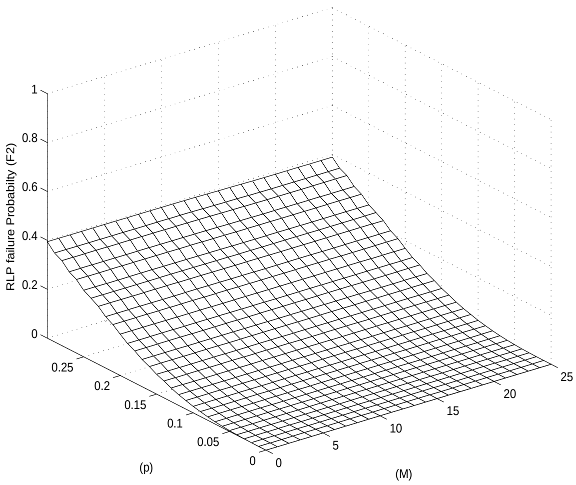

Figure 5.11: RLP failure probability vs. Number of redundancy bits and FER (Case 2)

5.7 Numerical Results For ARQ+FEC

To see the effect of differential FEC+ARQ, we maintain the same range for p (0 - 0.3) and M_2 (0 - 25) as in our previous results. The crucial frames have an ARQ scheme of (1,2,3). We have also assumed that the non crucial frames are

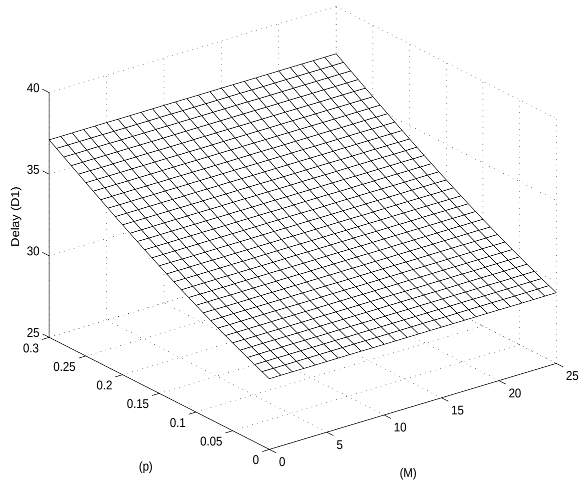

Figure 5.12: Delay vs. Number of redundancy bits and FER (Case 1)

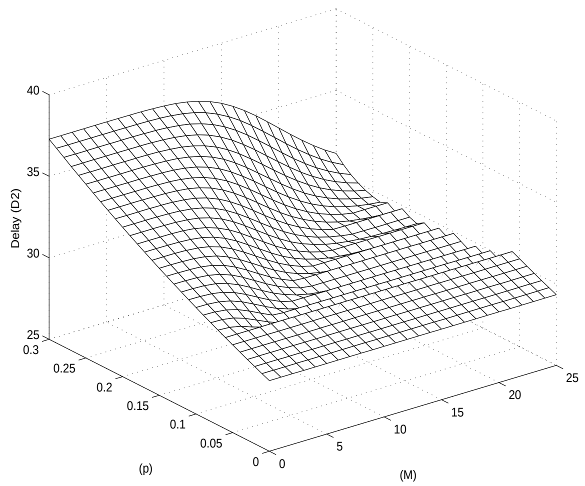

Figure 5.13: Delay vs. Number of redundancy bits and FER (Case 2)

not subjected to any FEC and so for them $M_1 = 0$ and the ARQ scheme (1,1,1). The RLP failure probabilities for the traditional RLP (Case 1) are plotted in Figure 5.10 and the proposed RLP (Case 2) in Figure 5.11. It can be observed how the RLP failure probability is lowered in case of Figure 5.11. We also observe from Figures 5.12 and 5.13 how the increase in the redundancy bits improves the

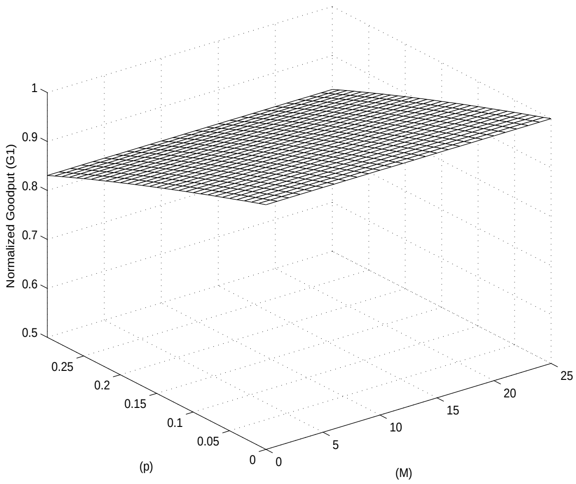

Figure 5.14: Goodput vs. Number of redundancy bits and FER (Case 1)

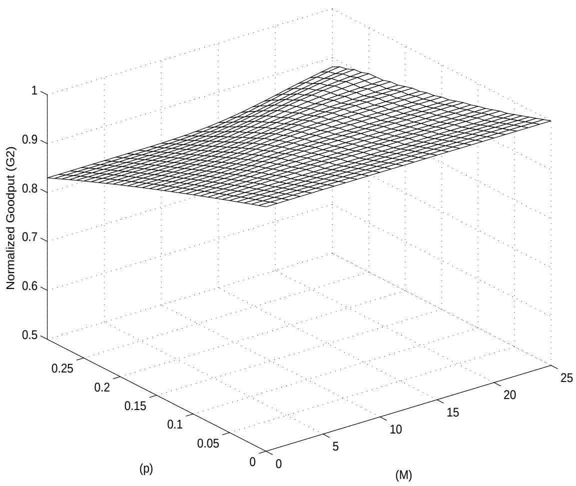

Figure 5.15: Goodput vs. Number of redundancy bits and FER (Case 2)

delay performance in Case 2. We can also see from the nature of the plot that for low values of frame error rate, with an increase in M_2 there is an abrupt improvement in delay. These are the sudden drops in the plot that appears as discontinuities and they are due to the improvement in delay with stronger FEC codes for smaller p's. The plots for goodput are shown in Figures 5.14 and 5.15.

Interestingly we find that initially the goodput decreases due to the increasing redundancy bits and also more number of retransmissions due to (1,2,3) scheme. However, when the frame error rate is high the goodput increases because with the increasing redundancy bits and the (1,2,3) scheme will actually recover more packets which then contributes towards increasing the goodput. Or, in other words, the differential RLP is more effective when the frame error rate is high.

It is to be noted that all the results discussed pertains to the assumed FEC and ARQ and thus the gain/loss in the performance are qualitative.

5.8 Effect On TCP Throughput

With the proposed RLP, let us now evaluate the improvement in the throughput at the TCP layer. It may be recalled from Section 5.1 that TCP segments are fragmented into multiple RLP frames. If we assume that the size of a TCP segment is T bytes and it is fragmented into equal sized RLP frames, then the number of RLP frames obtained from a TCP segment would be $L = \lceil \frac{T}{R} \rceil$, where R is the payload of each RLP frame. The actual size of the RLP frame would be R plus some header information. For a TCP segment to be reassembled successfully, all the L frames must be received correctly. If one or more RLP frames fail, the TCP segment is lost. Thus, the TCP segment loss probability, TCP_{loss}, is given by

$$TCP_{loss} = 1 - (1 - p)^L \qquad (5.26)$$

where, p is the frame loss probability at the physical layer. If however, we assume an underlying RLP (1, 2, 3) in operation, then the effective frame loss probability at the RLP layer is p^7. Hence, with the RLP layer, the TCP segment loss probability in Equation (5.26) changes to

$$TCP_{loss} = 1 - (1 - p^7)^L \tag{5.27}$$

We will assume that the TCP throughput is given by [43]

$$S_{TCP} = \frac{MSS}{RTT\sqrt{\frac{2bTCP_{loss}}{3}} + T_0 min(1, 3\sqrt{\frac{3bTCP_{loss}}{8}})TCP_{loss}(1 + 32TCP_{loss}^2)} \tag{5.28}$$

where, MSS is the maximum segment size, RTT is the round trip time for the TCP ACKs, T_0 is the TCP retransmission timer and b is a system constant. T_0 is evaluated as an exponentially moving average of the instantaneous RTTs.

We can now calculate the change in the TCP throughput when both differential ARQ and FEC are applied. For the sake of comparison, we will consider the two cases as discussed in Section 5.6. The TCP segment loss probability, TCP_{loss}, for both cases will be obtained from Equations (5.18) and (5.19) respectively. The round trip time for the TCP is the total delay in the wired network (between TCP end host and the base station) and the wireless network (between base station and the mobile terminal). We will consider that the delay in the wired network would remain the same and the only variation would be due to the two implementations of the RLP. Hence, we will consider Equations (5.20) and (5.23) to calculate the RTT. Figure 5.16 shows the improvement in the TCP throughput when the proposed RLP is applied. MSS is assumed to be 1500 bytes, $T_0 = 10 \times RTT$, and $b = 2$, as per the traces obtained in [43]. RLP frames were 50 bytes. The window size for TCP is assumed to grow without limit. The TCP throughput is plotted in the log scale, therefore the absolute improvement is much more. The improvement is more significant when the channel losses are high. The improvement in TCP throughput is due to two reasons. First, the fragmentation of the TCP segments into RLP frames prevents entire TCP segment to be retransmitted, if lost. Second, due to the differential treatment of the crucial frames, the RTT and TCP_{loss} are improved and hence increased TCP throughput.

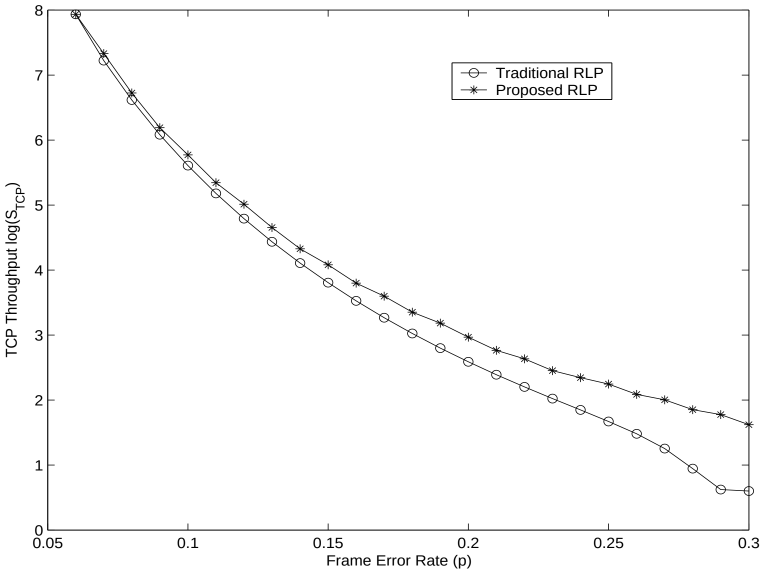

Figure 5.16: TCP Throughput

CHAPTER 6

CONCLUSIONS

One of the most challenging and interesting recent trends in computer networks is the integration of mobile communications. With the increasing importance of host mobility, and the popularity of TCP/IP on fixed networks, we are in need of a reliable mobile TCP/IP protocol to be used in wireless networks. In order that the mobile TCP protocol performs efficiently, a link layer protocol suitable for mobile networks must be used. The use of link layer protocols to provide an acceptable error performance over the wireless connection is now a standard industrial practice. While link layer protocols can efficiently provide reliability, transport layer protocols can be designed to efficiently deal with handoff and disconnections. This thesis demonstrates how the performance of radio link protocols can be improved if the RLP frames are treated differentially. The frames were categorized into crucial and non-crucial and their ratio as a function of segment size, round-trip time and frame error rate was obtained. Differential FEC and ARQ was then applied based on the relative position of the frames. We considered parallel transmission (as in HSDPA) of original and retransmit frames, and specific FEC and ARQ schemes to show the qualitative gain. Similar quantitative analysis can be done for generalized FEC and ARQ. The results clearly signify that if the performance of the differential RLP is known for various FEC and ARQ schemes under different channel conditions, then the RLP can choose the appropriate hybrid mechanism which will sustain the promised level of reliability expected from the applications to be supported.

APPENDIX A

GLOSSARY OF TERMS

- *3G (Third-Generation)*: The next generation of wireless technology that offers increased capacity and high-speed data applications up to 2 megabits. Integrates pico-micro and macrocellular technology and allows global roaming
- *3GPP*: 3rd Generation Partnership Project for W-CDMA (GSM)
- *3GPP2*: 3rd Generation Partnership Project for cdma2000
- *ARQ*: Automatic Repeat reQuest. A method of error correction where the receiver detects errors, and requests retransmission from the sender
- *Bit Error Rate (BER)*: A Measure of the rate at which error is introduced in the transmission of bits in a channel. An error is encountered when a 0 bit becomes a 1 or vice versa.
- *Bits per Second (BPS)*: A measure of how fast binary digits can be sent through a channel; the number of 0s and 1s that travel down the channel per second.
- *CDMA (Code Division Multiple Access)*: CDMA separates communications by code. Voice is broken into digitized bits, and groups of bits are tagged with a code. Each code is associated with a single call in the network. Groups of bits from one call are randomly transmitted along with those of other calls. Then they are reassembled in the correct order to complete the conversation.
- *CDMA 2000*: Trade name for CDMA air interface standards aimed at 3G requirements, including IS-2000. It operates in 1.25 MHz carriers at 1.2288 Mcps. There is some debate about whether the "CDMA" should be upper or lower case.

- *CRC*: Cyclic Redundancy Code (or check). Included in many digital protocols to check for errors in transmitted messages

- *Error Correction*: Digital technology's ability to verify the validity of the transmitted information and to automatically correct for errors caused by interference.

- *FCC*:US Federal Communications Commission

- *FEC*: Forward Error Correction

- *GSM*: GSM is the pan-European standard for digital cellular telephone service. It is also one of the technologies available in the Americas. GSM was designed for markets to provide the advantage of automatic, international roaming in multiple countries. The SIM (Subscriber Identification Module) card is a vital component in GSM operation. The user can store all relevant data for the phone on a removable plastic card. The card can be plugged into any GSM compatible phone and the phone is instantly personalized to the user.

- *Handoff*:The process of a MS changing from one frequency in one cell or sector to a different frequency in a neighboring cell or sector

- *HARQ*: Hybrid ARQ

- *HSDPA*: 3GPP High Speed Downlink Packet Access. Peak rates are planned to be 10-20 Mbps

- *IETF*: Internet Engineering Task Force. Standards setting body for the Internet

- *IMT-2000*: International Mobile Telecommunications for the year 2000

- *IMTS-2000*: The ITU 3G initiative. It does not define specific protocols, but just the performance goals for them, such as bandwidth. Specifications are being developed by 3GPP and 3GPP2.

- *IP Address*: (Internet Protocol Address) Location of a server assigned by your service provider. When loaded into a wireless device, such as the iDEN i1000plus or i500plus handset, the IP address allows you to use a mini-browser to access the Internet.

- *IS-2000*: cdma2000 air interface standard. A successor to TIA/EIA-95-B

- *IS-41*: Wireless intersystems operation standard. Now called TIA/EIA-41

- *IS-95*: cdmaOne CDMA air interface standard

- *PAN*: Personal Area Network. A network that connects personal devices, such as computer, keyboard, mouse, phone and monitor. Also known as Piconet

- *Protocol*: A specification of the messages used to communicate over one or more Interfaces

- *QoS*: Quality of Service. A list of measurable attributes such as bandwidth, delay and jitter that should be met for a specific communications service

- *RFC*: IETF Request for Comments. Internet standard (well, not officially, but in practice many internet 'standards' are still just RFCs)

- *RFP*: Request for Proposal

- *RLP*: Radio Link Protocol

- *RNC*: Radio Network Controller

- *TCP*: Transmission Control Protocol. A protocol that provides for reliable delivery of messages over the internet.
- *TIA*: Telecommunications Industry Association. A trade association that, among other things, defines standards for cellular and PCS, specifically AMPS, NAMPS, CDMA and TDMA
- *TIA/EIA*: A prefix for a standard produced by the TIA in association with the EIA
- *TIA/EIA-136*: ANSI version of the TDMA air interface standard. Replaces IS-136
- *TIA/EIA-95*: CDMA air interface standard
- *UMTS*: Universal Mobile Telecommunications System (a 3G initiative). See www.umts-forum.org. It operates in 5 MHz channels at 3.84 Mcps with 200 kHz between channels.
- *WCDMA*: Physical layer of the FDD mode of operation of UTRA. A 'European' version of CDMA and the 3G evolutionary step planned for GSM. Operates in pairs of 5 MHz channels at 3.84 Mcps

APPENDIX B

CONGESTION CONTROL IN TCP

B.1 Slow Start

TCP operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgments are returned by the other end. Slow start adds another window to the sender's TCP: the congestion window, called *cwnd*. When a new connection is established with a host on another network, the congestion window is initialized to one segment (i.e., the segment size announced by the other end, or the default, typically 536 or 512). Each time an ACK is received, the congestion window is increased by one segment. The sender can transmit up to the minimum of the congestion window and the advertised window. The congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver. The former is based on the sender's assessment of perceived network congestion; the latter is related to the amount of available buffer space at the receiver for this connection.

The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent. When each of those two segments is acknowledged, the congestion window is increased to four. This provides an exponential growth, although it is not exactly exponential because the receiver may delay its ACKs, typically sending one ACK for every two segments that it receives. At some point the capacity of the internet can be reached, and an intermediate router will start discarding packets. This tells the sender that its congestion window has gotten too large. Early implementations performed slow start only if the other end was on a different network. Current implementations always perform slow start.

B.2 Congestion Avoidance

Congestion can occur when data arrives on a big pipe (a fast LAN) and gets sent out a smaller pipe (a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets. The assumption of the algorithm is that packet loss caused by damage is very small therefore the loss of a packet signals congestion somewhere in the network between the source and destination. There are two indications of packet loss: a timeout occurring and the receipt of duplicate ACKs.

Congestion avoidance and slow start are independent algorithms with different objectives. But when congestion occurs TCP must slow down its transmission rate of packets into the network, and then invoke slow start to get things going again. In practice they are implemented together. Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window, *cwnd*, and a slow start threshold size, *ssthresh*. The combined algorithm operates as follows:

1. Initialization for a given connection sets *cwnd* to one segment and *ssthresh* to 65535 bytes.

2. The TCP output routine never sends more than the minimum of *cwnd* and the receiver's advertised window.

3. When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size (the minimum of *cwnd* and the receiver's advertised window, but at least two segments) is saved in

ssthresh. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment (i.e., slow start).

4. When new data is acknowledged by the other end, increase *cwnd*, but the way it increases depends on whether TCP is performing slow start or congestion avoidance.

If *cwnd* is less than or equal to *ssthresh*, TCP is in slow start; otherwise TCP is performing congestion avoidance. Slow start continues until TCP is halfway to where it was when congestion occurred (since it recorded half of the window size that caused the problem in step 2), and then congestion avoidance takes over.

B.3 Fast Retransmit

TCP may generate an immediate acknowledgment (a duplicate ACK) when an out- of-order segment is received. This duplicate ACK should not be delayed. The purpose of this duplicate ACK is to let the other end know that a segment was received out of order, and to tell it what sequence number is expected. Since TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for a small number of duplicate ACKs to be received. It is assumed that if there is just a reordering of the segments, there will be only one or two duplicate ACKs before the reordered segment is processed, which will then generate a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then performs a retransmission of what appears to be the missing segment, without waiting for a retransmission timer to expire.

B.4 Fast Recovery

After fast retransmit sends what appears to be the missing segment, congestion avoidance, but not slow start is performed. This is the fast recovery algorithm. It is an improvement that allows high throughput under moderate congestion, especially for large windows. The reason for not performing slow start in this case is that the receipt of the duplicate ACKs tells TCP more than just a packet has been lost. Since the receiver can only generate the duplicate ACK when another segment is received, that segment has left the network and is in the receiver's buffer. That is, there is still data flowing between the two ends, and TCP does not want to reduce the flow abruptly by going into slow start. The fast retransmit and fast recovery algorithms are usually implemented together as follows.

1. When the third duplicate ACK in a row is received, set *ssthresh* to one-half the current congestion window, *cwnd*, but no less than two segments. Retransmit the missing segment. Set *cwnd* to *ssthresh* plus 3 times the segment size. This inflates the congestion window by the number of segments that have left the network and which the other end has cached .

2. Each time another duplicate ACK arrives, increment cwnd by the segment size. This inflates the congestion window for the additional segment that has left the network. Transmit a packet, if allowed by the new value of *cwnd*.

3. When the next ACK arrives that acknowledges new data, set cwnd to ssthresh (the value set in step 1). This ACK should be the acknowledgment of the retransmission from step 1, one round-trip time after the retransmission. Additionally, this ACK should acknowledge all the intermediate

segments sent between the lost packet and the receipt of the first duplicate ACK. This step is congestion avoidance, since TCP is down to one-half the rate it was at when the packet was lost.

LIST OF REFERENCES

[1] 3G Partnership Project, Release 99.

[2] http://www.3gpp2.org

[3] http://www.cdg.org (CDMA Development Group)

[4] 3GPP TR25.848 V0.5.0,Physical Layer Aspects of UTRA High Speed Downlink Packet Access(HSDPA), ANNEX B, TSGR1#18(01)186, Jan 18,2001.

[5] TIA/EIA/IS-707-A.2, "Data Service Options for Spread Spectrum Systems: Radio Link Protocol", March 1999.

[6] M. Allman, D. Glover, and L. Sanchez, 1999, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", RFC 2488.

[7] E. Ayanoglu, S. Paul, T. F. Laporta, K. Sabnani and R. Gitlin, 1995, "AIRMAIL: A linklayer protocol for wireless networks", *Wireless Networks 1*, pp. 4760.

[8] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", *Proc. 15th Internationals Conference on Distributed Computing Systems*, May 1995, pp. 136-143.

[9] B. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan, 1997, "Improving performance of TCP over wireless networks", *17th International Conference on Distributed Computing Systems*, pp. 365373.

[10] H. Balakrishnan, V. Padmanabhan, and R. Katz, 1999, "Effects of asymmetry on TCP performance. Mobile Networks and Applications", *4,3*, pp. 219241

[11] G. Bao, "Performance Evaluation of TCP/RLP Protocol Stack over CDMA Wireless Link", *Wireless Networks*, No. 2, 1996, pp. 229-237.

[12] K. Brown, and S. Singh, 1997, "M-TCP: TCP for mobile cellular networks", *Computer Communication Review*, July, pp. 1943.

[13] R. Bruyeron, B. Hemon, and L. Zhang, 1998. “Experimentations with TCP selective acknowledgment ” *ACM SIGCOMM Computer Communication*, Review 28, 2, pp. 5477.

[14] R. Caceres and L. Iftode, “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments”, *IEEE Journal on Selected Areas in Communications*, Jun 1995, Vol. 13, pp. 850-857.

[15] M.C. Chan, R. Ramjee, “TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation”, *ACM MobiCom*, 2002, pp. 71-82.

[16] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, 2001, “A feedback-based scheme for improving TCP performance in ad-hoc wireless networks ”, *IEEE Personal Communications* 8, 1, pp. 3439.

[17] M. Chatterjee, S.K. Das and G.D. Mandyam, “Two-layer Retransmission Reliability for CDMA2000 1X-EV”, *Proceedings of IEEE Vehicular Technology Conference*, 2002-Spring, Vol. 1, pp. 75-79.

[18] A. Chockalingam and G. Bao, “Performance of TCP/RLP protocol stack on correlated fading DS-CDMA wireless links”, *IEEE Transactions on Vehicular Technology*, Volume 49, Issue 1, Jan 2000, pp. 28-33.

[19] D.E. Comer, *Internetworking with TCP/IP*, Vol. 1, Prentice-Hall, 1991.

[20] H. ElAarag, 2002, ”Improving TCP Performance over Mobile Networks”, *ACM Computing Surveys*, Vol. 34, No. 3, September 2002, pp. 357374.

[21] H. ElAarag, “Transport control protocols for the success of the Internet ”, *Ph.D. dissertation, University of Central Florida.*

[22] H. ElAarag and M. Bassiouni, “Performance evaluation of TCP connections in ideal and non-ideal network environments”, *Computer Communications Journal*, 24, 18, pp. 1769 1779.

[23] T. Faber, J. Touch, and W. Yue, 1999. “The TIMEWAIT state in TCP and its effect on busy servers.” *Proceedings of INFOCOM*, pp. 15731583.

[24] S. Floyd, J. Mahdavi ,M. Mathis, and M. Podolsky, 2000. “An extension to the selective acknowledgement (SACK) option for TCP”, *RFC 2883.*

[25] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, 2000, “Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments”, *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, IEEE3*, pp. 15371545.

[26] J.M. Harris and M. Airy, “Analytical model for radio link protocol for IS-95 CDMA systems”, *Proceedings IEEE 51st Vehicular Technology Conference (VTC)*, 2000-Spring, Vol. 3, pp. 2434-2438.

[27] G. Hasegawa, M. Murata, and M. Miyahara, 1999. “Fairness and stability of congestion control mechanisms of TCP”,*Proceedings of IEEE INFOCOM*, pp. 13291336.

[28] J. Heidemann, 1997. “Performance interactions between P-HTTP and TCP implementations”, *ACM SIGCOMM Computer Communication Review*, 27, 2, pp. 6473.

[29] J. C. Hoe, 1996. “Improving the start-up behavior of a congestion control scheme for TCP”, *ACM SIGCOMM*, pp. 270280.

[30] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, F. and Khafisov, F. 2001. “TCP over 2.5G and 3G wireless networks”,*IETF, Internet draft*, Available at: http://www.ietf.org/internetdrafts/ draft-ietf-pilc-2.5g3g-09.txt

[31] V. Jacobson, 1988, “Congestion avoidance and control”, *SIGCOMM Symposium on Communications Architectures and Protocols*, pp. 314329. ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z for an updated version.

[32] A.S. Joshi, Umesh M.N., A. Kumar, T. Mukhopadhyay, K. Natesh, S. Sen, and A. Arunachalam, “Performance evaluation of TCP over radio link protocol in TIA/EIA/IS-99 environment”, *IEEE International Conference on Personal Wireless Communication*, 1999, pp. 216-220.

[33] P. Karn, and C. Partridge, Nov. 1991, “Improving round-trip time estimates in reliable transport protocols”, *ACM Transactions on Computer Systems* 9, 4, pp. 364373.

[34] F. Khan, S. Kumar, K. Medepalli, and S. Nanda, “TCP Performance over cdma2000 RLP”, *Proceedings IEEE 51st Vehicular Technology Conference (VTC)*, 2000-Spring, Vol. 1, pp. 41-45.

[35] S.Lin and D.J Costello, Jr., “Error Control Coding: Fundamentals and Applications”, *Prentice Hall, 1983*

[36] H. Lin and S.K. Das, “Performance Study of Link Layer and MAC Layer Protocols in supporting TCP in 3G CDMA Systems,” To appear in IEEE Transaction on Mobile Computing.

[37] R. Ludwig, and B. Rathonyi, 1999, “Link layer enhancements for TCP/IP over GSM ”, *Proceedings of IEEE INFOCOM*, pp. 415422.

[38] Henrik Lundqvist and Gunnar Karlsson,”TCP with End-to-End Forward Error Correction”, *Radio Vetenskap och Kommunication*, RVK’02,Stockholm, June2002.

[39] E. Malkamaki, D. Mathew,and S. Hamalainen, “Performance of hybrid ARQ techniques for WCDMA high data rates” *IEEE VTC* 2001 Spring, Volume: 4, pp. 2720-2724.

[40] M. Mathis, J. Semke, and J. Mahdavi, July 1997. “The macroscopic behavior of the TCP congestion avoidance algorithm”, *ACM SIGCOMM Computer Communication Review* 27, 3.

[41] J. MO, R. LA ,V. ANANTHARAM , and J. WALRAND 1999. “Analysis and comparison of TCP Reno and Vegas ”, *Proceedings of IEEE INFOCOM*, pp. 1556 1563.

[42] S. Nanda, R. Ejzak, and B. Doshi, 1994, “A retransmission scheme for circuit-mode data on wireless links”, *IEEE Journal on Selected Areas in Communications*, 12, 8, pp. 13381352.

[43] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, *Proceedings of ACM SIGCOMM*, 1998, pp. 303-314.

[44] V. Paxson, August 1994. “Empirically derived analytic models of wide-area TCP connections”, *IEEE/ACM Transactions on Networking*, 2, 4, pp. 316336.

[45] C. Pazos, J. Agrelo and M. Gerla,1999. “Using back-pressure to improve TCP performance with many flows”, *Proceedings of IEEE INFOCOM*, pp. 431438.

[46] I. Rhee, N. Balaguruf, And G. Rouskasf,1999. ”MTCP: Scalable TCP-like congestion control for reliable multicast”, *Proceedings of IEEE INFOCOM*, pp. 12651273.

[47] N. Samaraweera and G. Fairhurst, April 1998, “Reinforcement of TCP error recovery for wireless communication”, *ACM SIGCOMM Computer Communication Review*, 28, 2, pp. 3038.

[48] S. K. Sen, J. Jawanda, K. Basu, N.K. Kakani, and S.K. Das, "TCP Source Activity and Its Impact on Call Admission Control in CDMA Voice/Data Network", *Proceedings of Fourth ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom) 1998, pp. 276-283.

[49] P. Sinha , N. Venkitaraman, R. Sivakumar and V. Bhargavan, 1999 "WTCP: A reliable transport protocol for wireless wide-area networks", *Proceedings of ACM Mobicom99*, Seattle, WA, pp. 231241.

[50] W. R. Stevens, 1994. "TCP/IP Illustrated", vol. 1, *Addison-Wesley.*

[51] Umesh M.N., A.S. Joshi, A. Kumar, T. Mukhopadhyay, "A NAK based hybrid type II ARQ scheme for cdmaOne/cdma2000 systems", *IEEE Vehicular Technology Conference* (VTC), Fall 1999, Volume 5, pp. 2596-2600.

[52] K. Wang and S. Tripathi, 1998, "Mobile-end transport protocol: an alternative to TCP/IP over wireless links", *Proceedings of IEEE INFOCOM*, 3, pp. 10461053.

[53] G. Xylomenos, and G. Polzos, 1999, "TCP and UDP performance over a wireless LAN ", *Proceedings of IEEE INFOCOM*, pp. 439446.

[54] X. Yang, 1999. "A model for window based flow control in packet-switched networks ", *Proceedings of IEEE INFOCOM*, pp. 423430.

[55] R. Yavatkar, and N. Bhagawat, 1995, "Improving end-to-end performance of TCP over mobile internetworks", *Proceedings of theWorkshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, pp. 146152.