PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI – University of Trento

SERVICE-AWARE PERFORMANCE OPTIMIZATION OF WIRELESS ACCESS NETWORKS

Nadhir Ben Halima

Advisor:
Prof. Fabrizio Granelli
University of Trento

Co-advisor:
Prof. Michael Devetsikiotis
North Carolina State University

December 2009

Abstract

Service-Aware Performance Optimization of Wireless Access Networks

By
Nadhir Ben Halima

Department of Information and Communication Technology
University of Trento, Italy

Internet was originally designed to offer best-effort data transport over a wired network with end machines using a layered network protocol stack to provide mainly reliability and quality of service for end user applications. However, the excess of wireless end devices and the demand for sophisticated mobile multimedia applications forces the networking research community to think about new design methodologies. In fact, this kind of applications is characterized not only by a large amount of required data-rate, but also by a significant variability of the data-rate over time due to the dynamics of scenes, when state of the art of video encoding techniques are considered and are especially challenging due to the time varying transmission characteristics of the wireless channel and the dynamic quality of service (QoS) requirements of the application (e.g., prioritized delivery of important units, variable bit rate and variable tolerance vs. bit or packet errors).

One of the focused issues in the improvement of multimedia transmission quality is to combine the characteristics of the video applications and the wireless networks. Traditional approaches, in which the characteristics of the video application and wireless networks are isolated, would induce the resources not being optimized. Cross-layer design also known as Cross-layering is a new paradigm in network design that not only takes the dependencies and interactions among the layers of the Open System Interface (OSI) structure into account, but also attains a global optimization of the layer-specific parameters. However, most existing cross-layer designs for Quality of Service (QoS) provisioning in multimedia communications are mainly either aiming at improving throughput of the network or reducing power consumption, yet regardless of the end-to-end qualities of multimedia transmission. Therefore, the application-driven cross-layer design over various multimedia communication systems is needed to be extensively investigated.

Following the extensive study of performance bounds and limitations of the sate-of-the art in this research area, we argue that performance improvement of multimedia applications over wireless access networks can be achieved through considering the application-specific requirements also called service- or context-awareness. Indeed, we designed two cross-layer design schemes called CORREC and SARC for Wi-Fi and 3G networks respectively. We show that further performance improvement can be achieved by

tuning ARQ and HARQ strength respectively based on the application requirements and protocol stack operation on the mobile terminal.

In the other hand, the Transmission Control Protocol (TCP) which accounts for over 95% of Internet traffic shows poor performance in wireless domain. We propose a novel approach aiming at TCP performance improvement in WLAN networks. It consists in proposing a joint optimization of ARQ schemes operating at the transport and link layers using a cross-layer approach called ARQ proxy for Wi-Fi networks.

To my mom and dad: Naziha and Achour,
To my brothers,

I dedicate this thesis.

Contents:

Abstract ---------- 2
List of Figures: ---------- 7
Acknowledgements ---------- 8
1. Introduction and Preview ---------- 9
1.1 Motivation ---------- 9
1.2. Challenges ---------- 10
1.3. Contributions and Thesis Structure ---------- 11
2. State Of the Art ---------- 13
2.1. Introduction ---------- 13
2.2 Performance Bounds and Limitations ---------- 13
2.2.1 Long Round Trip Time and Limited Bandwidth ---------- 13
2.2.2 High Loss Rate ---------- 13
2.2.3 Mobility ---------- 14
2.2.4 Asymmetric Links Bandwidth ---------- 14
2.3 TCP Performance Enhancements ---------- 14
2.3.1 Link-Layer Solutions ---------- 15
2.3.2 Split Solutions ---------- 17
2.3.3 End-to-End Solutions ---------- 19
2.4. Cross-Layer Design ---------- 31
2.4.1 ILC-TCP (Interlayer Collaboration Protocol) ---------- 31
2.4.2 ATCP. ---------- 31
2.5 Application-aware solutions ---------- 31
2.6 Conclusions ---------- 32
3. Cross-Layer Error Recovery Optimization in Wi-Fi Networks ---------- 33
3.1 Introduction ---------- 33
3.2 ARQ Proxy ---------- 34
3.3 Performance Evaluation ---------- 38
3.4 Conclusions ---------- 41
4. Context-aware Receiver-driven Retransmission Control in Wireless Local Area Networks ---------- 42
4.1 Introduction ---------- 42
4.2 Proposed Approach ---------- 43
Packet Importance Metric ---------- 44

File Transfer Applications ---44
Multimedia Applications--45
4.3 Performance Evaluation --46
Scenario Definition --46
Results: TCP flows --47
Results: Multimedia traffic --48
4.4 Conclusions---48
5. Service-Aware Retransmission Control in Cellular Networks ---------------------------52
5.1 Introduction ---52
5.2 Proposed Approach --53
Packet Importance Metric---54
Packet Importance Metric in Video Streams--55
Packet Importance Metric in VoIP Flows ---56
Packet Importance Metric in File Transfer--57
5.3 Performance Evaluation --58
Simulation Scenario ---58
Video Transfer Performance ---58
VoIP Transfer Performance --60
File Transfer Peformance ---61
5.4 Conclusions and Future Work --62
6. Conclusions and Future Work---64

List of Figures:

Fig. 1. Cross-layer Design: interaction between the different layers of the protocol stack. 11
Fig. 2.TCP data packet delivery over IEEE 802.11 wireless link. ----------------------------34
Fig. 3.ARQ proxy and ARQ client functionality. ---35
Fig. 4.Packet identification techniques: a) frame sequence numbers and b) hash values.--36
Fig. 5 ARQ Client position in MN's protocol stack.---36
Fig. 6.IEEE 802.11 ACK frame with extension capable to carry TCP ACK index. --------37
Fig. 7.The simulation scenario --38
Fig. 8.ARQ proxy TCP throughput comparison. --39
Fig. 9.ARQ proxy Round Trip Time (RTT) reduction.--------------------------------------40
Fig. 10.TCP throughput against wireless link errors. --------------------------------------40
Fig. 11.IEEE 802.11 ACK frame. ---43
Fig. 12. Packet importance metric for TCP-based data flow.--------------------------------44
Fig. 13. Prediction encoding of MPEG-4, GOP (N=9, M=3) -------------------------------45
Fig. 14.The simulation scenario. ---46
Fig. 15.Structure of the Integrated Tool Environment---47
Fig. 16.Performance of CORREC in terms of TCP throughput against packet error rate. -49
Fig. 17.PSNR against wireless link errors in case of static settings and with CORREC enabled. --49
Fig. 18.Number of I-, P- nad B- packets correctly received against PER on the wireless link.--50
Fig. 19.Level of improvement in terms of the ratio between packets sent and packets received of the same type (I, P, and B).---50
Fig. 20.Visual comparison of achieved performance using the legacy scheme (upper) and the proposed scheme (CORREC, lower).--51
Fig. 21.Architectural principles of SARC in a 3G cellular network. Blocks and links highlighted in "blue" underline the modules and signaling links employed by SARC approach. ---54
Fig. 22. Quality of the received video flow for different frames lost. ------------------------55
Fig. 23. Scheme of a VoIP application---57
Fig. 24.Simulation scenario used for ns-2 experiments.---58
Fig. 25.Quality of "Foreman" video clip for different error rates. ----------------------------59
Fig. 26.Percentage of improvement in correctly delivered I-, P-, and B-packets against PER.--60
Fig. 27.G.711 Voice MOS against Packet Error Rate. --61
Fig. 28GSM AMR MOS against Packet Error Rate. ---61
Fig. 29.Average TCP Throughput as a function of Packet Error Rate on the wireless channel. ---62
Fig. 30.Average TCP Throughput against Packet Error Rate in presence of video flows. -63

Acknowledgements

In completion of this work I am thankful to many people, as I firmly believe, without their help, guidance and most sincere cooperation; this accomplishment would not have been possible.

First of all, I would like to wholeheartedly thank my advisor Prof. Fabrizio Granelli, for his invaluable support during my study, for his strong guidance, availability, and trust throughout my Ph.D. Without his guidance and patience my PhD research would not be accomplished. His help and support will be extremely appreciated throughout my career.

I would like also to thank all my professors at the Department of Information and Communication Technology at the University of Trento. I have learned not only scientific techniques from them through doctoral courses, but also more importantly, the method of scientific thinking, how to identify a research problem, how to form and carry out a research plan and how to get results.

I wish to express my gratitude to my co-advisor Prof. Michael Devetsikiotis and his research group for hosting and supervising me in the Network Performance Research Group at the Department of Electrical and Computer Engineering at North Carolina State University. A Network Performance Research group member, Mr. Michalis Kallitsis is working towards his PhD now. I wish him best luck in his research.

My cordial thanks also extend to Prof. Piero Castoldi and Prof. Luigi Atzori for being on my dissertation defense committee.

I would like to acknowledge the members of the research group for the excellent atmosphere they created. I am especially thankful to Dr. Dzmitry Kliazovich, research fellow in our group, for collaboration and constant discussion on topics related to my thesis.

Finally, I would like to express my deep gratitude to my mother Naziha Thabet and my father Achour Ben Halima for their patience, continuous support and encouragement throughout this thesis and for bringing me up to what I am now.

Nadhir Ben Halima

Before beginning a Hunt, it is wise to ask someone what you are looking for before you begin looking for it

-Winnie the pooh

CHAPTER 1

1. Introduction and Preview

1.1 Motivation

In the last 30 years, the IT industries have witnessed two big waves of revolution, one being the invention of the Internet, and the other the wide applications of wireless technologies. Very soon everybody will be given the privilege that all voice telephone calls will be free of charge, thanks to the wide accessibility of the Internet throughout the world. The Internet operates on all-IP based network architecture, and thus the network level design and performance ensuring mechanism play a critical role in all Internet related applications.

On the other hand, the revolution of wireless technologies fuels the advancements in modern telecommunication systems through its mobile extension of wired networks, such as the Internet. Mobility is one of the most important characteristics of modern society. Everything and everyone are in motion. Therefore, the information dispatching facilities should also be made available while people are on the move. The explosive increase in mobile cellular telephone services around the world has reflected the great demand for mobile communications. The availability of mobile cellular communications has exerted a strong influence on the lifestyle, the business models, as well as on the sense of value, distance, and time.

The Internet in the absence of wireless technologies' support cannot offer the end users such convenience and readiness; while the wireless systems without the backup of the Internet infrastructure will limit its diversity in services and content. The combination of the Internet and wireless technologies will provide us access to information services at any time, in any place, and to anyone.

The combination of the Internet and wireless technologies has also created many challenging issues, such as the joint optimization of software and hardware implementations, all-IP wireless platforms, intelligent radios, and so on which has brought a fundamental change to the design of wireless networks. The demands on voice-centric services have been quickly overtaken by data-centric applications. Indeed, the circuit-switched end-to-end connection communication system and network design philosophy has been replaced by all-IP packet-switched connectionless architecture, and applications like video streaming, video conferencing and interactive networked 3D games are attracting an increasing number of mobile users. Supporting this kind of applications will be a major challenge for beyond third-generation (B3G) wireless networks. Mobile users are expecting high quality and transparent service independent of the network access technology.

The great demands on the capacity and quality offered over wireless communication links have pushed the networking research community to innovate new design methodologies and concepts for wireless systems and networks. In fact, the layered approach has been widely used in the past, but it is no longer adequate to meet the challenges of next-generation mobile systems. Mobile multimedia communication is especially challenging due to the time varying transmission characteristics of the wireless channel and the dynamic quality of service (QoS) requirements of the application (e.g., prioritized delivery of important units, variable bit rate and variable tolerance vs. bit or packet errors).

1.2. Challenges

Setting the control modes and tuning the parameters of the protocols at design time and for the worst case scenarios lead to poor performance and inefficient utilization of resources. Instead, a network observing the behavior of the application and of the physical channel and dynamically adapting to the changes is able to maintain optimal allocation of resources and performance improvement of applications over wireless access networks can be achieved through considering the application-specific requirements or in other words where application or service-driven scheme are used. This requires timely exchange of parameters across layers and periodic reconfiguration of modes and parameters of the protocol layers during network operation.

In the other hand, the problems that arise in the usage of the Transmission Control Protocol (TCP) [1] over wireless networks are due to their low reliability, as well as time-variant characteristics such as fading, shadowing, node mobility, hand-offs, limited available bandwidth and large Round Trip Times (RTTs). TCP protocol, originally designed for wired networks which are characterized by stable links with packet losses mainly caused by congestion, performs poorly in wireless environments.

That’s why; the traditional layered architecture of wireless communication systems or networks has faced a great challenge from cross-layer optimized design. The previously clearly defined boundaries between the seven Open System Interface (OSI) layers are diminishing. Indeed, cross-layer design utilizes interactions between protocols to increase network performance and throughput [2]. Cross-layer design violates the traditional layered approach of network architecture design by allowing direct communication or sharing of internal information between nonadjacent or adjacent layers (see Fig.1). Examples of violation of a layered architecture include creating new interfaces between layers, redefining the layer boundaries, designing protocol at a layer based on the details of how another layer is designed, joint tuning of parameters across layers, and so on. A common misconception about cross-layering is that it consists of designing networks without layers. Cross-layering should not be viewed as an alternative to the layered design approach, but rather as a complement.

Layering and cross-layering are tools that should be used together to design highly adaptive wireless networks.

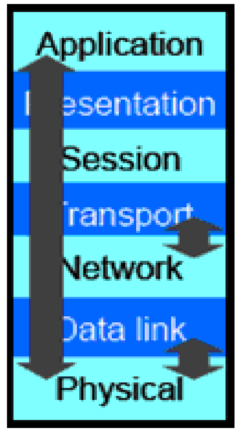


Fig. 1. Cross-layer Design: interaction between the different layers of the protocol stack.

1.3. Contributions and Thesis Structure

In this thesis, a number of problems are introduced and appropriate solutions are presented using cross-layering approach. The performed evaluation and comparison with existing solutions limited in optimization to a particular layer show tremendous advantages enabled by cross-layer design. The major contribution of this thesis can be summarized in three parts:

- Proxy ARQ for Wi-Fi: a novel approach for cross-layer error control optimization in Wi-Fi networks. The focus is on the reduction of the overhead deriving from the duplicate ARQ strategies employed at the link and transport layers. The proposed solution, called ARQ proxy, substitutes the transmission of a transport layer acknowledgement with a short request sent at the link layer. Specifically, TCP ACKs are generated based on in-transit traffic analysis and stored at the Access Point. Such TCP ACKs are released towards TCP sender upon a request from the mobile node, encapsulated into link layer acknowledgement frame. TCP ACK identification is computed at the Access Point as well as at the mobile node in a distributed way. ARQ proxy improves TCP throughput in the range of 25-100% depending on the TCP/IP datagram size used by the connection. Additional performance improvement is obtained due to RTT reduction and higher tolerance to wireless link errors.

- CORREC for WI-FI Networks: In this novel scheme, we propose to enable per-packet differentiation of link layer ARQ protection (in terms of no. of retransmissions) driven by requirements of the end applications as well as of communication protocols implemented on the mobile terminal. Experimental results demonstrate the potential benefits deriving from the proposed strategy, both on TCP data flows and MPEG-4 video streams.

- SARC for 3G Networks: It is an application-aware cross-layer approach between application/transport layers on the mobile terminal and link layer at the wireless base station to enable dynamic control on the strength of per-packet error protection for multimedia and data transfers. Specifically, in the context of cellular networks, the proposed scheme allows to control the desired level of Hybrid ARQ (HARQ) protection by using an in-band control feedback channel. Such protection is dynamically adapted on a per-packet basis and depends on the perceptual importance of different packets as well as on the reception history of the flow.

The rest of the thesis is organized as follows: Chapter 2 examines performance issues that arise in wireless networks defining performance bounds and limitations. Then, it presents an overview and comparison of the existing optimization solutions. In Chapter 3, we describe a cross-layer error recovery optimization called proxy ARQ for Wi-Fi networks. Chapter 4 introduces a Context-aware Receiver-driven Retransmission Control in Wireless Local Area Networks called CORREC for WI-FI. Chapter 5 instead, introduces SARC for 3G networks scheme. Finally, in Chapter 6, we present a summary of the research work drawing conclusions and outlining directions for future research in the field.

Study the past if you would define the future.

- Confucius

CHAPTER 2

2. State Of the Art

2.1. Introduction

Wireless networks are becoming increasingly popular, especially for the provisioning of mobile access to wired network services. As a consequence, efforts have been devoted to the provisioning of reliable data delivery for a wide variety of applications over different wireless infrastructures. This chapter aims at providing a comprehensive analysis of the performance limitations and potential enhancements to wireless access networks. Proposals to overcome such limitations are presented.

The structure of this chapter is as following: Section 2.2 surveys the performance issues related to throughput and delay in wireless networks. Section 2.3 introduces existing proposals to overcome those problems. Sections 2.4 and 2.5 present cross-layer solutions and mainly application/transport layer driven optimizations for wireless access networks. Finally, Section 2.6 draws some conclusions.

2.2 Performance Bounds and Limitations

Before describing the solutions proposed to enhance the application/transport performance over wireless networks, this section presents a brief overview of the basic factors that reduce the application/TCP performance in wireless networks.

2.2.1 Long Round Trip Time and Limited Bandwidth

The majority of wireless systems provide low data-transmission rates to users (e.g., wireless local area network (WLAN), Universal Mobile Telecommunications System (UMTS)). The limited bandwidth is one of the major factors along with channel impairments that degrade the TCP performance over wireless links.

The Internet and other wired data networks provide higher bandwidth than wireless links. The difference in bandwidths between Internet and wireless networks affects the behavior of the last-hop router which connects the wireless network to the other data. The last-hop router receives more TCP segments than it can route through the wireless network. This generates excessive delays due to segments queuing in the router buffer. These delays increase the RTTs of the TCP connections. This limits the increase of the TCP congestion window size, resulting in a limited TCP throughput.

In addition, if congestion occurs, the fast-recovery phase (in the case of triple duplicate) and the slow-start phase (in the case of timeout) become even more harmful. The segments queuing in the last-hop router can result in buffer overflow, can prevent packets from dropping at the router, and can cause congestion to increase.

2.2.2 High Loss Rate

The major cause of packet losses over wireless links is the high level of errors that occur during a transmission. These losses can generate triple duplicate acknowledgments or timeouts. To deal with the problem of losses and errors, the majority of wireless systems implement a retransmission protocol called ARQ (Automatic Respeat reQuest).

The ARQ protocol ensures error-free reception of packets at the receiver over the air interface. It does, however, mislead the TCP protocol that generates unnecessary packet retransmissions through its congestion mechanisms.

2.2.3 Mobility

In ad hoc networks, the mobile nodes move randomly, causing frequent topology changes. This results in packet losses and frequent route discovery algorithms initiation, which significantly reduces the throughput. In cellular networks (e.g., GPRS (General packet radio service), UMTS), the user mobility leads to handoff during the communication. During the handoff process, the necessary information has to be transferred form the previous base station to the new base station. According to the technology used, such as UMTS or HSDPA (High-Speed Downlink Packet Access), the handoff would result in excessive delays or disconnection. The handoff can be intra-technology (also know as horizontal handoff) where both of the cells are covered by the same cellular system and inter-technology (also know as vertical handoff) where different technologies are deployed in adjacent cells, such as handoff between UMTS and HSDPA. The vertical handoff is known to be more harmful, since in some cases the data stored in the RLC or node B buffer is not transferred to the new cell. The lost of data over the air induces severe degradations of overall throughput.

2.2.4 Asymmetric Links Bandwidth

TCP is a self-clocking protocol that uses the incoming acknowledgments in a direction to estimate the RTT and controls the packet transmission in the opposite direction [3]. In both directions, the delay between the received packets (or the received acknowledgments) depends on the link bandwidth of each direction. To have a normal behavior of TCP, the acknowledgments should maintain the same spacing of the transmitted data in the other direction. In wireless systems, the downlink and the uplink do not provide the same bandwidth. For example, in UMTS Release 5, the use of HSDPA provides a higher transmission rate on the downlink. On the uplink, the user continues to use the DCH channel, which provides a limited transmission rate. This causes an asymmetry between the downlink and the uplink.

2.3 TCP Performance Enhancements

To enhance TCP performance over wireless systems, many proposals have been developed which try to approach one of the two following ideal behaviors:

(1) The TCP sender should simply retransmit a packet lost due to transmission errors, without taking any congestion control actions; or

(2) The transmission errors should be recovered transparently and efficiently by the network that is, it should be hidden from the sender. According to their behaviors, the proposed schemes can be divided into three categories [4–7]: link layer solutions, split solutions, and end-to-end solutions. Each type of solution is described to acquire better understanding of the interactions that can occur between radio link layer protocols and TCP.

2.3.1 Link-Layer Solutions

Link-layer solutions try to make the wireless link layer behave like wired segments with respect to higher-level protocols. The basic idea is that errors over wireless links should be recovered in wireless system without including TCP in the recovery process. In other words, these solutions try to mask or hide the error recovery from TCP.

Forward Error Correction (FEC) combined with ARQ protocol is used in the majority of wireless systems to provide the reliable service needed by upper layers, such as TCP. FEC results in inefficient use of available bandwidth. ARQ may cause spurious retransmission at the TCP layer, especially when the wireless link suffers from high-level bursty errors. Neither approach is appropriate from the stand point of efficiency or layer interactions. Additional enhancements must be introduced in the link layer to improve TCP performance. Link-layer solutions can be either TCP aware or unaware.

2.3.1.1 Snoop Protocol

In the majority of the current wireless system, the use of ARQ protocol allows recovery from wireless link errors and provides relatively reliable transfer of packets to the upper layer. However, it was observed that the interaction between ARQ and a reliable transport layer such as TCP may result in poor performance due to spurious retransmissions caused by an incompatible setting of timers at the two layers.

The snooping protocol developed in [8, 9] provides a reliable link layer closely coupled with the transport layer so that incompatibility and unnecessary retransmissions are avoided. The main idea is to introduce an agent in the base station, or wireless gateway, which can snoop inside TCP connections to gather the TCP sequence number, to cache the unacknowledged segment in the base station, and to mask the wireless link errors from the TCP sender by crushing the correspondent acknowledgments.

Snoop does not consist of a transport layer module but of a TCP-aware link agent that monitors TCP segments in either direction. For this, two procedures SnoopData for data segments and SnoopAck for acknowledgments are implemented in the snoop agent at the base station. These two modules work jointly, as follows. The base station monitors all TCP segments received from the wired host, or sender, and maintains a cache of new unacknowledged packets before forwarding them to the mobile host. The snoop agent decides if a packet is new or not by gathering the sequence number inside the segment header. When an out-of-sequence packet passes through the base station, the packet is marked as retransmitted by the sender to facilitate the behavior of the SnoopAck module once it receives a duplicate acknowledgment of this segment. At the same time, the snoop acknowledgment module keeps track of the acknowledgment transmitted from the mobile host. When a triple duplicate acknowledgment sent by the mobile host corresponds to a packet cached in the base station, the SnoopAck deduces that it is due to wireless link errors. So it retransmits the correspondent packet and suppresses the duplicate acknowledgment. Note that packet losses can also be detected by local timeout since the SnoopAck module maintains an update estimate of the RTT. Further details on the SnoopData and SnoopAck modules can be found in [10].

The main advantage of this scheme is that it maintains the end-to-end semantics of the TCP connection. Another advantage is that it can perform efficiently during a handoff process since cached packets will be transmitted to the mobile as soon as it can receive them. In addition, the snoop relies on TCP acknowledgments to detect whether a packet is received or not, which results in small delays in the case of downlink data transmission when the TCP receiver, or the mobile host, is near the base station.

The main disadvantage of snoop is that it requires heavy storage and processing to cache TCP segments. In the case of cellular networks, especially macro-cell, the snoop is impractical due to the high number of users in each cell. In addition, a snoop agent will be implemented in the Radio Network Controller (RNC), for example, in UMTS, and not in the base station, which increases the processing complexity and the storage capacity since each RNC can control more than one base station.

2.3.1.2 Transport Unaware Link Improvement Protocol (TULIP)

To recover from wireless link errors and to prevent spurious triple duplicate, Transport Unaware Link Improvement Protocol (TULIP) was proposed in [11]. TULIP is a link-layer protocol that does not require any TCP modifications. In addition, TULIP is TCP unaware, which means that it does not maintain any TCP state and does not require any knowledge about TCP sessions status. More generally, it is transport unaware since it does not know anything about the transport layer; that is, it can be used for TCP or UDP. On the other hand, TULIP is service aware, which means it provides reliable link-layer services for applications relying on TCP since these application services need this reliability to achieve their QoS and non reliable link-layer services for UDP traffic. The reliability of the link layer is achieved by retransmitting the erroneous packets at this layer without regarding what happens on the TCP layer. The flow control across the link layer is maintained by a sliding window mechanism. To avoid spurious triple-duplicate acknowledgments and unnecessary congestion control triggers, the link layer delivers the received data in sequence to the upper layer.

This scheme presents several advantages. First, it does not require any TCP modifications and can deal with any transport layer. Second, it is very simple to implement and does not require any heavy storage or processing to maintain a TCP state like snoop does. Third, it ensures a local reliability by maintaining local recovery of the packet losses at the wireless link without waiting for TCP acknowledgments.

According to [11], this scheme performs well over WLANs and significantly reduces the end-to-end packet delay more than other link-layer solutions. Note that TULIP resembles the ARQ protocol implemented in UMTS where the RLC (Radio Link Control) layer can use acknowledge and unacknowledged modes and where the RLC SDUs are delivered to the upper layer in sequence. Studies conducted on TCP performance in UMTS have shown that the use of schemes such as ARQ and TULIP does not solve the problem of competing retransmissions due to spurious timeouts generated by an incompatible setting of timers at the two layers.

2.3.1.3 Delayed Duplicate Acknowledgments

This is a TCP-unaware scheme [12] that tries to imitate the behavior of the snooping protocol but makes modifications at the receiver rather than at the base station. Therefore, it is preferred over snoop when encryption is used; the intermediate node, such as the base station, does not have to look at the TCP header. When out-of-sequence packets are received at the TCP receiver (e.g., mobile host, user equipment), the receiver sends duplicate acknowledgments for the first two out-of-order packets. The third and subsequent duplicate acknowledgments are delayed for a duration ***d***. Indeed, this scheme assumes that the out-of-sequence is generated by packet losses over wireless links, and a reliable link-layer protocol such as ARQ is used. Therefore, the erroneous packets will finally be received correctly after a certain delay. Consequently, delaying the third duplicate acknowledgment gives the receiver more time to receive the in-sequence packet and to prevent a retransmission at the TCP layer. If during the delay ***d*** the in-sequence segment is not received, the destination releases the deferred duplicated acknowledgments to trigger a retransmission. Note that determining an appropriate value of ***d*** is mandatory to improve the performance of this scheme. A large value of ***d*** delays the duplicate acknowledgments and results in a long wasted time, especially when the loss is due to congestion, whereas a small value of ***d*** does not give the link layer the necessary time to recover from errors.

This scheme considers that the packet losses are due to wireless link errors and not to congestion. This can be considered as the main disadvantage of this technique.

2.3.1.4 Other Link-Layer Solutions

Many other link-layer solutions have been investigated in the literature. For example, in [13] a new MAC layer protocol called MACAW was proposed and is essentially designed to enhance current MAC layers such as Carrier Sense Multiple Access (CSMA). It proposes to add link-layer acknowledgments and less aggressive backoff policies to reduce the unfairness in the system and to improve the TCP performance.

In [14], the Asymmetric Reliable Mobile Access in Link Layer (AIRMAIL) protocol was proposed. Basically, this protocol is designed for indoor and outdoor wireless networks. It relies on a reliable link layer provided by a combined usage of FEC and local retransmissions like the ARQ protocol, which results in better end-to-end throughput and latency. The FEC implemented in this case incorporates three levels of channel coding that interact adaptively. The coding overhead is changed adaptively to reduce the bandwidth expansion. In addition, this scheme is asymmetric in terms of placing the bulk of the intelligence in the base station and reducing the processing load at the mobile host. Instead of sending each acknowledgment alone, the mobile host combines several acknowledgments into a single event-driven acknowledgment [14]. At the same time, the base station periodically sends a status message. The main drawback of using the event-driven acknowledgment is the delay in receiving acknowledgments and in retransmitting the erroneous packets [6]. This may invoke congestion-control mechanisms and exponential backoff of the timeout timer at the TCP layer.

2.3.2 Split Solutions

To shield the TCP sender from spurious retransmissions caused by wireless errors, several solutions propose to split the TCP connection into two connections at the point where the wired and the wireless networks meet, since these two sub-networks do not have the same characteristics and the same transmission rate. This point is the wireless gateway, or the mobile router, and it changes from a wireless system to another. In cellular networks like GPRS and UMTS for example, it corresponds to the GGSN (Gateway GPRS Support Node) since the base station is not IP capable, whereas in wireless LANs it corresponds to access points. The connection splitting is handled by a software agent implemented in this wireless gateway. The first connection from sender to the wireless gateway still uses TCP, whereas either TCP or other reliable connection-oriented transport protocol can be used between the wireless gateway and the mobile host, or the receiver. Consequently, TCP performance in the first connection is affected only by the congestion in the wired network, and wireless errors are hidden from the sender. The main disadvantage of these solutions is they violate the end-to-end TCP semantics (e.g., Indirect-TCP, or I-TCP). The acknowledgment of a given packet may be sent to the TCP sender before this packet is received by the mobile host, or mobile receiver. If for any reason, such as gateway crashes or poor channel conditions, this packet is not received by the mobile host, there is no way to retransmit it by the TCP sender. Also, these split solutions require heavy-complex processing and storage capacity, or scalability problem, since the packets should be buffered at the wireless gateway until they are acknowledged. In addition, in case of handoff the state information should be transferred from one gateway to another (e.g., base station to another in WLAN).

2.3.2.1 Indirect-TCP

Indirect-TCP [15,16] is one of the TCP proposals to deal with wireless links. This protocol consists of splitting the connection into two connections, the first one between a fixed host, such as a remote server, and a mobile support router located at the beginning of the wireless network, such as GGSN in UMTS or an access point in Wi-Fi. The second connection is established between the mobile host and the Mobility Support Router (MSR) so that faster reaction to mobility and wireless errors can be performed. To establish a connection with a remote server, the mobile host requests the MSR to open a TCP connection with the server, or fixed host. Hence, only an image of the mobile host on the MSR is seen by the fixed host. These two connections are completely transparent to each other. In case of handoff, the state information of each connection is transferred from an MSR to another without reestablishing the TCP connection with the fixed host. This hides the wireless effect from the TCP sender and provides better performance of the overall communication between the mobile host and the fixed host than classic TCP implementations, such as Reno or Tahoe. Although this scheme shields the TCP sender, or fixed host, from the wireless environment effects, the use of TCP in the second sub-connection between the MSR and the mobile host results in performance problems. The frequent packet losses over wireless link trigger several timeouts and congestion control mechanisms, causing the TCP sender to stall [4,5]. Also, the violation of the end-to-end TCP semantics and the heavy processing and storage at the MSR represent the other drawbacks of this scheme.

2.3.2.2 Mobile-TCP

During the handoff process, the wireless connection between the mobile host and the base station is stopped for a while to allow the transfer of the user state information between the base stations. Wireless disconnection is also caused by deep fading conditions over wireless channel. As a result of these disconnections, many TCP segments are delayed or lost, which triggers the congestion-control mechanism and exponential backoff at the TCP sender. This results in low throughput and poor TCP performance. Mobile-TCP (M-TCP) [17,18] was designed to deal with this situation of frequent wireless disconnection and dynamic wireless link bandwidth while maintaining end-to-end TCP semantics. The architecture of this scheme adopts a three-level hierarchy with the mobile host at the lowest level, the mobile support station at the cell level, and the supervisor host at the highest level. The mobile support station has the role of communicating with the mobile host and to transfer the packets to the supervisor host, which controls and manages the connections between the mobile host and the fixed host. The supervisor host controls several mobile support stations, or cells. Using two TCP clients at the supervisor host, the TCP connection is split into two connections: classic TCP connection between the fixed host and the supervisor host; and M-TCP connection between the supervisor host and the mobile host. To preserve the end-to-end TCP semantics, the supervisor host does not acknowledge a segment received from the fixed host until it receives its acknowledgment from the mobile host. This maintains the timeout timer estimate at the fixed host based on the whole round-trip time. In fact, once a TCP packet is received by the TCP client at the supervisor host, it is delivered to the M-TCP client to be transferred over the wireless link to the mobile host. A timeout timer at the supervisor host is triggered for each segment transmitted over the M-TCP connection. When the acknowledgment of a given TCP segment is received from the mobile host, the supervisor host acknowledges the last TCP byte of the data to the fixed host. The TCP sender then considers that all the data up to the acknowledged byte has been received correctly. If the supervisor host timer expires before receiving an acknowledgment for example, due to fading, handoff, or low bandwidth the supervisor host sends an acknowledgment segment for the last byte to the sender. This acknowledgment contains an update of the advertised window size to be set to zero. This freezes the TCP sender, or forces it to be in persist mode, without triggering the congestion-control mechanism and exponential backoff of the timeout timer. The handoffs at the highest level the supervisor host level occur when a mobile moves from a cell controlled by a given supervisor host to another cell controlled by another supervisor host. In other words, the handoffs occur by moving from one supervisor host domain to another and not from one cell to another. Compared to I-TCP, M-TCP allows the mobility of the host to be handled with minimal state information that is, with lower processing and scalability cost [7].

2.3.2.3 Mobile End Transport Protocol (METP)

Mobile End Transport Protocol (METP) [19] is a transport protocol that eliminates TCP and IP layers and operates directly over the link layer. By implementing METP on wireless links that is, on the second connection between the mobile host and the base station the performance problems illustrated previously can be avoided. The base station, or splitting point, acts as a proxy for a TCP connection providing a conversion of the packets received from the fixed network into METP packets. This results in a reduced header since the transport and IP headers are removed. Compared to TCP Reno, METP enhances the throughput by up to 37 percent [19]. On the other hand, this scheme presents many drawbacks. It violates the end-to-end TCP semantics and increases the complexity of the base station with an increased overhead related to packet processing, such as conversion from TCP/IP to METP and vice versa. In addition, during handoff a large amount of information, including states, sending and receiving windows, and contents of buffers, has to be handed over to the new base station, which results in greatly increased complexity.

2.3.3 End-to-End Solutions

This solution category includes changes to TCP mechanics and involves more cooperation between the sender and the receiver (hence the name end-to-end) to separate wireless losses and network congestion. This section presents an overview of the most popular end-to-end TCP enhancement schemes, along with their advantages and drawbacks.

2.3.3.1 TCP SACK

In wireless channels, there is a high likelihood of burst errors that span a few TCP segments in a given window. In this case, the lack of selective acknowledgments in the traditional TCP versions (e.g., TCP Reno) raises a problem. In fact, TCP uses a cumulative acknowledgment scheme that does not provide the TCP sender with sufficient information to recover quickly from multiple losses within a single transmission window. This forces the sender to either wait an RTT to find out about each lost packet or to unnecessarily retransmit segments that have been correctly received. In addition, when multiple segments are dropped successively, TCP may lose its ACK-based clock, which drastically reduces the throughput. To partially overcome this weakness, Selective ACKnowledgment (SACK) is introduced in TCP [20] and is defined as one of the TCP options. Its activation is negotiated between peer TCP entities during the establishment phase of a TCP connection. If no other TCP options are utilized (e.g., timestamps), the SACK scheme makes it possible to inform the sender about a maximum of four losses in a single transmission window. This allows the sender to recover from multiple packet losses faster and more efficiently. Note that the TCP SACK version uses the same congestion algorithm: When the sender receives three selective duplicate acknowledgments, it retransmits the segment and halves the congestion window size. The number of outstanding segments is stored in the so-called variable pipe [21]. During the fast recovery phase, the sender can transmit new or retransmitted data when pipe is less than the congestion window *cwnd*. The value of pipe is incremented by one when a segment is transmitted and decremented by one at each duplicate acknowledgment reception. The fast-recovery phase is terminated by an acknowledgment of all the segments that were outstanding at the beginning of the fast recovery. Finally, even though the improvement gain by using selective acknowledgment is very high compared to Reno, the SACK scheme does not overcome totally the degradation of TCP performance when applied to wireless links. In TCP SACK, the sender still assumes that all packet losses are due to congestion and does not maintain the value of the congestion window size *cwnd* when losses are due to errors. In addition, the SACK scheme does not improve the performance when the sender window size is not sufficiently large [22,23].

2.3.3.2 Forward Acknowledgment

Forward ACKnowledgment (FACK) [24] makes more intelligent decisions about the data that should be retransmitted. However, it is more or less targeted toward improving the performance of TCP when losses are due to congestion rather than to random losses. This scheme incorporates two additional variables *fack* and *retran-data*. The variable, *fack*, represents the forward most data acknowledged by selective acknowledgments and is used to trigger fast retransmit more quickly, whereas *retran-data* is used to indicate the amount of outstanding retransmitted data in the network [24,25]. When the sender receives a triple duplicate, TCP FACK regulates the amount of data that should be retransmitted during the fast-recovery phase within one segment of the congestion-window size as follows:

(forward most data sent) − (fack) + (retran-data).

2.3.3.3. SMART Retransmissions

Self-Monitoring Analysis and Reporting Technology (SMART), proposed in [26], is an alternate proposal to TCP SACK. Each acknowledgment segment contains the standard cumulative acknowledgment and the sequence number of the packet that caused the acknowledgment. This scheme allows the sender to retransmit lost packets only. In addition, it decouples error and flow control by using two different windows: an error-control window at the receiver for buffering the out-of-order packets; and a flow-control window at the sender for storing unacknowledged packets. This scheme reduces the overhead needed to generate and to transmit acknowledgments at the cost of some “resilience to reordering” [4]. Like TCP SACK, SMART retransmissions do not make any difference between losses due to congestion and those due to wireless link errors.

2.3.3.4 TCP Eiffel

In the case of spurious timeouts, the TCP sender proceeds to a retransmission of the segment interpreted to be dropped. When the sender receives an acknowledgment after the segment retransmission, it cannot decide if that acknowledgment corresponds to the retransmitted segment or to the first one. This phenomenon is called *retransmission ambiguity*. The TCP Eiffel was proposed to deal with this situation of retransmission ambiguity [27,28]. For this end, it uses the timestamp option in the TCP header to assign a number to the retransmitted segment [29]. The timestamp value, the current congestion-window size *cwnd*, and the slow start threshold are stored by the sender. Once an acknowledgment is received, the sender compares the timestamp values of the acknowledgment and the retransmitted segment. If the acknowledgment timestamp value is less than that of the retransmitted segment, the sender concludes that a spurious retransmission has occurred. Therefore, it resumes the transmission with new data and stores the congestion-window size and the slow-start threshold values before the spurious retransmission. The advantage of this scheme is that it uses one of the TCP options: It does not require a new standardization. On the other hand, use of TCP options increases the size of TCP segments. TCP SACK uses the header options field to indicate up to four lost segments in an RTT. The use of timestamp or other TCP options limits the number of lost segments indicated by the SACK scheme. To overcome this constraint, Eiffel can use a new specific bit in the TCP header to indicate whether an acknowledgment is for a retransmitted segment or for the original one. The disadvantage of this solution is that it requires a new standardization to specify the new bit in the TCP segment header.

2.3.3.5 Explicit Congestion Notification

The TCP Explicit Congestion Notification (ECN) [30–32] feature provides a method for an intermediate router to notify the end hosts of impending network congestion. This prevents TCP connections, especially short or delay-sensitive connections, from unnecessary packet drop or retransmission timeouts. Basically, the ECN scheme was not proposed to deal with TCP performance over wireless links. However, the congestion-notification mechanism implemented in this scheme can be extended to help the sender differentiate between losses due to congestion and those due to wireless link errors ([48–52]).

In the Internet, the use of the active queue management (AQM) gateway, like random early detection (RED), allows detection of a congestion before an overflow of the gateway buffer occurs. The RED gateway drops the probability of the packets according to the average queue space. The ECN scheme enhances the AQM behavior by forcing the gateway to mark the packets instead of dropping them. This avoids unnecessary packet drops and allows the TCP sender to be informed quickly about congestion without waiting for triple duplicates or timeouts.

To support the ECN scheme, an extension of the TCP/IP stack should be introduced [30–32]. This extension introduces two new flags in the TCP header reserved field: the ECN echo flag and the congestion window reduced (CWR). In addition, two bits in the header are used: ECT-bit and CE-bit. In the three-way handshake connection phase, the TCP sender and receiver negotiate their ECN capabilities by setting the CWR and echo flags in the SYN and the ACK segments; however, in the ACK segment only the ECN echo flag is set by the receiver. In case when both of the peer TCP entities are ECN capable, the ECT-bit is set to one in all packets; otherwise it is set to zero. Once the connection is established, the TCP segments transferred through the network can be dropped or marked by the RED gateways based on the queue status. In fact, if the average queue size, which is an exponentially weighted instantaneous queue size, is between a lower threshold min and a higher threshold max, the probability of the packets are marked using a given function. In this case, the CE-bit is set to one by the RED gateway. The packets are dropped if the average queue length at a given gateway exceeds the higher threshold max. Once the receiver detects a packet with a CE-bit set to one, it sets the ECN echo flag of the subsequent acknowledgment segments until it receives a segment with the CWR flag set. This means that the sender has reacted to the congestion echo by adjusting the slow-start threshold and reduction of the congestion window size. The sender does not retransmit the marked packets, which prevents the connection from wasting time in the fast-recovery phase.

2.3.3.6 Explicit Bad State Notification (EBSN)

The Explicit Bad State Notification (EBSN), proposed in [33,34], tries to avoid spurious timeouts by using an explicit feedback mechanism. A TCP agent is introduced in the base station. This scheme, essentially proposed for TCP downlink data transfers, is not considered as a link-layer solution since it requires a TCP sender support. If the wireless links is in bad state, that is, a packet is received with errors, the ARQ protocol is in charge of retransmitting the packet until it is received correctly. At each packet retransmission, the base station sends an EBSN message to the TCP sender indicating a delay of correct packet transmission. On the receipt of an EBSN message, the TCP transmitter reinitializes the TCP retransmission timer. This provides a reliable link layer that can correctly deliver the packet to the destination without triggering spurious timeouts and unnecessary packets retransmission by the TCP layer. One of the disadvantages of this scheme is that it requires TCP code modifications at the source to be able to interpret EBSN messages. In addition, the base station must be able to keep track of all the packets going through it to note which packets have been acknowledged or not. On the other hand, no state maintenance is required, and the timer has little impact on performance [33,34].

2.3.3.7 Explicit Loss Notification

The snooping protocol described in Section 2.3.1.1 was proposed only for downlink data traffic from a fixed host toward a wireless host. To deal with the degradation of the TCP performance over the wireless uplink that is, when the source is the wireless host a scheme called explicit loss notification (ELN) was developed in [35]. Contrary to snoop, this scheme is not a pure link-layer solution since it requires modifications to the transport layers of remote wired hosts. This scheme consists of implementing an agent at the base station, or wireless gateway, to detect the wireless link losses. The cause of losses either congestion or wireless errors is communicated to the sender through a bit in the TCP segment header called ELN. When this ELN bit is set in the TCP acknowledgment segment header, the TCP sender deduces that the packet loss is due to errors over the wireless link. Unlike snoop, the base station in this case keeps track of the TCP sequence numbers without caching the TCP segments. When a duplicate acknowledgment is detected by the base station, the sequence number of the lost segment is extracted from the ACK. The base station verifies if it has received correctly this segment. If it is not the case, the base station sets the ELN bit in the TCP header and forwards the duplicate acknowledgment to the wireless host. The TCP sender reacts to this ACK with ELN bit set by retransmitting the lost packet without triggering any congestion-control mechanism. Consequently, this scheme detects if the packet losses are due to congestion or to wireless link errors and prevents unnecessary reduction in the congestion window. Although this scheme can detect the exact cause of packet loss, it introduces unnecessary delay for retransmitted packets. Even if the base station detects a packet loss over the wireless link, it should wait for a duplicate acknowledgment from the TCP receiver, or the remote wired host, to ask the wireless host for a retransmission [25].

2.3.3.8 TCP over Wireless Using ICMP Control Messages

The Internet Control Message Protocol (ICMP), proposed in [36], is similar to the EBSN scheme. It allows the sender to distinguish whether losses are likely due to congestion or wireless errors, thereby avoiding the sender from needlessly cutting down its congestion window. If the first attempt of a packet transmission over wireless link fails, an ICMP control message (see [37,38]) called ICMP-DIFFER and containing TCP and IP headers is sent to the TCP sender. Therefore, the sender receives surely either an acknowledgement or an ICMP-DIFFER within one RTT after the packet transmission. At the receipt of the ICMP-DIFFER message, the sender resets its retransmission timer according to the current RTT estimate without changing its congestion-window size and slow-start threshold. The timer's postponement of a RTO gives the base station sufficient time to locally retransmit the erroneous packet [36]. In the case of successive failures of the packet transmission, another ICMP message, called ICMP-RETRANSMIT, is sent to the TCP source. At the same time, the TCP receiver generates duplicate acknowledgments since other subsequent packets have been received. At the receipt of ICMP-RETRANSMIT and the first duplicate ACK concerning the segment indicated by the ICMP message the sender goes into fast-recovery phase and retransmits the erroneous packet. Once a new acknowledgment is received, the recovery phase is ended and the congestion-window size is set to the value before IMCP-RETRANSMIT was generated (i.e., *cwnd* is not halved like in conventional TCP). If neither ICMP-DIFFER nor ICMP-RETRANSMIT are sent to the sender, the TCP connection follows conventional TCP algorithms once a triple duplicate ACKs or a retransmission timeout occur. It is better to generate ICMP messages at the base station rather than at the TCP receiver, since wireless errors may span all the corrupted packets including the TCP and IP header. On the other hand, this solution requires an intelligent and sophisticated base station capable of keeping track of all the packets going through it and to note whether a packet has been acknowledged or not.

2.3.3.9 Non-congestion Packet Loss Detection (NCPLD)

Non-congestion packet loss detection (NCPLD) is an algorithm proposed in [39] to implicitly determine whether a loss is caused by congestion or wireless-link errors. It can be classified into the category of implicit congestion notification schemes. This solution is an end-to-end scheme that requires modifications at the TCP sender only, which facilitates its implementation.

The main idea is to use the concept of the network knee point. The knee point is the point of the throughput load-graph at which the network operates at optimum power [40,41]. Before the knee point, the network is underutilized, and the throughput increases greatly with the network load. In this case, the RTT increases slowly, and it can be assumed that it remains relatively constant. After the knee point, the round trip delay increases highly since the transmitted packets need to be queued at the network routers. NCPLD consists of comparing the current RTT estimate to the RTT of the knee point, called delay threshold. If the estimated RTT is below the delay threshold, the TCP sender considers that the network does not reach yet its knee point and that the loss is due to wireless errors. The packet is then retransmitted without triggering any congestion mechanism. On the other hand, when the RTT is greater than the threshold, the NCPLD scheme is very conservative in this case and considers the loss to be due to congestion. NCPLD complies with the congestion algorithms of standard TCP in this case. Compared to EBSN and ECN, NCPLD is less accurate since it is based on measurements and assumptions. The knee point is not the congestion point of the network. Even if the estimated RTT is greater than the delay threshold, the loss can be due to wireless errors, especially when these errors occur in burst and span more than one packet in a transmitted window, which delays the received acknowledgments and inflates the RTT estimate.

2.3.3.10 Explicit Transport Error Notification

Explicit Transport Error Notification (ETEN), proposed in [42–44] for error-prone wireless and satellite environments, is not a pure end-to-end scheme since it requires modifications at the intermediate network nodes. ETEN assumes that when a loss occurs over wireless link, the TCP segment header, containing the source port number and the TCP sequence number, is still intact. Therefore, the sender notified by an ETEN retransmits the lost packet without needlessly reducing the congestion-window size. This scheme is similar to ECN described already.

2.3.3.11 Multiple Acknowledgments

Multiple Acknowledgments (MA), proposed in [45], is not a pure end-to-end protocol either. However, it is presented in this section since it requires modifications at the TCP sender, meaning the receiver is not modified. This scheme based on some existing solutions, such as snoop, ELN, and I-TCP, combines three approaches: end-to-end semantics, link retransmission, and ELN [7]. It is applied only in the downlink data transmission that is, when the wireless host is the destination and the last hop is wireless. The main idea is to distinguish between wireless errors and congestion using two types of acknowledgments: partial and complete. The partial acknowledgment informs the sender that the packet has been received by the base station, whereas the complete acknowledgment is the normal TCP acknowledgment indicating that the packet has been received correctly by the receiver. In MA, the base station monitors the packets going though it and buffers all unacknowledged packets. A partial acknowledgment is generated by the base station either if a wireless transmission attempt of a packet fails when the base station timer expires before receiving an acknowledgment from the mobile host or when a negative acknowledgment is received or if an out-of-sequence packet is received at the base station and if that packet is in the buffer. The sender reacts to the partial acknowledgment by updating its RTT estimate and postponing its retransmission timer according to the estimated RTT. This prevents the TCP connection from triggering unnecessary retransmissions and congestion mechanisms [7]. In the case where neither partial acknowledgment nor complete acknowledgment are received by the sender, the TCP connection complies with the congestion algorithms slow start, fast recovery, timeout, and window reduction of standard TCP Reno.

2.3.3.12 Negative Acknowledgments

This scheme uses the options field of the TCP segment header to send a negative acknowledgment (NACK) to the sender when a packet is received in errors [46]. Negative acknowledgments indicate to the sender that no congestion has occurred and then there is no need to trigger the congestion control mechanism. The sender reacts to NACK by retransmitting the erroneous segment without changing neither the congestion window size nor the RTT estimate. The main drawback of this scheme is that it assumes that the TCP header of an erroneous packet that holds the port number and the source address remains intact, which is not the case over wireless links where bursty errors may span not only the entire TCP packet but also several data packets in more than one TCP window [6]. The NACK cannot be transmitted to the sender in this case, or it will transfer to a wrong TCP source.

2.3.3.13 Freeze TCP

In wireless systems, mobility and dynamic environment produce a fade period in the received signal. In certain conditions, such as temporary obstacles that block the signal, it is possible that the received signal goes to a deep fade for a very long period of time and that a temporary disconnection and reconnection of the wireless communication may occur. The deep fade may span several windows and may generate segment losses across more than one window. In addition, disconnection and losses that are long in duration can be generated by the handoff process. In certain cases, such as handoff between HSDPA and UMTS Release 99, the data stored in the RLC or node B buffer are not transferred from the original cell to the new cell due to the complexity of the handoff management process in these cases. During long fade durations or handoff process, it is more efficient to stop the TCP transmission, even if no congestion has occurred, though without changing the congestion window size. For this end, freeze TCP was proposed [47]. In case of handoff or long fade durations, the TCP receiver can predict if the actual wireless conditions (e.g., fading level, errors, and duration of handoff according to the wireless techniques used) will cause a temporary disconnection of the wireless link. If a disconnection is predicted, the TCP receiver sets the advertised window size to zero and transmits it to the sender in an acknowledgment segment. Since the TCP sender selects the minimum between the congestion window and the advertised window to be the transmission window, the zero of the advertised window freezes the sender [47]; that is, it forces the sender to be in a persist mode where it stops sending more packets without changing its congestion window and retransmission timeout timer [48]. Once the receiver detects a reconnection, it transmits several acknowledgments to the sender acknowledging the last received packet before disconnection. This process prevents the sender from an exponential backoff and allows the data transmission to resume promptly at the original transmission rate before the disconnection. The timing to send out the zero window size acknowledgment is a critical factor that can affect the performance. Acknowledgment that is too late may trigger timeout and the congestion control mechanism, whereas early acknowledgment causes stop of packets sending earlier than necessary. Experimental results show that the optimal time to send out the zero window size acknowledgment is exactly one RTT before the disconnection [47].

2.3.3.14 TCP Probing

TCP probing, proposed in [49,50], introduces a new error-detection and recovery strategy capable of ruling on the nature of a detected loss. The main idea is to probe the network and to estimate the RTT to deduce the level of congestion and to trigger the responsive recovery mechanism, thereby avoiding needless *cwnd* shrinks and backoffs. When the sender detects a loss via a triple duplicate or a timeout, it goes into a probe cycle, wherein probe segments are exchanged between the sender and the receiver [51]. The probe segment consists of the TCP segment header without any payload, which alleviates the congestion. The probe cycle is terminated when the sender makes two successive RTT estimations. This means that the probe cycle is extended if a probe segment is lost that is, if the error persists. In other words, the probe cycle duration is adapted to the channel conditions, which is very useful in the case of persistent bursty errors or handoff between cells. Moreover, the interruption of data transmission during the probe cycle allows adaption of the TCP transmission rate to the wireless channel conditions, such as capacity or errors [49,50]. Once the probe cycle is terminated, the sender deduces the congestion level using the measured probe RTTs. If the loss is caused by congestion, such as in the case of persistent errors, TCP probing complies with the congestion-control algorithms of standard TCP, such as fast recovery and timeout [51]. Otherwise, the transmission is resumed without cutting down the congestion-window size. TCP probing does not significantly outperform standard TCP versions (e.g., Reno, Tahoe) in the case of transmission with a small congestion window; however, it is more effective in the case of large sending windows since it avoids the slow-start and congestion phase after wireless losses.

2.3.3.15 Wireless TCP

In Wireless Wide Area Networks (WWAN), the link connection suffers from very low and highly variable bandwidth, bursty random packet errors, and large RTT with high variance [51]. This results in highly variable latency experienced by the endpoints, which may trigger spurious timeouts and exponential timer backoffs at the TCP layer. Moreover, the acknowledgments from the mobile host to the fixed host get bunched, which skews the RTT estimation and inflates the calculated Retransmission Timeout (RTO). In [52,53], it was shown that the RTT varies between 800 ms and 4 sec and that the RTO may reach 32 sec. It also was observed in [52,53] that WWAN experience occasional blackouts ranging from 10 sec to 10 min during the course of a day. These blackouts are generated by deep fades, temporary lack of available channel, or handoff between non-overlapping cells. With regard to the hard conditions of WWAN, regular TCP suffers from heavy performance degradation. Advanced TCP implementations that use changes in RTT to estimate congestion do not perform well in WWAN since they can be confused by large RTT variations [52,53]. Specific TCP enhancements are then required. In this context, wireless TCP (WTCP) [52,53] was proposed to improve the TCP performance over WWAN by addressing the causes of throughput degradation. This protocol is designed to be deployable, robust, fair, efficient, and reliable [7]. In the literature many TCP enhancement proposals use the same name of wireless TCP (e.g., [54–56]). In this section, the WTCP presented is the one proposed in [52,53] for WWAN.

WTCP is developed using three key schemes. First, WTCP uses a rate based transmission control rather than the window based transmission control used in TCP. The transmission rate is estimated at the receiver using the ratio of the inter-packet separation at the receiver and the inter-packet separation at the source. The sender transmits the current inter-packet separation to the receiver with each data packet and receives the updated estimated transmission rate from the receiver. The packet loss or the retransmission timeouts are not used in the rate control. This makes WTCP insensitive to wireless losses, non congestion losses, large RTT variations, and bunched acknowledgments. As a result, WTCP is able to handle asymmetric channels and to ensure fairness between competing connections having different RTTs [57].

Second, since RTTs are large over WWAN links and data transmissions may be short-lived, WTCP does not go through the slow-start phase upon startup or to recover from blackouts [52,53]. Rather, it incorporates the packet-pair approach [58] to compute the appropriate initial transmission rate. This is performed by sending two back-to-back packets of maximum size MSS and computing their inter-packet delay.

Third, WTCP ensures the reliability by using periodic cumulative and selective acknowledgment rather than retransmission timeouts, because RTT estimates are skewed by bunched acknowledgments. The sender compares the state of unacknowledged packets in the received acknowledgments to what is stored locally with last retransmission to determine whether an unacknowledged packet is lost or is still in transit. When an acknowledgment is not received by the sender at specified intervals, the sender goes into blackout mode.

2.3.3.16 TCP Peach

TCP peach [59,60] is an end-to-end solution that presents some similarity with TCP probing. This scheme is essentially proposed to solve the slow-start problem in satellite networks and can be used to distinguish between congestion and wireless losses. The main idea is to send low priority segments called dummy segments to probe the available capacity of the network. These dummy segments have the same content as normal segments, but they are served with low priority at the network routers; that is, when congestion occurs these segments are dropped by the routers. Consequently, these segments do not increase the congestion level of the network. This assumes that some priority mechanism needs to be supported at the network routers [59,60]. In TCP peach, dummy segments are generated in the following three phases [59, 60]: sudden start, rapid recovery, and congestion avoidance. In the sudden-start phase of a new connection, the transmission of dummy segments is useful to adapt the slow-start growth since the sender does not know any information about the available resource of the network. When congestion occurs, the sender goes into rapid recovery phase and sends dummy segments to detect the nature of loss. In the case of congestion, these segments are discarded by the network routers; non-acknowledgments for these segments are received. The recovery phase complies with the congestion algorithm of standard TCP versions (e.g., Reno, Tahoe). When the loss is due to wireless link errors, an acknowledgment of the dummy segment is received; hence, the sender retransmits erroneous packet without reducing its congestion-window size. Finally, the dummy segments can be used in the congestion-avoidance phase to probe the congestion level of the network, thereby avoiding congestion before it occurs.

2.3.3.17 TCP Vegas

TCP Vegas [61,62] approaches the problem of congestion from another perspective than classic TCP versions such as Reno and Tahoe. By adopting a sophisticated bandwidth estimation scheme, it estimates the level of congestion before it occurs and consequently prevents unnecessary packet drops. In fact, Vegas estimates the backlogged data in the network (i.e., the congestion level) using the difference between the expected flow rate and the actual flow rate every RTT. The expected flow rate, which represents the optimal throughput that the network can accommodate, is determined by *cwnd*/*baseRTT*, where *cwnd* is the congestion window size and *baseRTT* is the minimum round-trip time. The actual flow rate is determined by *cwnd*/*RTT*, where *RTT* is the current round-trip time estimated as the difference between the current time and the recorded timestamp. During the congestion avoidance phase and using the estimated backlogged data in the network, Vegas decides to linearly increase or decrease the window size to stabilize the congestion level around the optimal point [61,62]. In addition to the congestion avoidance scheme, Vegas incorporates modified slow start and retransmission policies. During the slow start, the sender exponentially increases its congestion window every RTT until the actual rate (*cwnd*/*RTT*) is less than the expected rate (*cwnd*/*baseRTT*) by a certain value *i.*. When a packet is lost or delayed, the sender does not necessarily have to wait for three duplicate acknowledgments to retransmit it. If the current RTT is more than the retransmission timeout value, the packet is retransmitted after the reception of one duplicate acknowledgment. This has the advantage of fast retransmitting lost packets when three duplicate acknowledgments are not received, such as when wireless errors span more than one packet in a window. The main drawback of this scheme is its reliance on the RTT estimate to adjust the congestion-window size. In case of path asymmetry or path rerouting, the estimated RTT is inaccurate and results in skewed estimations of *cwnd* [63]. This may cause a persistent congestion or an underutilization of the network capacity. Finally, it is important to note that Vegas do not make any difference between losses generated by wireless errors and those due to congestion. On the other hand, the approach used to estimate the backlogged packets in the network constitutes the base of other enhanced TCP schemes developed over wireless networks (e.g., TCP Veno).

2.3.3.18 TCP Santa Cruz

TCP Santa Cruz (TCP-SC) [64] makes use of the options field of the TCP header and attempts to decouple the growth of the congestion window from the number of returning acknowledgments. This is gracious in the cases of path asymmetries, wireless links, and dynamic bandwidth systems. TCP-SC uses congestion detection and reaction schemes similar in spirit to those used in TCP Vegas, but TCP-SC relies on delay estimates along the forward path rather than the RTT. In addition, this scheme detects the congestion at an early stage and prevents undesired congestion-window reduction. In fact, TCP-SC determines whether a congestion exists or is developing. It also determines the direction of the congestion by estimating the relative delay a packet experiences with respect to another in the forward direction, thereby isolating the forward throughput from reverse path events. By estimating the congestion level and direction, TCP-SC tries to achieve a target operating point for the number of packets in the bottleneck without generating congestion. Finally, TCP-SC provides a better loss recovery mechanism by making better RTT estimates including the RTT during retransmission and congestion period retransmitting promptly lost packets, and avoiding retransmissions of correctly received packets when losses span more than one packet in one window [64].

2.3.3.19 TCP Westwood

TCP Westwood (TCPW) [65–67] is a rate-based end-to-end TCP scheme where the sender adjusts the slow-start threshold and the congestion window size according to the estimated available bandwidth. The sender keeps track of the rate of received acknowledgments (interACK gap) to estimate the Eligible Rate Estimate (ERE) that reflects the available network resource. TCPW detects a loss upon receiving three triple duplicates or after a timeout timer expiration. In this case, the slow-start threshold and the congestion-window size are adjusted as follows [85,86]: In the case of triple duplicate ACKs, the ssthresh is set to $(ERE * RTTmin)/segmentsize$ and *cwnd* to the minimum between the current *cwnd* and ssthresh, whereas in the case of timeout *cwnd* is set to one and ssthresh to the maximum between two and $(ERE * RTTmin)/segmentsize$. TCPW exhibits better performance and fairness than conventional TCP versions such as Reno and SACK [65–68]. To deal with heavy loss environment, [69,70] proposes an enhanced version called TCPW with Bulk Repeat (BR). This scheme includes a Loss Discrimination Algorithm (LDA) to distinguish between congestive losses and wireless losses using a combination of RTT estimation and the difference between expected and achieved rates. When the sender detects losses caused by wireless errors, it enables the use of the BR mechanisms [69]:

- Bulk retransmission used to retransmit all unacknowledged packets in the current congestion window instead of sending only one lost packet. This is useful to recover promptly from bursty errors that span more than one packet in a window.
- Fixed retransmission timeout instead of exponential backoff of the timer when consecutive timeouts occur.
- Intelligent window adjustment allowing the sender to not reduce *cwnd* to ssthresh after a wireless loss.

2.3.3.20 TCP Veno

TCP Veno [71,72] integrates the advantages of both TCP Reno and Vegas and proposes a scheme to distinguish between congestion and non-congestion states. It adjusts dynamically the slow-start threshold according to whether a packet loss is due to a congestion or wireless errors. In addition, it adjusts the window-size evolution linearly during the congestion avoidance phase. Veno adopts the same methodology as Vegas to estimate the backlogged data in the network. When the number of backlogged packets is below a threshold β, the sender considers that the loss is random, and the connection is said to have evolved into a non-congestive state. In this case, the sender decreases its congestion window by a factor of 1/5. However, when the number of backlogged packets is higher than β, the loss is considered to be congestive, and the connection in this case adopts the Reno standard to recover from losses (i.e., *cwnd* halved and fast recovery). The efficiency and the performance of this scheme depends greatly on the selected value of the threshold β. Experimentally, it was shown that adopting $\beta = 3$ represents a good setting [72].

2.3.3.21 TCP Jersey

TCP Jersey [73] is an end-to-end transport protocol that assumes the use of network routers capable of handling ECN. This protocol is essentially designed for heterogeneous network containing two environments: wired and wireless. The main idea is to adjust the congestion-window size proactively at an optimal rate according to the network condition. For this, it incorporates two schemes: the Available Bandwidth Estimation (ABE) implemented at the TCP sender; and the congestion warning (CW) configuration introduced in the routers. In the case of incipient congestion, the CW-configured routers inform the peer TCP entities by marking all packets going through these routers. This allows the sender to distinguish whether a loss is due to congestion or to wireless link errors. The ABE algorithm allows the sender to estimate the available resource in the network. Consequently, the sender adapts its congestion window size to an optimal value using the information collected via ABE and CW schemes.

2.3.3.22 TCP Pacing

TCP pacing [74] is a hybrid between rate-based and window-based control transport schemes. It uses the TCP window to determine the data flow, or number of TCP segments, to send through the network and relies on the rates to deduce the transmission instant of each data packet. The idea is to achieve a rate controlled packet transmission, thereby avoiding bursty traffic that can result in packet losses, delays, and lower performance, such as frequent triple duplicates and timeouts. Two pacing methods can be used to achieve rate control. In the first one, the sender spreads the transmission of a window of packets across the entire RTT [74] that is, at a rate defined by the congestion control algorithm. The second method consists of spreading the packet transmission by delaying the acknowledgments. This second pacing method is less effective since each acknowledgment may generate multiple data packets at the transmitter. In the literature, many proposals incorporate the pacing scheme in various contexts. In [75], it was used to correct for the compression of acknowledgments due to cross traffic. It also can be used to avoid burstiness in asymmetric networks or when acknowledgments are not available to use for clocking such as to avoid a slow start at the beginning of the connection [76,77] or after a loss [78,79]. Concerning the performance of pacing, it is claimed in [80] that this scheme can improve the TCP performance over long latency links and high bandwidth satellite environments. It is important to note that pacing does not distinguish whether the losses are due to congestion or to wireless errors, which limits its improvement of TCP performance in wireless systems.

2.3.3.23 TCP Real

TCP Real, proposed in [81,82], is an attempt to enhance the real-time capabilities of TCP in heterogeneous wired or wireless data networks. Unlike standard TCP versions where the sender adjusts the congestion window size, real is a receiver-oriented base control that adapts the sending window size by implementing the wave concept presented in [83–85]. The wave concept can be regarded as a congestion window with fixed size during each RTT and is known by both of the TCP end entities. It consists of sending a side-by-side predetermined number of fixed data segments within one RTT so that the receiver estimates the network congestion level based on the successive segments received that is, the perceived data rate. According to the congestion level, the receiver adjusts the wave level, or the number of segments the wave contains, and transmits it to the sender in the options field attached to acknowledgment segments [81,82]. The sender reacts to the ACK by adapting its congestion window to the wave level to the network conditions which allows the sender to avoid congestion and thereby undesired timeout and recovery phases. In the case of path asymmetry that is, when the reverse path is slower than the forward path TCP Real allows decoupling the size of the congestion window from the timeout. Since the asymmetry does not affect the perceived congestion level of the forward path, the congestion window is still intact. However, the RTO should be increased due to the delays of the reverse path, retarding the receipt of the acknowledgments. Potential errors over wireless links do not affect the perceived data rate at the receiver [81,82]. This allows the receiver to distinguish between congestion and wireless errors to recover promptly from wireless losses and to avoid needless congestion-window reduction. When a timeout is triggered, the TCP sender goes into backoff-like standard TCP versions. However, it does not use slow start and adjusts its congestion window quickly to the appropriate wave level when the timeout is due to errors on wireless links.

2.3.3.24 Ad Hoc TCP

Ad Hoc TCP (ATCP) is proposed in [86] to deal with the network partitions problem in the ad hoc network. This scheme is an end-to-end cross-layer solution. In fact, it keeps TCP intact and introduces a new layer between IP and TCP layers called the ATCP layer. In addition, it relies on the use of ECN and ICMP messages to indicate, respectively, whether a congestion has occurred or the network is partitioned. In the case of congestion, the ATCP layer, informed by ECN messages, puts the TCP sender into congestion state that is, the connection goes into the recovery phase. Using the ECN message prevents the sender from waiting for retransmission timeout. In the case of network partition signaled by ICMP messages, ATCP freezes the TCP sender, or forces it into persist mode. Finally, when a loss is generated by wireless transmission errors, ATCP retransmits the erroneous packet for TCP.

2.4. Cross-Layer Design

When several different variables should be taken into account at the same time in order to achieve a truly optimal solution for the adaptation of a TCP originally developed for a wired environment to a wireless scenario, cross-layering should be used.

Such joint optimization can be included in the wide range of recently-proposed solutions for optimizing wireless network design that are labeled "Cross-Layer Design" [87-90]. This approach breaks the ISO/OSI layering principles by allowing interdependence and joint design of protocols throughout different layers.

2.4.1 ILC-TCP (Interlayer Collaboration Protocol)

IILC-TCP [91] was designed to improve the performance of the TCP in wireless environments, involving long and frequent disconnections. The main modification is introduction of a State Manager (SM) in parallel with the protocol stack for gathering information about TCP, IP and link/physical layers. If necessary, this information can be furnished upon the request of the TCP layer. Each layer periodically reports its state to the SM. In case the conditions are not appropriate for TCP flow SM signals the TCP sender to stop sending packets. When conditions have improved, TCP can proceed with regular data delivery.

This approach tries to optimize performance in a scenario in which mobile hosts act as TCP senders. It is an end-to-end approach which requires no changes in the fixed TCP receiver.

The authors in [92] report an improvement of up to 25% in throughput in relation to standard TCP when disconnections and varied mobility patterns are present. However, in the absence of connectivity problems, ILC-TCP offers no improvement in TCP operation.

2.4.2 ATCP.

In this approach, feedback between the network and the transport layers is allowed as well as between the application and transport layers [93]. On the application level, information about priority is specified by the user and interpreted by the transport layer so that priorities can be established.

2.5 Application-aware solutions

Most of the ongoing research in cross-layering focuses on joint optimization of the physical layer and data link or MAC layer [94]. Only recently, approaches that explicitly include the application layer in cross-layer optimization appeared [6–13].

For instance, the proposed scheme in [95] schedules packet transmission over orthogonal frequency-division multiplexing (OFDM) channels giving higher priority to the most important packets (I-frames). The approach takes the size of the queues into account in order to determine the number of OFDM subcarriers to allocate to each user. Also, the observed quality of the channels is explored to assign to each terminal a set of sub-carriers in a good transmission state.

The authors in [96] propose an opportunistic scheduling algorithm for multiple video streams using a priority function depending on channel conditions, importance of frames, queue size, and multiplexing gain.

The authors in [97] propose a cross-layer design for multiuser scheduling at the data link layer, with each user employing adaptive modulation and coding (AMC) at the physical layer.

In [98] application layer adaptation schemes are combined with lower-layer adaptation strategies for low-delay wireless video streaming. Corrupted data is passed across the layer boundaries, and retransmissions and forward error correction are performed end to end at the application layer.

In [99] the authors evaluated different error control and adaptation mechanisms available in the different layers for robust transmission of video, namely MAC retransmission strategy, application-layer forward error correction, bandwidth-adaptive compression using scalable coding, and adaptive packetization strategies.

The authors in [100] propose a new paradigm for wireless communications based on *coopetition*, which allows wireless stations to harvest additional resources or free up resources as well as optimally and dynamically adapt their cross-layer transmission strategies to improve multimedia quality and/or power consumption.

In [101] an application-driven multi-user resource allocation and frame scheduling concept for wireless video streaming is introduced. The approach is based on joint optimization of the application layer, the data link layer and the physical layer. For this, key parameters from these three layers are abstracted. The abstracted parameters at the application layer describe the rate-distortion characteristics of the pre-encoded video streams. At the lower layers they describe the current transmission characteristics of all users. The outcome of the joint optimization leads to adaptive resource allocation at the lower layers and an adaptive decision on which frames to send on the application layer.

In [102] the authors introduced a novel queue management technique for the buffer at the base station of infrastructure 802.11 networks that considers mobile user's receiving characteristics. The maximum amount of the base station buffer that can be used by a given flow is updated proportionally to RTT, measured at the mobile nodes and sent to the base station by the mean of link layer acknowledgements. In this way, the proposed scheme remains transparent to high-level protocols. Results show the advantage of the proposed queue management scheme when compared to that of traditional drop-tail queue management.

2.6 Conclusions

Nowadays, wireless networks are becoming increasingly popular due to the growing use of mobile services. Consequently, significant efforts have been devoted to provide reliable data delivery for a wide variety of applications over a different wireless networks.

By providing an extensive state-of-the-art for Application/TCP over wireless, this chapter reveals most of the currently used approaches to handle interactions between radio link-layer protocols and application/TCP.

We can see that cross-layering is currently one of the most promising techniques for optimization towards high performance in such a challenging environment.

The acknowledgment of our weakness is the
first step in repairing our loss.
- Thomas Kempis

CHAPTER 3

3. Cross-Layer Error Recovery Optimization in Wi-Fi Networks

3.1 Introduction

Wireless Local Area Networks (WLAN) represented by IEEE 802.11 standard, often referred to as Wi-Fi, provide mobile access to networks and services – omitting the requirement for a cable (and fixed) infrastructure, thus enabling fast and cost-effective network organization, deployment and maintenance. Indeed, Wi-Fi has become the de facto standard technology for wireless access in the last mile, specially the so called infrastructure mode in which a base station (BS) is connected to the Internet, allowing mobile nodes (MN) to communicate with fixed users and servers.

As a drawback, the capacity offered by wireless links is relatively low as compared to wired networks. That’s why; the wireless link is usually the bottleneck of the end-to-end path between the fixed sender located in the Wide-Area Network (WAN) and the mobile receiver. Such capacity limitations derive from the very physical nature of the wireless medium, characterized by limited bandwidth, time-varying behavior, interference, etc.

In particular, Bit Error Rate (BER) on wireless links ranges from 10^{-3} to 10^{-1} as opposed to 10^{-6} to 10^{-8} in wired links [103]. This difference of several orders of magnitude results in poor performance of Transmission Control Protocol (TCP) [2] which accounts for over 95% of Internet traffic [104]. The reason for that is in TCP congestion control mechanism which treats all packet losses as congestion related and halves the outgoing rate for every loss detected.

In order to counteract such variation of error rates, IEEE 802.11 standard employs an Automatic Repeat reQuest (ARQ) at the link layer. Following a “stop-and-wait” approach, it does not allow the sender proceeding with next frame transmission until positive acknowledgement is received for the previous frame. Lack of positive acknowledgement triggers frame retransmission until a maximum number of retransmissions is exceeded.

However, the link layer is not the only layer which acknowledges packet delivery: TCP reliability is obtained through the utilization of a positive acknowledgement scheme, which specifies TCP receiver to acknowledge data successfully received from the sender. TCP header reserves special fields for enabling it to carry acknowledgement information. As a result, the TCP receiver can produce a TCP acknowledgment (TCP ACK) as standalone packet or, in case of bi-directional data exchange, encapsulate it into outgoing TCP segments.

Considering data transmission over an IEEE 802.11 link using the TCP/IP protocol stack (Fig. 2), whenever a TCP segment is transmitted over the wireless link, the sender first receives an acknowledgement at the link layer. Then, TCP entity at the receiver generates an acknowledgement at the transport layer. This acknowledgement represents ordinary payload data for the link layer, which should be acknowledged by the link layer protocol of the sender node. As a result, a single application data block is acknowledged three times: one at the transport level and two times at the link layer.

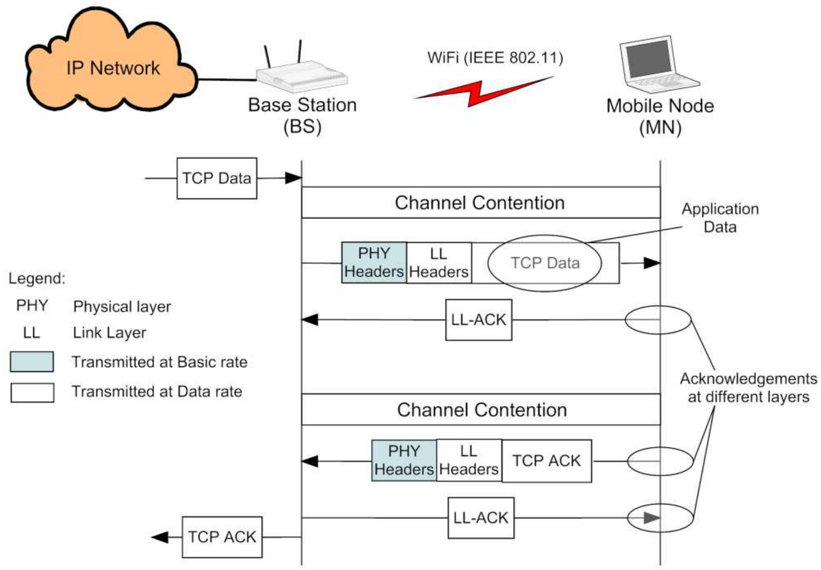


Fig. 2.TCP data packet delivery over IEEE 802.11 wireless link.

In this chapter, we propose a joint optimization of ARQ schemes operating at the transport and link layers using a cross-layer approach called ARQ proxy. The main idea behind ARQ proxy is to substitute the transmission of TCP ACK packet (including the associated physical and link layers overheads) with a small link layer request which is encapsulated into the link layer acknowledgement frame - which does not require any additional bandwidth resources. As a result, ARQ proxy releases network resources associated with TCP ACK transmission over the shared link, thus allowing the corresponding resources to be utilized for a concurrent data transmission originated at any station located within the cell.

The rest of the chapter is organized as follows: Section 3.2 provides design and implementation details of ARQ proxy approach focusing on the infrastructure network scenario; Section 3.3 provides ARQ proxy performance evaluation in terms of TCP throughput and delay performance with the respect to TCP/IP datagram size and wireless link error rate; Section 3.4 concludes the chapter with summary, conclusions, as well as directions for future work on the topic.

3.2 ARQ Proxy

ARQ proxy design is primarily focused on infrastructure network scenario – the most widely deployed WLAN scenario nowadays. Implementation details in single-hop and multi-hop scenarios are discussed afterwards, as they have significant similarities.

The main idea of the proposed approach is to avoid the transmission of standalone TCP ACK packets over the radio channel on the link between the Base Station (BS) and Mobile Node (MN). In order to support this functionality, no changes are needed to the TCP protocol, but new software entities need to be introduced: the ARQ proxy and ARQ client (see Fig. 3).

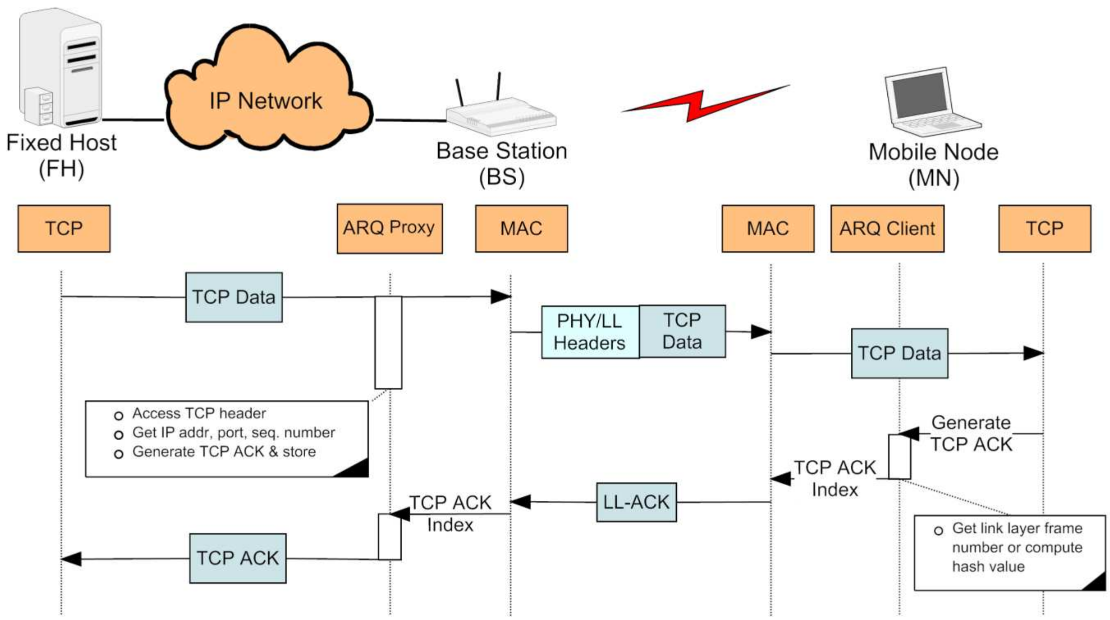


Fig. 3.ARQ proxy and ARQ client functionality.

ARQ proxy is a software module located in the protocol stack of the wireless Base Station (BS) or Access Point (AP). Having access to TCP and IP headers of the in-transit traffic, ARQ proxy generates TCP ACK for every TCP data packet destined to MN which confirms successful data reception up to the flow segment carried in this TCP data packet. ARQ proxy does not require any flow-related state information or TCP layer implementation in a conventional sense. Indeed, TCP ACK is generated using a simple memory copy operation applied to the fields (IP addresses, port numbers, and flow sequence numbers) of the received TCP data packet into a previously generated template of TCP ACK.

The fact that no TCP flow state-related information is used in TCP ACK generation process implies the assumption that all the segments of a given TCP flow are successfully received at the destination node. Since this assumption is not always true, TCP ACKs generated by ARQ proxy module are not released to the Fixed Host (FH) immediately, but stored in BS memory until requested by the ARQ client.

Packet Identification: TCP ACKs generated by ARQ proxy should be easily identifiable by ARQ client without direct communication between these parties. There are two alternative approaches that satisfy this property: frame sequence numbers and hash values (see Fig. 4).

The IEEE 802.11 standard specifies that every sender needs to mark outgoing frames with continuously incremented, 12-bit long sequence numbers at the link layer. The reader should note that in case of TCP/IP datagram fragmentation at the link layer frame sequence number remains the same for all the fragments. As a result, ARQ client located at the MN can indirectly identify TCP ACK generated by ARQ proxy, by referring to the frame sequence number added by the BS at the link layer to the TCP data packet used in TCP ACK generation.

An alternate approach for packet identification that can be used in wireless network with no sequence numbers provided at the link layer is the use of hash values. In this way, TCP ACK is associated with a hash value computed by applying a proper hash function to TCP data packet headers for which the TCP ACK is generated. Traditionally, hash functions are used in cryptography, data storage and search applications. In networking, the use of hash functions is mostly limited to integrity check, error detection and error correction techniques – commonly performed using Cyclic Redundancy Check (CRC) or MD5 algorithms.

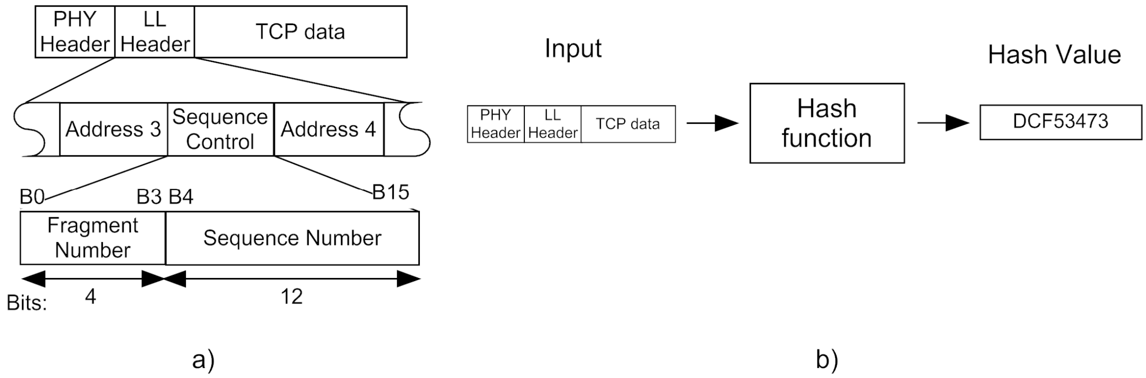


Fig. 4.Packet identification techniques: a) frame sequence numbers and b) hash values.

In this chapter we limit our choice to frame sequence numbers due to simplicity, while for further details on hash functions the reader is directed to [105].

ARQ Client is a software module which logic position is between the link and transport layers of the MN protocol stack (See Fig. 5). It suppresses all outgoing standalone TCP ACK packets and replaces them with MAC layer requests for the appropriate TCP ACK transmission initiated at ARQ proxy.

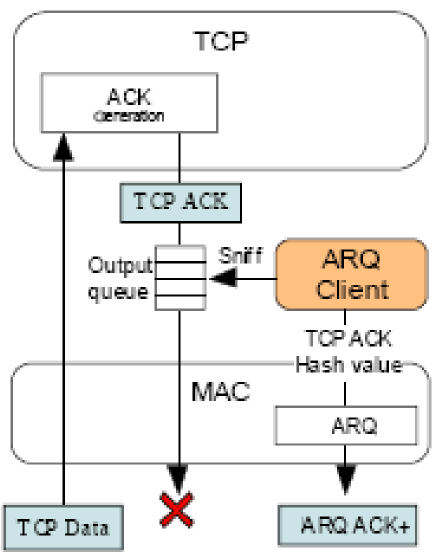


Fig. 5 ARQ Client position in MN's protocol stack.

In order to do so, whenever a standalone TCP ACK is produced at MN transport layer, a TCP ACK suppression request is scheduled for the transmission at the link layer immediately, while the original TCP ACK packet travels down the protocol stack which involves corresponding processing at each layer, output queuing delay, shared medium access and other procedures.

Whichever comes first to the physical layer (the TCP ACK or the corresponding suppression request) will be transmitted, while the other one cancelled.

TCP ACK suppression request includes identification associated with TCP ACK generated by ARQ proxy. This identification depends on the chosen packet identification technique: a frame sequence number or a hash value. At the link layer, TCP ACK identification is inserted into the next outgoing link layer acknowledgement (LL-ACK) frame. In particular, it is inserted into the reserved portion of the "duration" field of LL-ACK frame (see Fig. 6), which does not require modification of the frame structure specified by IEEE 802.11 standard.

Octets:	2	2	6	4
	Frame Control	Duration	RA	FCS

Bit 15	Bit 14	Bit 13-0	Usage
0	0 - 32767		Duration
1	0	0	Fixed value during CFP
1	0	1-4096	TCP ACK index
1	0	4097-16383	Reserved
1	1	0	Reserved
1	1	1-2007	AID in PS-Poll frames
1	1	2008 - 16383	Reserved

Fig. 6.IEEE 802.11 ACK frame with extension capable to carry TCP ACK index.

The use of the reserved portion of LL-ACK frame favors incremental deployment of the proposed technique enabling operation in the mixed network environment where nodes which implement ARQ proxy co-exist with those not implementing the proposed approach.

The lack of TCP flow related information at the BS allows ARQ proxy to generate TCP ACK which acknowledges only in-sequence segment delivery. For that reason, in order to maintain TCP error recovery procedure, ARQ client does not request TCP ACK generated at ARQ proxy in the following cases:

- During TCP connection establishment and connection termination phases, which are explicitly marked by SIN and FIN flags in the packet headers. These packets carry initial sequence numbers, maximum window sizes, and other parameters required by connection setup, and cannot be substituted;
- TCP ACK encapsulated into outgoing TCP data packet. In case of bidirectional data transfer and delayed-ACK option enabled, TCP receiver delays TCP ACKs assuming to have outgoing data segment in order to encapsulate TCP ACK using ACK bit and ACK sequence number fields into the packet header. Consequently, with such encapsulation, TCP ACKs do not create any additional overhead, and thus are not a subject for ARQ proxy optimization.
- Duplicate TCP ACKs. Upon out-of-order segment reception, TCP receiver must generate duplicate ACK for the last successfully received in-sequence segment. Due to the lack of TCP state related information at the BS, duplicate ACKs can not be generated by ARQ proxy and should be transmitted by MN.

- TCP ACK advertising exhausted buffer resources at the receiver (reported in *rwnd* field of TCP header). This ensures the receiver is not running out of the buffer space in case not being able to process traffic at the incoming link rate.

TCP ACKs generated at the BS are associated with a lifetime timer at the moment of generation. Upon expiration of this timer, which recommended value should be equal to or greater than TCP timeout, TCP ACK is silently dropped from the buffer. This lifetime technique is designed to clean up resources from TCP ACKs not requested by ARQ client module.

By the level of the achieved performance improvement the proposed technique is similar to LLE-TCP, proposed by the authors in [106]. However, ARQ proxy conceptually extends LLE-TCP by changing the point triggering TCP ACK generation. In fact, in infrastructure network scenario, LLE-TCP base station is completely responsible for TCP ACK generation with no feedback available from the receiver. On the contrary, in ARQ proxy approach, the generation of all TCP ACKs received at TCP sender is triggered by the receiver following the end-to-end principle of Internet protocol design. Additionally, ARQ proxy avoids the need for storing TCP flow related information at the base station unloading the hardware and enabling application of the technique in scenarios with high mobility.

Implementation of ARQ proxy approach in single hop ad hoc network scenario is the same as in infrastructure network scenario presented above. The only difference is that ARQ proxy module is located at the mobile sender node and not at the BS, and TCP ACKs it produces are not routed through the network but immediately directed to the transport layer following ARQ client request sent by wireless receiver at the link layer.

In a multi-hop ad hoc network scenario, ARQ proxy technique can be applied at the last hop of multi-hop connection.

3.3 Performance Evaluation

In order to analyze the performance of the proposed scheme, the corresponding modules of the ns-2 network simulator (version 2.31) [107] are added supporting ARQ proxy and ARQ client functionality. ARQ proxy module is attached to the BS, while ARQ client is located in MN protocol stack. The configuration of the wireless link between BS and MN follows IEEE 802.11b specification parameters with 11 Mb/s physical data rate. Parameters of the wired link (100 Mb/s, 15 ms) model the situation when a mobile user is connecting to an Internet server physically located within the same metropolitan area. The BS ingress buffer is limited to 700 packets, and RTC/CTS exchange is turned off at the MAC layer as the most appropriate configuration widely used in infrastructure network scenario (see Fig 7).

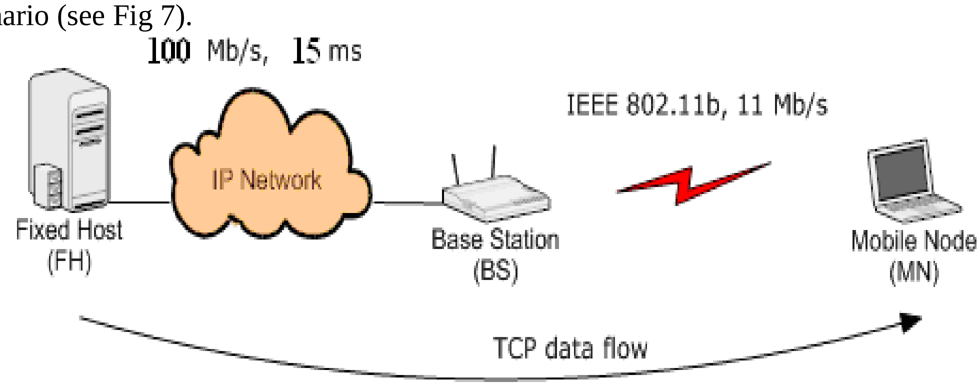


Fig. 7.The simulation scenario

Obtained results are averaged over 10 runs with different seeds used for random generator initialization.

TCP NewReno is chosen for performance evaluation as the most widespread TCP version in Internet nowadays. However, it is important to underline that ARQ proxy approach is not constrained to any specific TCP implementation.

Connection throughput and Round Trip Time (RTT) are chosen as main performance metrics of TCP flow evaluated against variable TCP/IP datagram size as well as Packet Error Rate (PER) on the wireless link.

Fig. 8 shows the throughput level achieved by TCP NewReno for different TCP/IP datagram sizes. The throughput and level of performance improvement of ARQ proxy approach is reversely proportional to the size of TCP data packet. Indeed, the smaller the packet the larger the resources released from TCP ACK substitution. For the maximum considered TCP/IP datagram size of 1500 bytes (which corresponds to the Ethernet MTU), ARQ proxy performance improvement is only 25-30%. For small packets (40 - 200 bytes), it is in the range of 60-70%. However, the general rule is that for TCP data packets which tend to be similar in size to TCP ACK packets the throughput improvement can reach 100%.

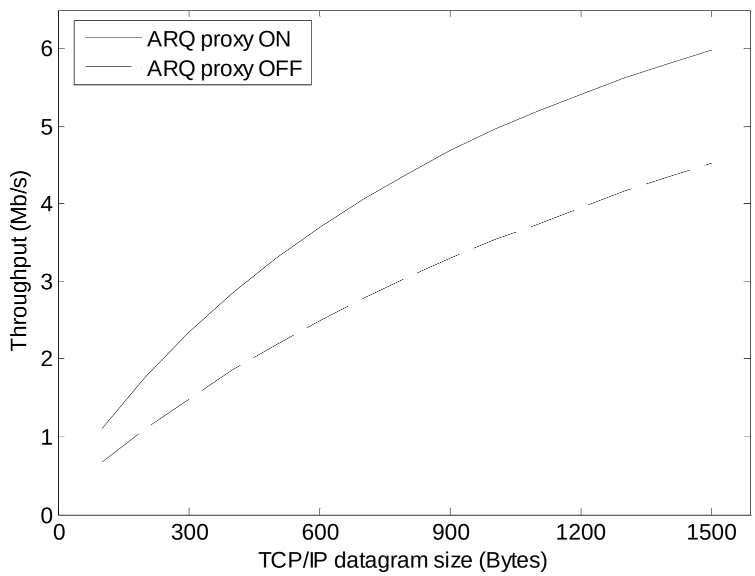


Fig. 8.ARQ proxy TCP throughput comparison.

Along with throughput performance improvement, ARQ proxy reduces RTT of TCP connection. TCP ACKs generated at the BS by ARQ proxy agent avoid transmission, propagation and queuing delays experienced at the wireless link. As it can be observed in Fig. 9, this delay is typically in the order of several milliseconds for IEEE 802.11b.

RTT reduction leads to TCP flow performance increase due to faster window evolution and faster reaction to packet drops performed by Additive Increase Multiplicative Decrease (AIMD) flow control mechanism [108].

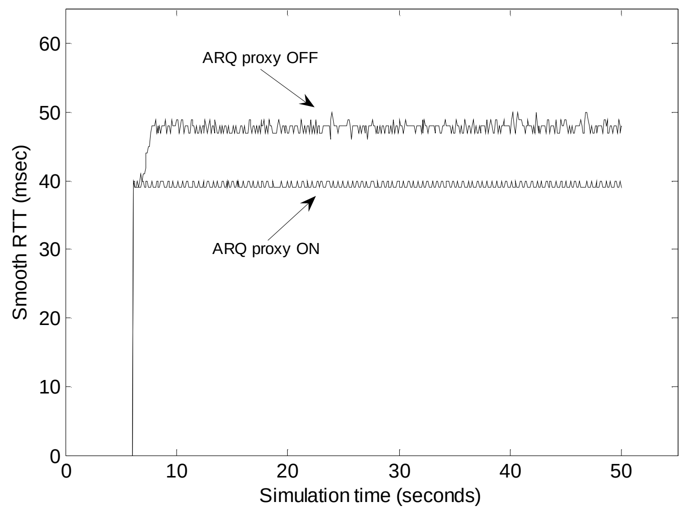


Fig. 9.ARQ proxy Round Trip Time (RTT) reduction.

Fig. 10 illustrates TCP throughput with variable link error rate and TCP/IP datagram size equal to 1500 bytes. While the throughput level is linearly decreasing, ARQ proxy performance improvement remains constant and corresponds to around 30 % for PERs of up to 0.25. Additionally, by enabling ARQ proxy, TCP NewReno is able to sustain higher PERs (see Fig. 10 for PER > 0.25). This is motivated by the fact that in such scenario no wireless link errors propagate into TCP ACKs generated at the base station.

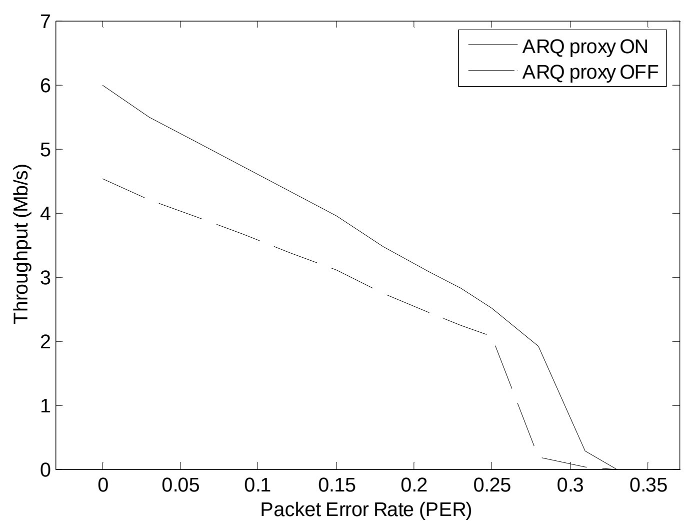


Fig. 10.TCP throughput against wireless link errors.

Summarizing, performance evaluation results validate the proposed approach and confirm ARQ proxy design initiatives. In details, ARQ Proxy provides:

- TCP throughput improvement from 25 to 100%, depending on TCP/IP datagram size;
- RTT reduction for TCP ACK delivery over the wireless link (typically several milliseconds for IEEE 802.11b standard);
- Higher tolerance to link errors.

3.4 Conclusions

The chapter introduced a novel approach aiming at TCP performance improvement in WLAN networks. The core of the work lies in the substitution of the transmission of TCP ACK packets with a short link layer request sent over the radio link. Specifically, TCP ACKs are generated by an ARQ proxy located at the base station based on in-transit traffic analysis and stored in the buffer until requested by the ARQ client (located at the mobile node). All TCP ACK transmissions are triggered by the mobile end-node, which maintain end-to-end TCP semantics.

Two packet identification methods are considered for the proposed scheme: using frame sequence numbers available from the link layer (link layer dependant) and using hash values (link layer independent). In the latter case, hash values can be obtained by applying a predefined hash function onto the raw packet headers' data.

No TCP flow related information is stored at the BS, enabling applicability of the technique in a scenario with high mobility as well as incremental deployment in already operational networks.

Performance evaluation results demonstrate that ARQ proxy brings TCP throughput improvement in the range of 25-100% (depending on payload size), reduction of RTT of the connection for several milliseconds, as well as higher tolerance to errors on the wireless link.

Ongoing activities on ARQ proxy include application of the presented technique in other wireless environments.

If everything seems under control, you're just not going fast enough.

- Mario Andretti

CHAPTER 4

4. Context-aware Receiver-driven Retransmission Control in Wireless Local Area Networks

4.1 Introduction

Wireless technologies represent a networking sector growing at unprecedented rate. They enable fast and cost-effective network organization, deployment and maintenance and allow users to easily join, leave or switch to other networks in a simple and possibly transparent way.

Nowadays wireless technologies are mostly considered in the last mile connecting end-users to the core of the network, while leaving transport of data in the core to wired architectures. As a result, characteristics of wireless links often determine the performance of entire system in terms of capacity and time-varying characteristics having a relevant impact on the user experience.

One of the crucial limitations of wireless networks is related to the high Bit Error Rate (BER) on the radio link, considerably reducing the performance of the widely used Transmission Control Protocol (TCP) [2].

In order to counteract high BERs, most of the wireless network technologies employ Automatic Repeat reQuest (ARQ) protocols at the link layer [4]. Such ARQ protocols are commonly based on a stop-and-wait error control strategy, requiring the sender to wait for a positive acknowledgement for the transmitted frame before continuing with the transmission of the subsequent frame. Lack of positive acknowledgement triggers retransmission.

Such choice of ARQ protocols is mainly motivated by their advantages over Forward Error Correction (FEC) techniques in terms of flexibility and low bandwidth overhead, i.e. ARQ protocols consume bandwidth only for retransmission of erroneous frames, while FEC schemes are typically tuned to the worst case introducing constant bandwidth overhead also in case of correctly transmitted frames.

The performance of stop-and-wait ARQ schemes is determined by *retry limit* parameter, which specifies the maximum number of retransmission attempts taken for a single packet delivery. If the retry limit is reached, the frame is discarded.

In most of wireless network technologies like Wi-Fi, WiMAX, or cellular networks, the retry limit parameter is fixed to a default value, computed to provide a certain radio link BER improvement tuned to an average case (e.g. to compensate typical levels of signal fading and interference for predefined packet sizes).

To counteract the resulting non-optimal performance deriving from such approach, researchers proposed several techniques [109-111] for making ARQ protocols adaptive to the radio link conditions and type of modulation employed. This is achieved by increasing ARQ strength for noisy channels and decreasing it for errorless channels. Such solutions are able to keep the BER provided to upper layers (i.e. above the link layer) of the protocol stack stable.

Summarizing, the use of ARQ protocols, in most of the wireless technologies deployed nowadays as well as in the research proposals available in the literature, aims at compensating radio link bit errors up to the levels in the range of $10^{-6} – 10^{-8}$ required for enabling a reasonable performance of the TCP/IP protocol stack.

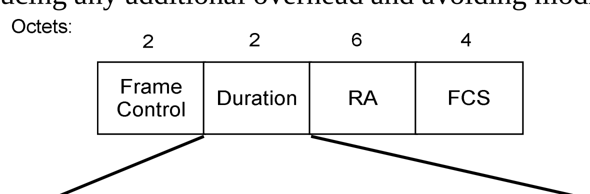
In this work, we argue that further performance improvement can be achieved by tuning ARQ strength based on the application requirements and protocol stack operation on the mobile terminal. Specifically, in the framework of an IEEE 802.11 network, the proposed method allows the mobile terminal to control ARQ strength through specifying the retry limit on a per-packet basis by using a feedback channel. This feedback channel is transparently encapsulated into IEEE 802.11 MAC protocol avoiding the requirement for modifications of the standard.

The rest of the chapter is organized as follows: Section 4.2 provides a detailed description on the proposed approach; Section 4.3 presents performance evaluation results for real-time multimedia and TCP-based data transfer scenarios; Section 4.4 concludes the chapter final remarks.

4.2 Proposed Approach

The proposed Context-aware Receiver-driven Retransmission Link-Layer Control in Wireless Local Area Networks (CORREC) approach enables the mobile receiver to tune the *retry limit* parameter used at the link layer of the Base Station (BS) for the transmission of the next frame.

In IEEE 802.11 WLAN standard, each data frame transmitted at the link layer should be positively acknowledged by the receiver. The acknowledgement frame (presented in Fig. 11) has a reserved portion of 12 bits, which provides a suitable place for organizing a feedback channel between the mobile node and the base station without introducing any additional overhead and avoiding modifications of the MAC protocol.



Bit 15	Bit 14	Bit 13-0	Usage
0	0 - 32767		Duration
1	0	0	Fixed value during CFP
1	0	1-4096	(reserved)
1	0	4097-16383	Reserved
1	1	0	Reserved
1	1	1-2007	AID in PS-Poll frames
1	1	2008 - 16383	Reserved

Bit 13-11	Usage
0	Default ***retry limit***
1-15	***retry limit*** - *1*

Fig. 11.IEEE 802.11 ACK frame.

In fact, only 4 bits are required for the specification of maximum retry limit equal to 14, while the value equal to 0 is used to signal the BS to use the default value for the retry limit.

Packet Importance Metric

For the purpose of the chapter, we extend the definition of packet importance, Imp, given in [112] for VoIP flows to the following:
"The importance of a given packet corresponds to the level of quality reduction for a given flow in case this packet is lost during transmission or corrupted at the receiver."

The notion of flow quality depends on end-to-end application requirements. For example, for VoIP flows, the quality can correspond to Mean Opinion Score (MOS) metric of the flow, for video flows it is commonly represented by Peak Signal-to-Noise Radio (PSNR), and for file transfer applications the quality may simply correspond to the average throughput achieved during the transfer.

File Transfer Applications

Most of the data flows transferred on the Internet are TCP-based. Currently, all the packets produced by TCP layer (connection setup packets, data segments, and acknowledgements) are treated by the link layer equally, i.e. with the same level of error protection (or same value of *retry limit*).

The proposed CORREC approach assigns packets different levels of importance with an algorithm that captures TCP semantics, leading to significant performance improvement.

Fig.12 presents congestion window evolution in TCP New Reno and the corresponding proposed variation of the packet importance metric. Specifically, the proposed approach assigns the highest importance (*High Imp*) to TCP segments produced right after each window reduction and decreases it down to the *Low Imp* threshold following linear or any other function.

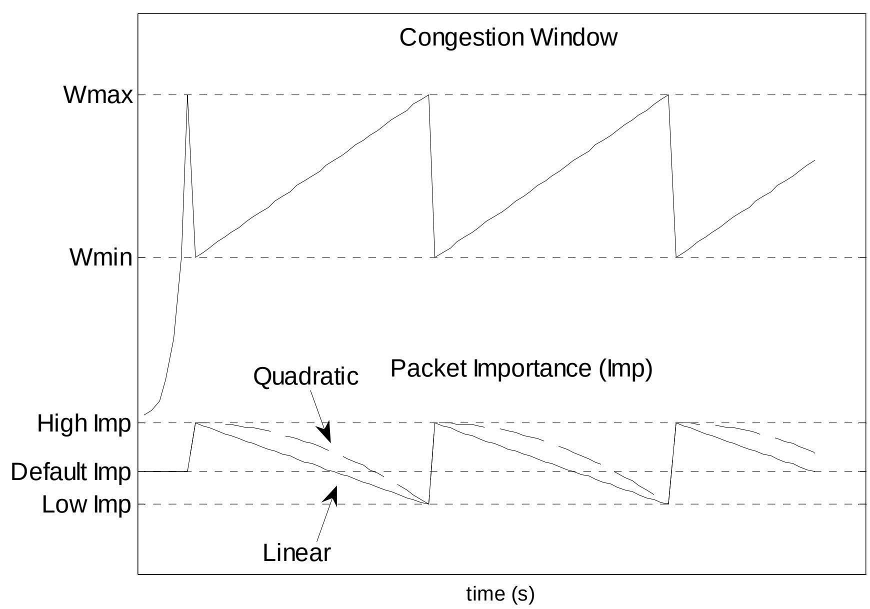


Fig. 12. Packet importance metric for TCP-based data flow.

Each value of the *Imp* parameter has a direct correspondence to the *retry limit* parameter configured at the base station for packet transmission at the link layer. In this chapter, we use *retry limit = 7* for packets with *High Imp* and *retry limit = 0* for packets with *Low Imp*.

The main idea behind the proposed approach is to provide higher protection on the radio link (and more retransmission attempts) when congestion window is small and lower protection for high window values. Indeed, when the congestion window is small, any link error will trigger window reduction to its half unnecessarily reducing the throughput of the TCP flow. In the opposite case, the impact of the link error becomes less significant, since the window will be possibly reduced due to congestion related losses.

Multimedia Applications

CORREC has several applications, ranging from optimizing TCP-based traffic, to VoIP and real-time multimedia traffic. In order to evaluate our scheme for the multimedia traffic case, we selected MPEG-4 video flows which represent the current coding and transport standard for video delivery over the Internet.

In brief, a MPEG-4 video is composed of Groups of Pictures (GOPs). Each GOP includes video frames of three types. The I-Frames (Intra coded frames) are encoded without reference to any other frame in the sequence, and are inserted every 12 to 15 frames as well as at the beginning of a sequence. Video decoding can start only at an I-frame. P-Frames (Predicted frames) are encoded as differences from the last I- or P-frame. The new P-frame is first predicted on the basis of the reference I- or P-frame through motion compensation and then the prediction error is encoded. B-Frames (Bidirectional frames) are encoded as the difference from the previous or following I- or P-frames. B-frames use prediction as for P- frames but for each block either the previous or the following I- or P-frame is used.

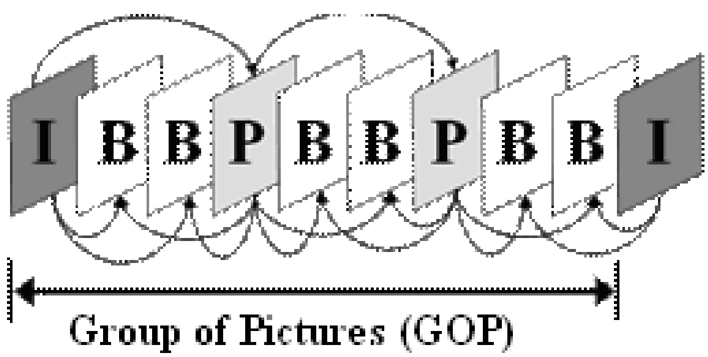


Fig. 13. Prediction encoding of MPEG-4, GOP (N=9, M=3)

The video sequence can be decomposed into smaller units, GOP (Group Of Picture), similar to a deterministic periodic sequence of frames (as shown in Fig 13). A GOP pattern is characterized by two parameters, G (N, M): N is the I-to-I frame distance and M is the I-to-P frame distance. For example, G (9, 3) means that the GOP includes one I frame, two P frames, and six B frames. Similarly, the second I frame in the figure marks the beginning of the next GOP. The arrows indicate that the B frames and P frames decoded are dependent on the preceding or succeeding I or P frames.

Due to the correlation property of P- and B-frames, the effective impact deriving from the loss of an I-frame is much higher than that of P- or B-frame. In addition, the loss of one packet may generate error propagation. While the loss of a B-frame does not affect the quality of the consecutive frames, the loss of an I-frame may disable correct decoding of subsequent P and B frames. This leads to the conclusion that I-frames are more important than P-frames, which are more important than B-frames.

As in wireless LANs a relevant packet loss is expected, we propose that the portion of the multimedia stream which is crucial to the overall quality is to be retransmitted with a higher *retry limit*, i.e. packets belonging to an I-frame are retransmitted with retry limit = 7, while packets belonging to a P-frame and B-frames are retransmitted with a retry limit equal to 4 and 2, respectively.

Similar reasoning is clearly applicable for H.263 and H.264 encoded video streams.

4.3 Performance Evaluation

Scenario Definition

To test the CORREC approach, NS-2 (version 2.31) network simulator [107] is used. Fig. 14 illustrates the considered simulation wireless-cum-wired scenario, where a Mobile Node (MN) is receiving a data flow from a server (the Fixed Host, FH) located on the fixed Internet. The IEEE 802.11b specification parameters, with 11 Mbps physical data rate, characterize the wireless link between the Base Station (BS) and the MN. The FH is connected to the Base Station (BS) using a link providing a data rate of 10 Mbps and one-way delay of 200 ms. The bottleneck buffer located at the BS is of FIFO (First-In-First-Out) and chosen to be limited to 700 packets.

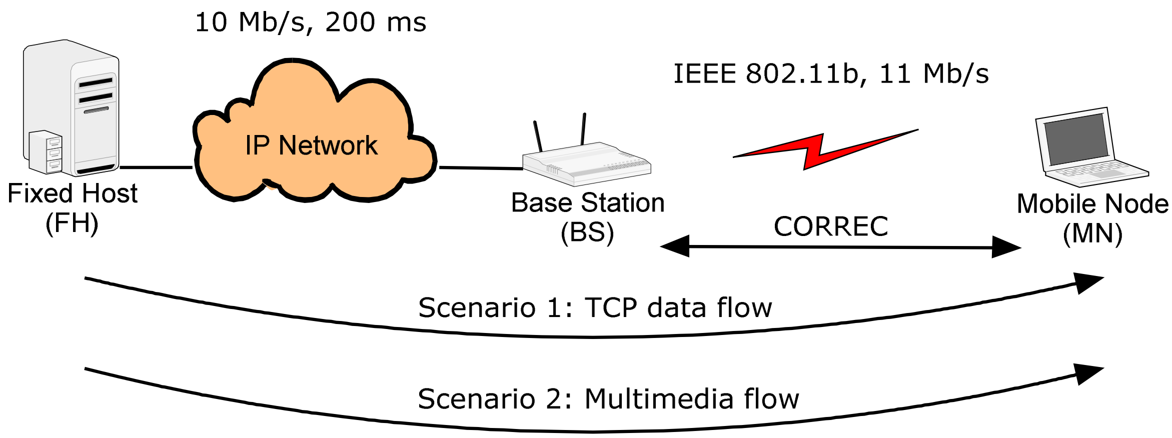


Fig. 14.The simulation scenario.

In the first scenario, an FTP connection (over TCP protocol) is established between the MN and the FH. TCP/IP datagram size is 1500 bytes which corresponds to the most commonly used Ethernet MTU.

In this second scenario, the FH is a video server transmitting video streams, while the video receiver located at MN is connected using the wireless link. Several video traces are used in the experiments. However results are presented in the chapter only for the "Foreman" video sequence, using MPEG-4 video coding. Video format is Quarter Common Intermediate Format (QCIF, 176 x 144). GOP structure is IBBPBBPBBPBBPB. The video trace is composed of 300 frames (10 I-frames, 102 P-frames and 188 B frames) further divided into IP packets. For our simulation, the employed integrated simulation tool consists of a MPEG-4 encoder, a video sender (VS), Network Simulator 2 (ns-2), a MPEG-4 decoder, an evaluate trace (ET) program and a PSNR calculation program. The integrated environment methodology was proposed and developed within the framework of EvalVid [113], enhanced as in [114] for including into NS-2.

The video clip, stored in YUV format, is introduced into an MPEG-4 encoder, which generates an encoded video stream. The open-source ffmpeg MPEG-4 encoder and decoder [115] are used for our experiments. The encoded video stream is read by the Video Sender (VS) for generating a trace file, which contains information such as frame type, size, etc. for each video frame. The trace file is then introduced to the streaming server in the ns-2 simulator to produce video streams in the network simulation platform (ns-2). The effect of streaming video over the network is captured in a streaming client log file, which is generated by NS-2. The log file contains information such as timestamp, size and ID of each packet, both for the client and the streaming server. The trace file and the log files are used by the Evaluate Trace (ET) program to generate the corresponding corrupted video files as a result of transmission over the considered network. The corrupted video file is then processed by the Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS) calculation modules to evaluate end-to-end video quality (See Fig. 15).

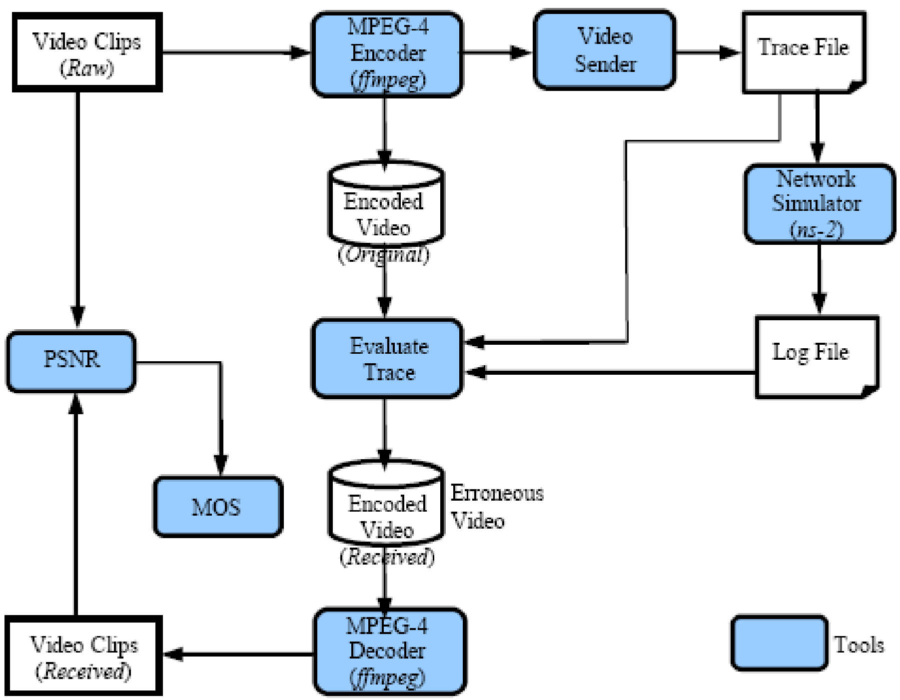


Fig. 15.Structure of the Integrated Tool Environment

Average throughput of TCP connection is chosen as the main metric for evaluation of the proposed scheme in application data transfer scenario, while PSNR and the achieved goodput are the metrics for the second scenario, to evaluate the advantages enabled by the proposed approach if applied for multimedia flows.

Results: TCP flows

In this scenario, a TCP data flow is considered. Prediction of the congestion window (*cwnd*) of the server is performed by simply counting the number of TCP packets successfully received between two consequent events of error detection at the receiver (with three duplicate ACKs). This measure provides the length of the linear increase phase between two window reductions. Dividing this length by the difference between the maximum and the minimum value of the retry limit, a packet interval is obtained, after which the retry limit is decreased by 1 from retry limit max = 7 to retry limit min = 0.

Fig.16 provides a comparison between the achieved TCP throughput using CORREC approach and the legacy scenario for variable link error rates. The main advantages come for PERs higher than 10^{-4} due to non-congestion related reduction of window size and unnecessary degradation of TCP throughput.

Results: Multimedia traffic

In the second scenario, a MPEG-4 video flow is transmitted on the topology presented in Fig. 14. The GOP structure is given by n (the repetition rate of I-frame in data sequence) and m (the repetition of anchor frames I- or P-frames). For instance, a GOP structure of IBBPBBPBBPBBPBB has n = 15 and m = 3.

Fig. 17 compares the achieved PSNR against wireless link errors in 3 different cases:

1. with no link layer retransmissions performed (corresponds to with the retry limit equal to 0);

2. With retry limit constantly set to be equal to 3 for all the outgoing packets; and

3. With dynamic retry limit varied according to the proposed CORREC approach.

The results demonstrate significant performance improvement for the latter case (i.e. when CORREC is used). The quality of video remains stable for PERs of up to 10^{-2}.

Fig. 18 presents two superposed histograms of the packets successfully received by MN's decoder in order to compare the number of packets delivered by using CORREC scheme versus the number of packets delivered using standards scheme with retry limit constantly set to 3. It also underlines that the gain in video quality depends on the selective protection of I- and P-frames. The ratio between the packets sent and packets received is highlighted in Fig. 19.

Fig. 20 shows sample snapshots of the "Foreman" video for PER = 0.15, comparing the visual quality in case retry limit = 3 and when CORREC is employed. The resulting PSNR is equal to 21,88 db for standard scheme and 33,15 db for CORREC.

4.4 Conclusions

In this work, a flexible and dynamic per-packet differentiation of link layer ARQ protection driven by end application requirements is proposed. The method is applicable to data, voice and video flows. Experimental results demonstrate the potential benefits deriving from the proposed strategy.

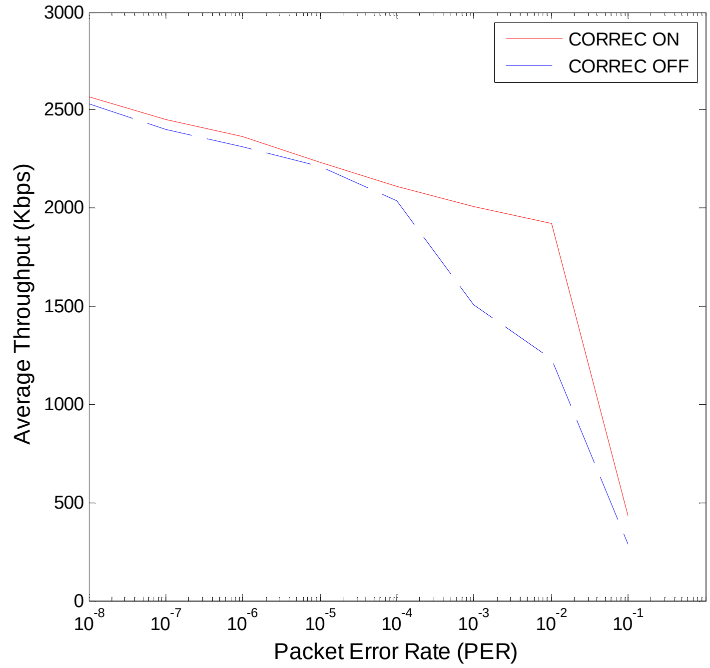


Fig. 16.Performance of CORREC in terms of TCP throughput against packet error rate.

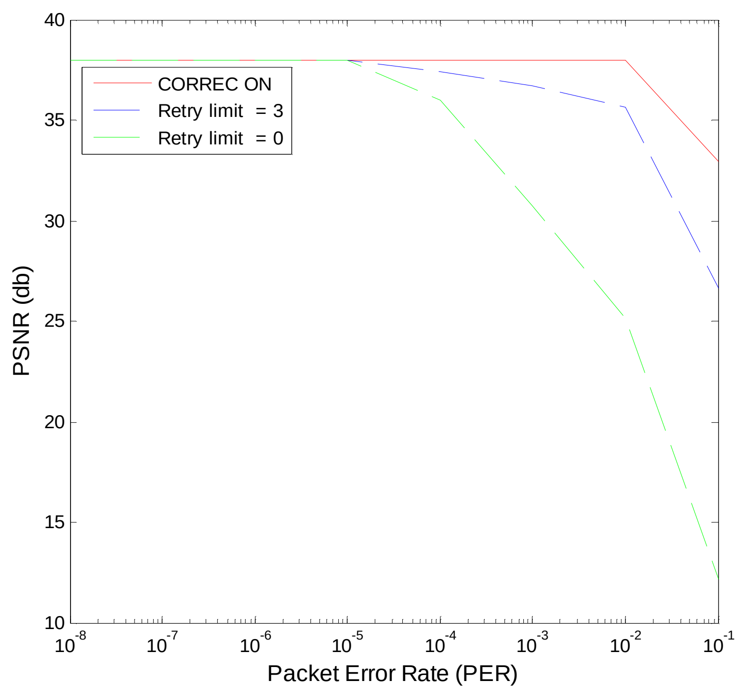


Fig. 17.PSNR against wireless link errors in case of static settings and with CORREC enabled.

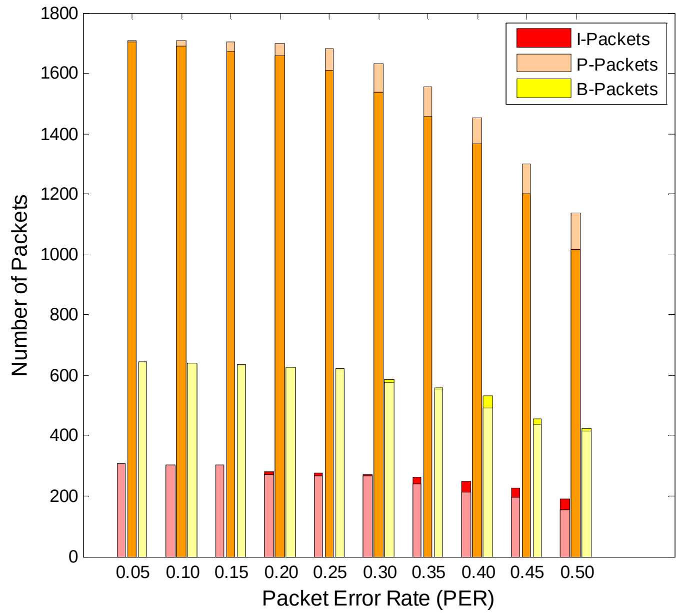


Fig. 18.Number of I-, P- nad B- packets correctly received against PER on the wireless link.

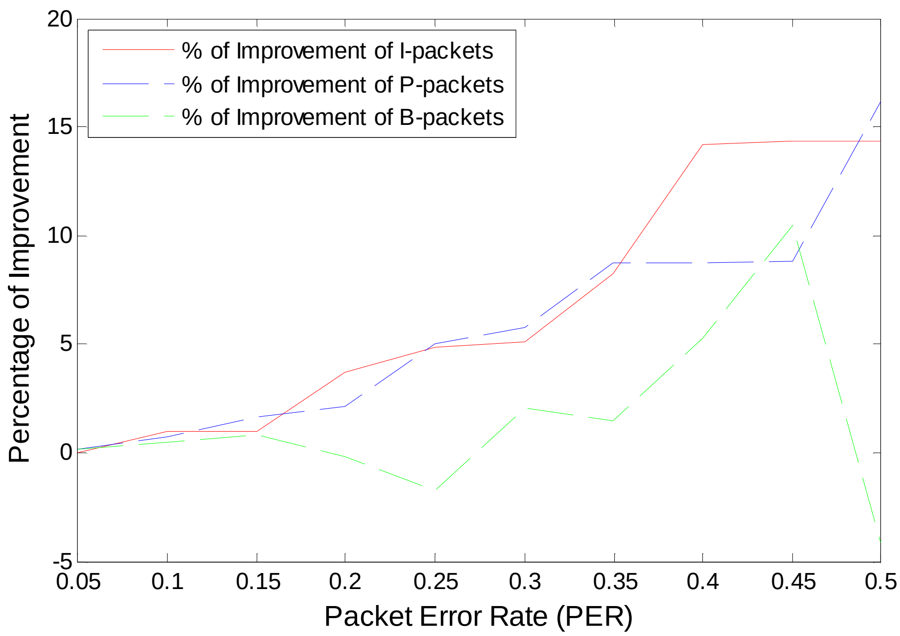


Fig. 19.Level of improvement in terms of the ratio between packets sent and packets received of the same type (I, P, and B).

CORREC OFF (PER = 0.15, Average PSNR = 21.88db)

CORREC ON (PER = 0.15, Average PSNR= 33.15 db)

Fig. 20.Visual comparison of achieved performance using the legacy scheme (upper) and the proposed scheme (CORREC, lower).

CHAPTER 5

5. Service-Aware Retransmission Control in Cellular Networks

5.1 Introduction

Nowadays, IP networks and the Internet in particular are used as transport facilities for a whole plethora of novel applications that go far beyond the data transfer for which IP was originally designed. Those applications introduce specific requirements in terms of delivery performance of the underlying transport infrastructure. Indeed, as services are evolving to a “triple play” vision, implying delivery of data, voice and video to the end user using the same IP transport facility, strong emphasis is put on providing a satisfactory user experience and as a consequence on identifying techniques able to control packet losses and delay.

In addition, more than 30% of the current Internet users are mobile, i.e. use wireless networks to access the Internet and its services. The usage of a wireless access technology increases the complexity in the management of delivery of data and multimedia flows, due to the time-varying performance of the wireless medium, handover management, etc.

While no solution for end-to-end quality of service (QoS) assurance over heterogeneous networks is available, still several approaches are available for improving data transfer performance on the wireless access trunk [121]. However, most of the available solutions are flow-based (i.e., intended to differentiate services based on flows) and furthermore need to introduce relevant modifications to the protocol stacks on the mobile node and wireless based stations which reduces the possibility of deployment of such schemes.

In the specific framework of multimedia (e.g. voice and video), several works are available based on the Unequal Error Protection (UEP) paradigm [116-120]. The goal of UEP is to provide higher protection to the most perceptually relevant data, where protection can be achieved through means of adaptive power levels, forward error correction codes, retransmission control, etc. Nevertheless, since UEP is usually performed or managed at source level and thus without specific knowledge of the contingent operating scenario, such solutions (while increasing the complexity of multimedia codecs) can lead to non-optimal performance due to waste of available capacity in case network / channel conditions are good (and no packet drops are experienced) or time-varying performance of the transport infrastructure (particularly true in the case of wireless networks).

The proposed scheme represents a novel paradigm of dynamic and “link-level” UEP, focused on the access network and the actual “reception history” at the receiver. The scenario is “triple-play” service delivery over 3G cellular networks, with specific focus on the wireless link between the Base Station and the User Terminal. The core idea is to adaptively tune the level of HARQ protection based on the relative importance on the overall user experience of the packet being transmitted by the base station. The introduction of such term enables to differentiate protection on the basis of the actual content of the packet; for voice and video flows, the impact of losing the current packet (and the corresponding required level of protection) is estimated in terms of the potential decrease in audio or visual quality as measurable by MOS or PSNR, respectively. The authors agree that other automatic means for finer and more accurate evaluation of the multimedia quality are available in the literature (e.g. E-model for audio, V-model for video). However, the method is presented using PSNR for sake of simplicity (as the focus in on the overall approach and not on the specific building blocks), while the introduction of more sophisticated algorithms based on finer models or Rate-Distortion characteristic is possible due to the modular architecture of the proposed solution.

An important aspect to be underlined is that the above concept is applicable also in the case of data transfer. In this case, assuming data flows are transported by TCP (which is true for more than 80% of the Internet traffic in the Internet), packet losses can have a different impact on the overall performance (in terms of time required to complete the delivery) due to the corresponding modifications of the congestion window evolution.

The structure of the chapter is as follows: Section 5.2 describes in details the proposed framework, while performance evaluation is presented in Section 5.3. Finally, Section 5.4 concludes the chapter with final remarks and outlines about future work on the topic.

5.2 Proposed Approach

The main idea of the proposed approach, called Service-Aware Retransmission Control (SARC), is to allow the mobile terminal receiver to control the level of HARQ protection applied by the base station for every frame transmitted on the radio link. The decision of the mobile terminal is based on the potential benefit in correctly receiving the next packet given the current reception history and the actual perceptual relevance of the packet itself.

Automatic Repeat Request (ARQ) is an error detection mechanism used in UMTS, where the transmitter uses a stop-and-wait procedure, transmitting a data block and waiting for a response from the receiver before sending a new data block or retransmitting an incorrectly received data block. As an evolution of such approach, Hybrid ARQ (HARQ) scheme is used in HSDPA, where incorrectly received data blocks are not discarded but stored and soft-combined with successive retransmissions of the same information bits.

3GPP specifications have defined two HARQ processes for HSDPA: Incremental Redundancy and Chase Combining [123]. In the former scheme, successive retransmissions of an incorrectly received data block are sent with additional redundancy that is increased with each consecutive retransmission. The retransmissions consist of redundant information in order to increase the chances of successful delivery. Since each transmitted block is not the same as the previous transmission, it is demodulated and stored at the receiver and consequently soft-combined to reproduce the original data block [126]. In the chase combining strategy, an erroneously received data packet is stored and soft-combined with later retransmissions that are an exact copy of the original transmission.

HSDPA uses HARQ (Hybrid Automatic Repeat Request) retransmission mechanism with Stop and Wait (SAW) protocol. HARQ mechanism allows the User Equipment (UE) to rapidly request retransmission of erroneous transport blocks until they are successfully received. HARQ functionality is implemented at MAC-hs (Medium Access Control - high speed) layer, which is a new sub-layer introduced in HSDPA. MAC-hs is terminated at node B, instead of RLC (Radio Link Control) which is terminated at RNC (Radio Network Controller). This enables a smaller retransmission delay (< 10 ms) for HSDPA rather than UMTS Rel. 99 (up to 100 ms).

In this chapter, the level of HARQ protection (also indicated as “HARQ Strength” in the following) is considered in terms of the maximum number of retransmission attempts taken for a packet delivery in case of failure.

Fig. 21 illustrates architectural principles of the proposed SARC approach. As outlined in the previous sections, SARC operates on the wireless 3G link. At the mobile terminal side, whenever a packet is received by the application, the latter can specify packet importance for subsequent incoming packets for a given flow. The packet importance is then transferred into corresponding values of HARQ protection by the SARC module (implemented within the protocol stack at the mobile terminal) and delivered to the HARQ entity at the link layer of the Base Station using cross-layer signaling. At the link layer, the specified HARQ protection parameter is sent along with HARQ acknowledgement, which is generated for every frame received according to stop-and-wait HARQ type.

The SARC module implemented at the BS analyses incoming traffic (assuming to have access to TCP and IP protocol headers) and specifies the HARQ entity to use the requested HARQ protection on a per-packet basis.

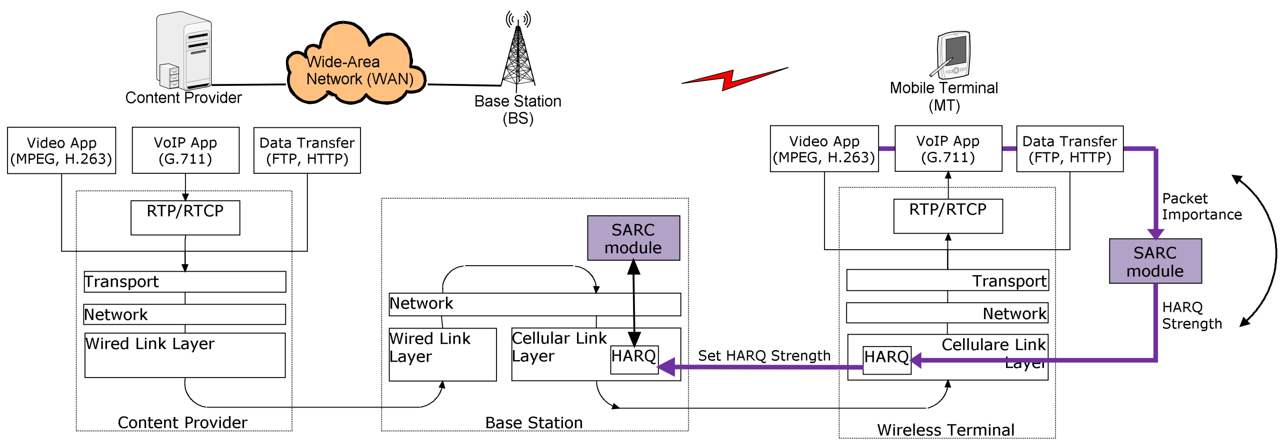


Fig. 21.Architectural principles of SARC in a 3G cellular network. Blocks and links highlighted in "blue" underline the modules and signaling links employed by SARC approach.

Packet Importance Metric

The adaptation of the level of HARQ protection based on the content carried in the packet payload represents a solution belonging to the framework of cross-layer service-aware networking solutions, which optimize "pure" networking techniques based on services and their traffic demand.

In this chapter, we address three different classes of services; voice, video, and data transfer; improving delivery performance by adapting network response to the relevance of packet being delivered over the radio link.

The level of HARQ protection in the proposed approach varies on the basis of a packet importance metric, which consists of two components:

- *Initial packet importance* corresponds to the level of quality reduction for a given flow in case the packet is lost during transmission or corrupted at the receiver [124]. The quality of the flow is determined by end-to-end application requirements and user demands. For example, commonly used metric for VoIP is Mean Opinion Score (MOS), for video is Peak Signal-to-Noise Radio (PSNR), and for TCP-based data is transfer throughput level.

- *Dynamic packet importance* component accounts for the "reception history" of the flow and adjusts initial packet importance. For example, the importance of frame *i* in a video sequence can be dynamically adjusted in case its decoding depends on the neighboring frames *i-1* and *i+1* and frame *i-1* is not correctly received.

Packet Importance Metric in Video Streams

For sake of a clear explanation, we consider a scenario with a mobile node receiving MPEG-4 video flows from a streaming server located in the wired Wide-Area Network (WAN). However, similar reasoning can be applicable to H.263 and H.264 encoded video streams, as well as embedded video streams. The Base Station (BS) serves as a gateway between fixed and wireless network segments.

An MPEG-4 video is composed of Groups of Pictures (GOPs), consisting of video frames of three types. I-Frames (Intra coded frames) are encoded without reference to any other frame in the sequence, and are usually inserted every 12 to 15 frames as well as at the beginning of a sequence. Video decoding can start at an I-frame only. P-Frames (Predicted frames) are encoded as differences from the last I- or P-frame. The new P-frame is first predicted on the basis of the reference I- or P-frame through motion compensation and encoding of the prediction error. B-Frames (Bidirectional frames) are encoded as the difference from the previous or following I- or P-frames. B-frames use prediction as for P-frames but for each block either the previous or the following I- or P-frame is used.

Due to the correlation property of P- and B-frames, the effective impact deriving from the loss of an I-frame can be clearly considered much higher than that of P- or B-frame. In addition, the loss of one I- or P-packet may generate error propagation: while the loss of a B-frame does not affect the quality of the consecutive frames, the loss of an I-frame may disable correct decoding of subsequent P- and B- frames. This leads to the conclusion that I-frames are more important than P-frames, which are more important than B-frames.

To validate the above considerations, Fig. 22 shows the quality reduction of a real video flow transmitted using VideoLan software [127] in terms of PSNR measured at the receiver versus the loss of different types of packets within a GOP. The horizontal scale indicates which frame within the GOP was lost, while the first value (obtained with no losses) serves as a reference point.

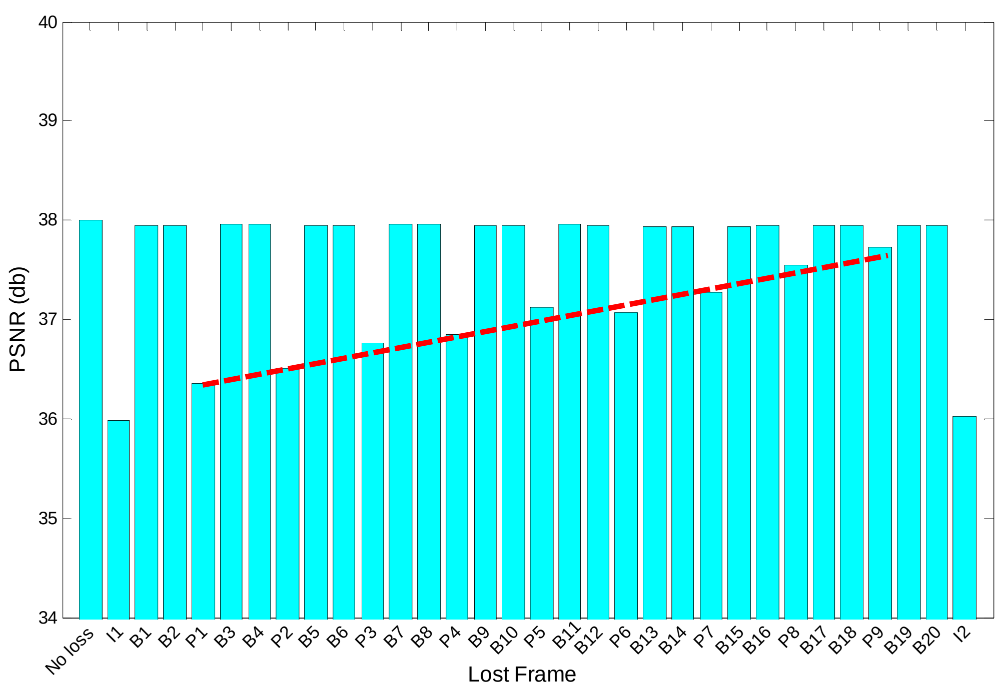


Fig. 22. Quality of the received video flow for different frames lost.

The highest loss in PSNR quality corresponds to the case when the I-frame is lost - making decoding of the entire GOP either not possible or prone to error propagation. On the other hand, the loss of any of B-frames does not degrade the quality for more than a minor fraction. However, the loss of a single P-frame has high influence on the video quality and the level of its degradation depends of the on the relative position of the lost P-frame within the transmitted GOP sequence: higher quality degradation is measured for P-frames losses located closer to the beginning of the GOP.

Following such observation, the importance of P-frames P_{imp} is defined ranging linearly from I_{imp} to B_{imp}, where I_{imp} is the importance level of I-frames and B_{imp} is the importance level of B-frames with $I_{imp} \geq P_{imp} \geq B_{imp}$. Indeed, the loss of P1 (which follows immediately after the reference I-frame) leads to almost the same drop in PSNR as the loss of the I-frame, while the loss of P9 which is transmitted right before the last pair of B-frames leads to PSNR loss comparable with those caused by B-frame losses. The bold line presented in Fig. 22 is obtained by curve fitting with the first order polynomial RSM model for PSNR values achieved for different P-frames lost. The obtained R-square equal to 0.97 shows good match between the experimental data and the proposed linear model and, as a result, for the chosen P-frame packet importance.

Packet Importance Metric in VoIP Flows

A VoIP application is composed of several building blocks. (See Fig. 23). At the sender side, the first component is an encoder, which periodically samples the voice signal.

A large variety of Voice-over-IP (VoIP) encoders are available, representing different trade-offs between quality and bandwidth consumption. Encoders can be either sample based (e.g., G.711) or frame based (e.g. G.729), periodically coding individual speech samples or grouping a certain number of samples within a time window, respectively. A number of speech frames can be multiplexed into the same packet payload, so as to reduce the overhead of transport, network and MAC headers, though at the expense of increasing the transmission delay. VoIP speech payload is typically encapsulated into RTP/UDP/IP packets.

At the receiver side, speech frames are de-multiplexed and inserted into a playout buffer. The playout buffer plays an important role in perceived speech quality since it enforces speech frames delivery at the same interval at which they are generated by the encoder. This is done through re-ordering, delaying or even dropping the frames which arrive later than their expected playback time. However, whenever the frame is dropped it causes a relevant decrease of the quality of the voice stream.

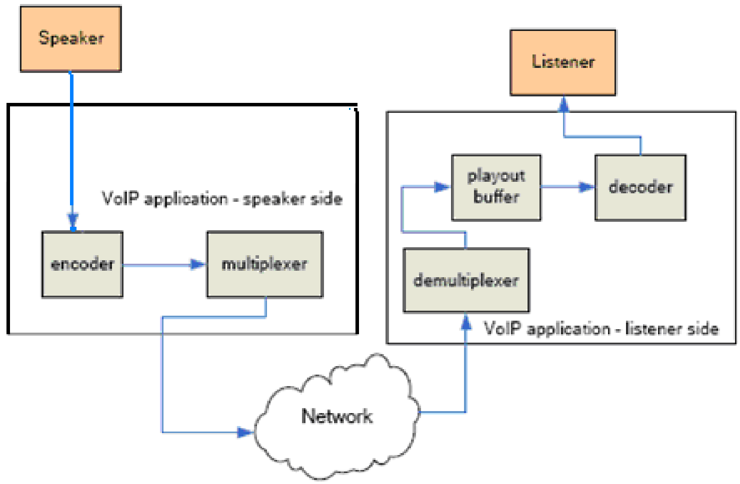


Fig. 23. Scheme of a VoIP application

Based on the above, initially, equal packet importance (i.e., "initial packet importance") is associated to all transmitted speech frames. However, in case the receiver detects frame losses after out-of-order frame reception, it increases importance (and error redundancy) for the subsequent packets of the stream (i.e. increases the "dynamic packet importance"). Summarizing, SARC aims at avoiding bulk frame losses, which are critical for the quality of the speech stream, while single frame losses can be easily compensated or concealed by the decoder.

Packet Importance Metric in File Transfer

TCP is the most widely used protocol in Internet and it provides a flow of equally-important packets for the user viewpoint. However, depending on the context (e.g. the evolution of the TCP congestion window), packet losses can severely decrease the data transfer performance.

The proposed SARC scheme dynamically adapts the level of HARQ protection used on the radio link based on the value of the TCP congestion window computed at the receiver node.

The core idea is to provide higher protection on the radio link (and more retransmission attempts) when congestion window is small and lower protection for high window values. Indeed, when congestion window is small, any link error will trigger window reduction to its half unnecessarily reducing the throughput of the TCP flow. In the opposite case, the impact of link errors becomes less significant, since the window will be possibly reduced due to congestion-related losses.

Fig. 12 presented in the previous chapter presents congestion window evolution in TCP New Reno and the corresponding proposed variation of the packet importance metric. Specifically, the proposed approach assigns the highest importance ("High Imp") to TCP segments produced right after each window reduction and decreases it down to the "Low Imp" threshold following linear or any other monotonically decreasing function and defined as follows:

$$Imp(w) = \begin{cases} -f(w) \cdot k, \text{if } linear \\ -f^2(w) \cdot m, \text{if } quadratic \end{cases}, \tag{1}$$

where

$$k = \frac{|Imp_{\max} - Imp_{\min}|}{|W_{\max} - W_{\min}|}, \tag{2}$$

$$m = \frac{k^2}{Imp_{\min}}. \tag{3}$$

Summarizing, SARC provides higher protection for low congestion window values or flow sending rates. This reduces the probability of packet losses due to link errors on the wireless channel, which is a well-known reason for TCP performance degradation [122].

5.3 Performance Evaluation

Simulation Scenario

The proposed scheme is evaluated in the context of an UMTS/HSDPA cellular network. Network Simulator 2 (NS-2) [107] is used to perform experiments, with the additional Enhanced UMTS Radio Access Network Extensions (EURANE) module [128] for HSDPA implementation.

Fig. 24 illustrates the reference scenario and the main parameters employed in the experiments. All considered flows originate from a server (the Fixed Host – FH) on the Internet and are delivered to the User Equipment (UE) located in a 3G cellular network. SARC approach is implemented between the Node-B and the UE.

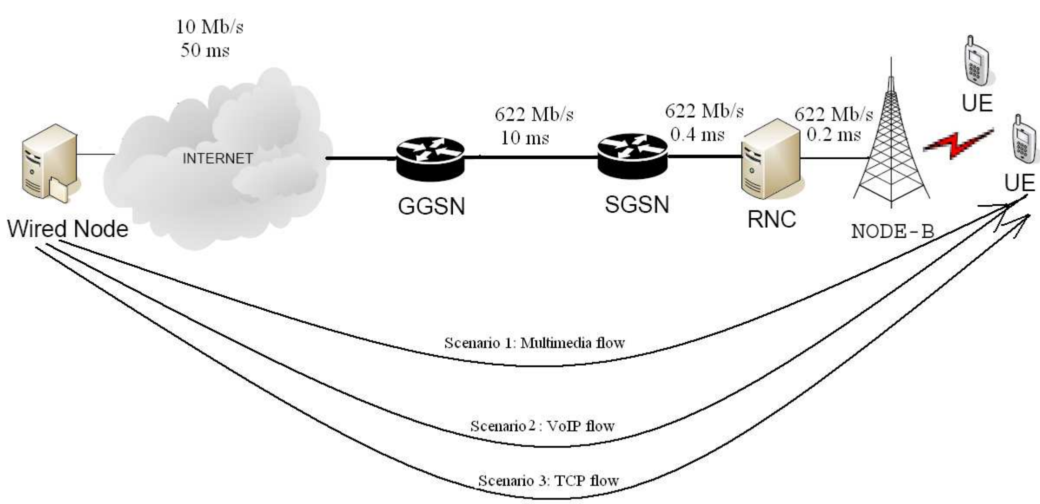


Fig. 24.Simulation scenario used for ns-2 experiments.

Video Transfer Performance

In the first scenario, the FH is a video server which transmits video streams to the video receiver located at UE. Results are presented for the “Foreman” video sequence, using MPEG-4 (open-source ffmpeg [115]) video coding. The video format is Quarter Common Intermediate Format (QCIF, 176 * 144). The GOP structure is IBBPBBPBBPBBPB. Stored in YUV format, the video clip is processed by MPEG-4 encoder which generates the encoded video stream.

The Video Sender (VS) reads encoded video stream and generates the trace file containing such information as frame type, size, etc. for each video frame. Based on this trace file the NS-2 streaming server application generates the data which is being encapsulated at all the protocol layers is sent over the network.

The effect of streaming video over the network is captured in a streaming client log file generated by NS-2. It contains such information like timestamp, size and ID for each transmitted and received packet. The trace file and the log files are used by the Evaluate Trace (ET) program to generate an output corresponding to the product of video file transmission over the error prone network. In order to examine the video quality obtained at the receiver the original video file and the one obtained after transmission over the network are compared using PSNR calculation module The integrated environment methodology was proposed and developed within the framework of EvalVid [113], enhanced as in [114] for including NS-2.

The portion of the multimedia stream which is crucial to the overall quality is retransmitted by SARC with a higher HARQ strength, i.e. packets belonging to an I-frame are retransmitted with HARQ Strength = 8, while packets belonging to B-frames are retransmitted with a HARQ strength = 2. P-frames are retransmitted with a variable HARQ strength ranging from 8 to 3 depending on the position of the frame in the GOP. Default value of HARQ strength is set to 4 for all packets in the legacy scenario (i.e. without SARC).

Achieved results are illustrated in Fig. 25, where SARC increases the range of packet error tolerance to 10^{-2}-10^{-1}. The detailed behavior of the proposed scheme with respect to the legacy approach is described in Fig. 26 where it is possible to clearly identify the unequal and dynamic protection implemented by SARC on I- and P-frames.

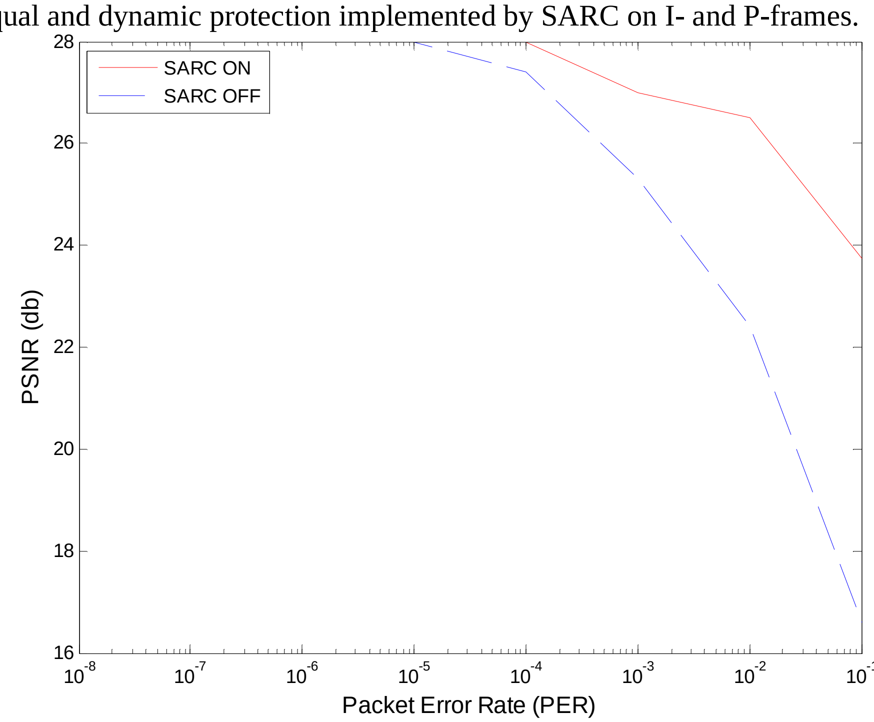


Fig. 25.Quality of “Foreman” video clip for different error rates.

VoIP Transfer Performance

Experiments on VoIP flows are performed using the simulation model presented in [125]. The sender and the receiver side are separately modeled. The sender includes: a customizable codec, which generates generic speech frames (the latter being either voice samples of voice frames, depending on the codec); a multiplexer, which aggregates several speech frames into one payload. The most common codecs employed in network simulation (e.g. G.711, GSM.AMR) are supported by the VoIPSender, while others can be easily added.

Initially, equal HARQ strength equal to 3 is associated to all transmitted speech frames. However, in case the receiver detects frame losses after out-of-order frame reception, it increases HARQ strength linearly for the subsequent packets of the stream (with HARQ strength max equal to 8) in order to avoid bulk frame losses. Once no loss is detected, SARC decreases the HARQ strength to the initial value.

Achieved results (Figs. 27 and 28) demonstrate that SARC is able to provide a relevant improvement in terms of MOS both for G.711 and GSM AMR speech flows. In average, application of SARC scheme enables the codec to deliver the same speech quality for error rate of 5% higher if compared with the case when SARC is not enabled.

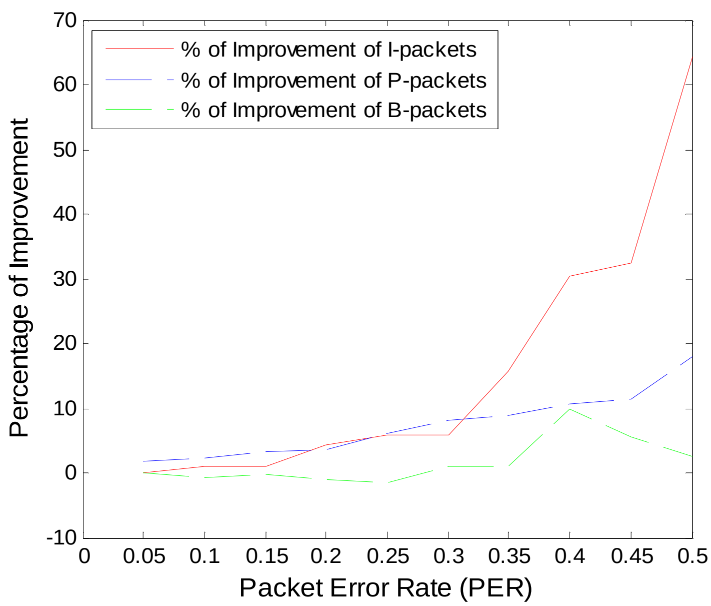


Fig. 26.Percentage of improvement in correctly delivered I-, P-, and B-packets against PER.

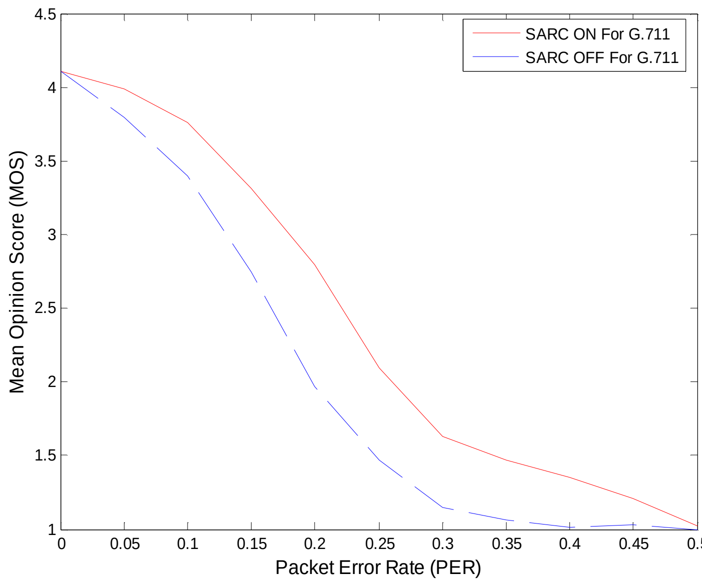


Fig. 27.G.711 Voice MOS against Packet Error Rate.

File Transfer Peformance

In this scenario, TCP sender located at FH connects the receiver implemented at UE. For the entire duration of the flow the receiver maintains up-to-date value of the congestion window (cwnd) computed by counting the number of packets received for the last RTT. Whenever the loss detection signal (three duplicate acknowledgements) is sent to the sender packet importance is increased according the function presented in Section II causing higher strength of HARQ process and, as a result, producing higher resistance to the link errors.

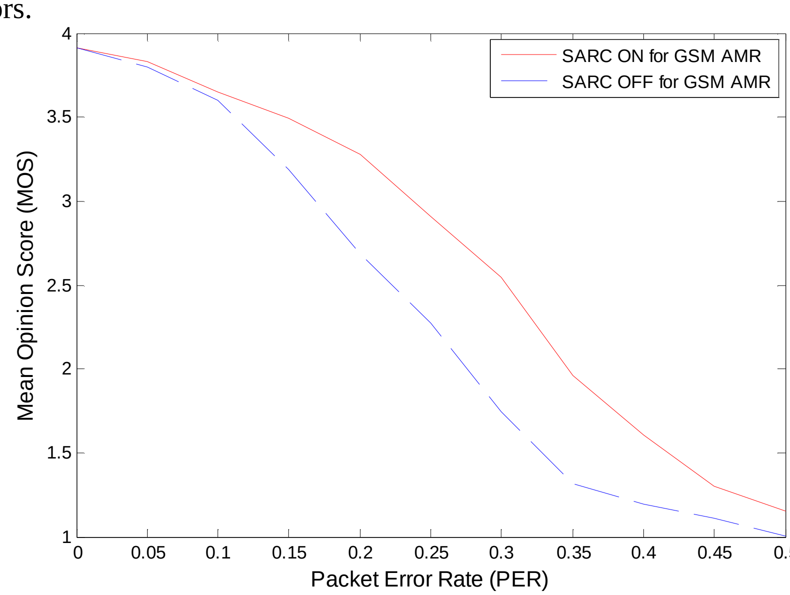


Fig. 28GSM AMR MOS against Packet Error Rate.

Figure 5.10 presents TCP throughput achieved by the flows for different PERs of the wireless link. As expected, higher protection against the link errors for low congestion values of the congestion window brings evident performance improvement and underlines advantages of dynamic error protection techniques based on application awareness introduced by SARC.

Figure 5.11 analyzes a scenario where both video and data flows are delivered on the wireless link. In this scenario, two UEs are considered: one is receiving a video stream, while the other is receiving data via FTP. The same parameters are used as in the previous scenarios. It is possible to observe that while video performance remains as in Fig. 5, data transfer is affected by relatively lower protection – while still achieving better results than in the legacy scenario (without SARC).

5.4 Conclusions and Future Work

This chapter proposes a cross-layer method between application/transport layers on a mobile terminal and link layer at the wireless base station to enable dynamic control on the level of per-packet HARQ protection. The level of protection is dynamically adapted on a per-packet basis and depends on the perceptual importance of different packets as well as on the reception history of the flow. Experimental results demonstrate the potential benefits deriving from the proposed strategy, underlining relevant improvements either for audio and video flows as well as for TCP-based data transfers.

Future work will be aimed at validating and optimizing the proposed scheme in the framework of embedded multimedia streams.

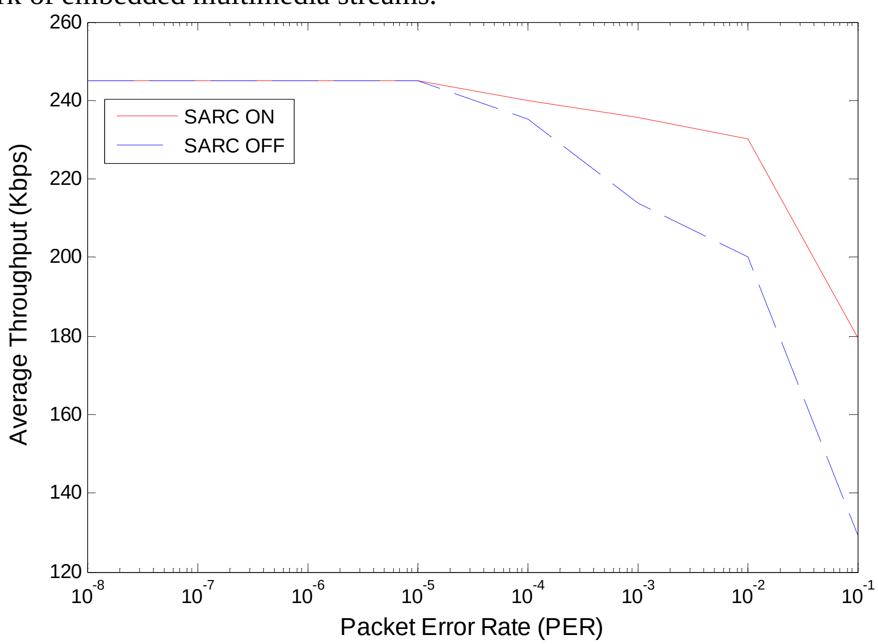


Fig. 29.Average TCP Throughput as a function of Packet Error Rate on the wireless channel.

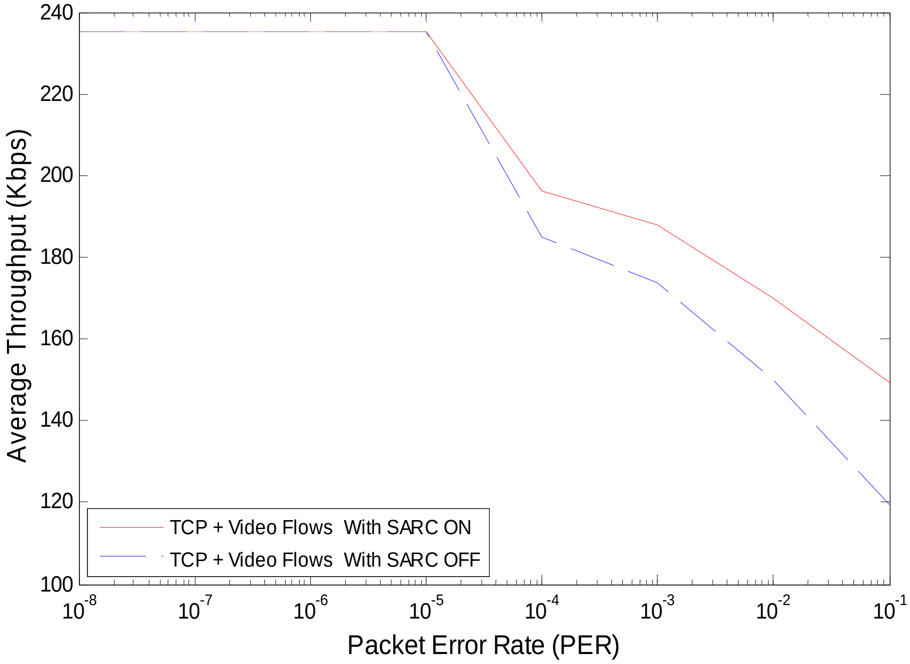


Fig. 30.Average TCP Throughput against Packet Error Rate in presence of video flows.

If we knew what it was we were doing, it would not be called research, would it?

-Albert Einstein

CHAPTER 6

6. Conclusions and Future Work

In this research, we investigate the application of service-driven cross-layer optimizations for wireless systems..

The TCP/IP protocol suite, which is the de facto standard for communications in Internet today, is originally designed for traditional wired networks. As a result, TCP/IP shows poor performance in wireless network environment due to the limitations of wireless medium terms of bandwidth, latency, information loss, and mobility.

Traditionally, the proposals for TCP/IP performance improvement optimize a single layer at a time. However, in this thesis we show that Cross-Layering allows better performance optimization and more flexibility in the design. It overcomes layering principles employed in network architectures and protocol stacks allowing joint interlayer optimization.

This thesis proposes novel scheme to enable per-packet differentiation of link layer protection driven by requirements of the end applications as well as of communication protocols implemented on the mobile terminal.

It proposes also a novel approach for cross-layer error control optimization in Wi-Fi networks.

BIOGRAPHY

Nadhir BEN HALIMA received his computer science engineering degree from the National School of Computer Sciences (ENSI), Mannouba, in 2005, and the M.S degree in communication networks engineering from SantAnna School of Advanced Studies, Pisa, Italy in 2006. Since November 2006, he is a PhD student at the Department of Information and Communication Technology of the University of Trento. In 2009 he was a visiting researcher at the Department of Electrical and Computer Engineering at North Carolina State University. His main research interest lies in field of wireless networking with a focus on performance optimization, cross-layer design and service oriented networks.

Publications published during PhD study and work toward this research thesis:

D. Kliazovich, N. Ben Halima, and F. Granelli, "Cross-Layer Error Recovery Optimization in Wi-Fi Networks," Tyrrhenian International Workshop on Digital Communication (TIWDC), Ischia island, Naples, Italy, September, 2007.

D. Kliazovich, N. Ben Halima, and F. Granelli, "Context-aware Receiver-driven Retransmission Control in Wireless Local Area Networks," IEEE International Conference on Communications (ICC), Dresden, Germany, June 2009.

N. Ben Halima, D. Kliazovich and F. Granelli, "Application-Aware Dynamic Retransmission Control in Mobile Cellular Networks", MOBILIGHT 2009, Athens, Greece, May 2009 (Also an invited journal paper)

BIBLIOGRAPHY

[1]. J. B. Postel, “Transmission Control Protocol,” RFC 793, September 1981.

[2] V. Srivastavaand and M. Motani, “Cross-Layer Design: A Survey and the Road Ahead” IEEE Communications Magazine December 2005

[3] V. Jacobson, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review, v.25 n.1, p.157-187, Jan. 1995

[4] Balakrishnan, H., V. N. Padmanabhan, S. Seshan, and R. H. Katz. 1996. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. In Proc. of the ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM) 96, Stanford, CA,

[5] Balakrishnan, H., V. Padmanabhan, S. Seshan, and R. H. Katz. 1997. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking* 5, no. 6

[6] Fahmy, S., V. Prabhakar, S. R. Avasarala, and O. Younis, 2003. TCP over Wireless Links: Mechanisms and Implications. Technical report, Purdue University.

[7] Natani, A., J. Jakilnki, M. Mohsin, and V. Sharma. 2001. TCP for Wireless Networks. Technical report. University of Texas at Dallas, November.

[8] Balakrishnan, H., S. Seshan, and R. H. Katz. 1995. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks,* 1, no. 4

[9] Balakrishnan, H., S. Seshan, E. Amir, and R. H. Katz. 1995. Improving TCP/IP Performance over Wireless Networks. In Proc. of 1st ACM Conference on Mobile Computing and Networking (ACM Conference on Mobile Computing and Netuoring (MOBICOM 95), Berkeley, California

[10] Balakrishnan, H., S. Seshan, and R. H. Katz. 1995. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks,* 1, no. 4

[11] Parsa, C., and J. J. Garcia-Luna-Aceves. 1999. TULIP: A Link-Level Protocol for Improving TCP over wireless Links. In Proc.IEEE Wireless Communications and Networking Conference (WCNC 99), New Orleans, Louisiana

[12] Vaidya, N. H., M. Mehta, C. Perkins, and G. Montenegro. 1997. Delayed Duplicate Acknowledgements: A TCP-Unaware Approach to Improve Performance of TCP over Wireless Links. Technical Report, Texas A&M University

[13] Gerla, M., K. Tang, and R. Bagrodia. 1999. TCP Performance in Wireless Multihop Networks. In Proc. of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA

[14] Ayanoglu, E., S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. 1995. AIRMAIL: A Link-Layer Protocol for Wireless Networks. *ACM Wireless Networks,* no. 1

[15] Bakre, A., and B. R. Badrinath. 1995. Handoff and System Support for Indirect TCP/IP. Paper presented at 2nd the USENIX Symposium on Mobile and Location-Independent Computing. Ann Arbor, MI

[16] Badrinath, B. R., and A. Bakre. 1995. I-TCP: Indirect TCP for Mobile Hosts. Paper presented at the 15th International Congerence on Distributed Computing Systems, Vancouver, Canada

[17] Brown, K., and S. Singh. 1996. M-TCP: TCP for Mobile Cellular Networks. In Proc. of IEEE Conference on Computer Communication (INFOCOM) 96, San Francisco.

[18] Brown, K., and S. Singh. 1997. M-TCP: TCP for Mobile Cellular Networks. *ACM Computer Communication Review,* 27, no. 5

[19] Wang, K. Y., and S. K. Tripathi. 1998. Mobile-End Transport Protocol: An Alternative to TCP/IP over Wireless Links. In Proceedings of IEEE Conference on Computer Communication (INFOCOM) 98, San Francisco,California

[20] Floyd, S., J. Mahdavi, M. Mathis, and M. Podolsky. 2000. An Extension to the Selective Acknowledgement (SACK) Option for TCP. RFC 2883, July.

[21] Mathis, M., J. Mahdavi, S. Floyd, and A. Romanow. 1996. Selective cknowledgment option, RFC-2018, October.

[22] Allman, M., S. Floyd, and C. Partridge. 2002. Increasing TCPs Initial Window. RFC 3390, October.

[23] Balakrishnan, H., V. Padmanabhan, S. Seshan, M. Stemm, and R. H. Katz. 1998. TCP Behavior of a Busy Internet Server: Analysis and Improvements. In Proc. of IEEE Conference on Computer Communication (INFOCOM) 98, San Francisco

[24] Mathis, M., and J. Mahdavi. 1996. Forward Acknowledgement: Refining TCP Congestion Control. In Proc. of the ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM), Stanford, CA

[25] Hassan, M., and R. Jain. 2004. *High Performance TCP/IP Networking*. Upper Saddle River, NJ: Prentice Hall.

[26] Keshav, S., and S. P. Morgan. 1997. SMART Retransmission: Performance with Random Losses and Overload. In Proc. of IEEE Conference on Computer Communication (INFOCOM)97, Kobe, Japan

[27]Ludwig, R., and R. H. Katz. 2000. The Eifel Algorithm: Making TCP Robust against Spurious Re-Transmission. *ACM Computer Communication Review* 30, no. 1

[28]Ludwig, R., and M. Meyer. 2002. The Eiffel Detection Algorithm for TCP. Internet Draft, IETF, February.

[29] Jacobson, V., R. Braden, and D. Borman. 1992. TCP Extensions for High Performance. RFC-1323, May.

[30]Floyd, S. 1994. TCP and Explicit Congestion Notification. *ACM Computer Communication Review* 24

[31] Ramakrishnan, K., and S. Floyd. 1999. A Proposal to Add Explicit Congestion Notification (ECN) to IP. RFC2481, IETF, January.

[32] Ramakrishnan, K., S. Floyd, and D. Black. 2001. The Addition of Explicit Congestion Notification (ECN) to IP. RFC3168, IETF, September.

[33] Bakshi, B. S., P. Krishna, N. H. Vaidya, and D. K. Pradhan. 1996. Improving Performance of TCP over Wireless Networks. Technical Report, Texas A&M University

[34] Bakshi, B. S., P. Krishna, N. H. Vaidya, and D. K. Pradhan. 1997. Improving Performance of TCP over Wireless Networks. Paper presented at the 17th International Conference on Distributed Computing Systems,Baltimore, MD, May.

[35] Balakrishnan, H., and R. H. Katz. 1998. Explicit Loss Notification and Wireless Web Performance. Paper presented at the IEEE Globecom Internet Mini-Conference, Sydney, Australia

[36] Goel, S., and D. Sanghi. 1998. Improving TCP Performance over Wireless Links. *Proceedings of the IEEE International Conference on GlobalConnectivity in Energy, Computer, Communication and Contro*

[37]Postel, J. 1981. Internet Control Message Protocol. RFC-792, September.

[38] Braden, R. 1989. Requirements for Internet Host Communication Layers. RFC-1122, October.

[39]Samaraweera, N. K. G. 1999. Non-congestion Packet Loss Detection for TCP Error Recovery Using Wireless Links. In *Communications* 146

[40] Jain, R. 1989. A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks. *ACM ComputerCommunication Review* 19

[41] Brakmo, L., and S. Peterson. 1995. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communications*

[42]Krishnan, R., M. Allman, C. Partridge, and J. P. G. Sterbenz. 2002. Explicit Transport Error Notification for Error-Prone Wireless and Satellite Networks. BBN Technical Report BBN Technologies

[43]Krishnan, R., M. Allman, C. Partridge, J. P. G. Sterbenz, and W. Ivancic. 2002. Explicit Transport Error Notification (ETEN) for Error-Prone Wireless and Satellite Networks—Summary. Paper presented at the Earth Science Technology Conference, Pasadena, CA

[44] Krishnan, R., J. P. G. Sterbenz, W. M. Eddy, C. Partridge, and M. Allman.Explicit Transport Error Notification (ETEN) for Error-Prone Wirelessand Satellite Networks, unpublished.

[45] Biaz, S., M. Mehta, S. West, and N. H. Vaidya. 1997. TCP over Wireless Networks Using Multiple Acknowledgements. Technical Report, Texas A&M University

[46] Chan, A., D. H. K. Tsang, and S. Gupta. 1997. TCP (Transmission Control Protocol) over Wireless Links, In Proc. of the 47th IEEE Vehicular Technology Conference 97

[47]. Goff, T., J. Moronski, D. S. Phatak, and V. Gupta. 2000. Freeze-TCP: A True End-to-End Enhancement Mechanism for Mobile Environments. In Proc. Of IEEE Conference on Computer Communication (INFOCOM 2000)

[48] Tian, Y., K. Xu, and N. Ansari. 2005. TCP in Wireless Environments: Problems and Solutions, *IEEE Communications Magazine*

[49] Tsaoussidis, V., and H. Badr. 1999. TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains. Technical Report,Department of Computer Science, State University New York Stony Brook

[50] Tsaoussidis, V., and H. Badr. 2000. TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains. Paper presented at the 8th IEEE International Conference on Network Protocols

[51] Tsaoussidis, V., and I. Matta. 2002. Open Issues on TCP for Mobile Computing, *Journal of Wireless Communications and Mobile Computing*

[52] Sinha, P., N. Venkitaraman, R. Sivakumar, and V. Bharghavan. 2002. WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. Paper presented at the ACM Conference on Mobile Computing and Netuoring (MOBICOM) 99, Seattle, WA

[53] Sinha, P., T. Nandagopal, N. Venkitaraman, R. Sivakumar, and V.Bharghavan. 2002. WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. *ACM/Baltzer Wireless Networks Journal* 8

[54] Ratnam, K., and I. Matta. 1998. WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links. Paper presented at the 3rd IEEE Symposium on Computers and Communications

[55] Ratnam, K., and I. Matta. 1998. Effect of Local Retransmission at Wireless Access Points on the Round Trip Time Estimation of TCP. Paper presented at the 31st Annual Simulation Symposium

[56] Li, Y., and L. Jacob. 2003. Proactive-WTCP: An End-to-End Mechanism to Improve TCP Performance over Wireless Links. Paper presented at the 28th Annual IEEE International Conference on Local Computer Networks,

[57] Lakshman, T. V., and U. Madhow. 1997. The Performance of TCP/IP for Networks with High Bandwidth Delay Product and Random Loss. IEEE/ACM Trans. On Networking

[58] Keshav, S. 2001. Congestion Control in Computer Networks. PhD diss. University of California at Berkeley, September.

[59]Akyildiz, I. F., G. Morabito, and S. Palazzo. 2001. TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks. *IEEE/ACM Trans. On Networking* 9
[60] Akyildiz, I. F., X. Zhang, and J. Fang. 2002. TCP-Peach: Enhancement of TCP-Peach for Satellite IP Networks. *IEEE Communications Letters*
[61] Brakmo, L., S. O'Malley, and L. Peterson. 1994. TCP Vegas: New Techniques for Congestion Detection and Avoidance. In Proc. of the ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM 94), London
[62] Brakmo, L., and L. Peterson. 1995. TCP Vegas: End to End Congestion Avoidance on a Global Internet. *IEEE Journal on Selected Areas in Communication*
[63] Mo, J., R. La, V. Anantharam, and J. Walrand. 1999. Analysis and Comparison of TCP Reno and Vegas. Paper presented at the IEEE Conference on Computer Communication (INFOCOM), New York
[64] Parsa, C., and J. J. Garcia-Luna-Aceves. 1999. Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media. Paper presented at the 7th Annual IEEE International Conference on Network Protocols, Toronto, Canada, November
[65] Mascolo, S., C. Casetti, M. Gerla, S. S. Lee, and M. Sanadidi. 2000. TCP Westwood: Congestion Control with Faster Recovery. Technical Report. Department of Computer Science, University of California at Los Angeles.
[66]Casetti, C., M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. 2001. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. Paper presented at the ACM Conference on Mobile Computing and Networking (MOBICOM), Rome, Italy, July 16–21.
[67]Gerla, M., M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo. 2001. TCP Westwood: Congestion Window Control Using Bandwidth Estimation. In Proc. of IEEE Globecom 2001
[68] Zanella, A., G. Procissi, M. Gerla, and M. Y. Sanadidi. 2001. TCP Westwood: Analytic Model and Performance Evaluation. In Proc. of IEEE Globecom 01
[69] Yang, G., R. Wang, F. Wang, M. Y. Sanadidi, and M. Gerla. 2002. TCP Westwood with Bulk Repeat for Heavy Loss Environments. Technical Report. Department of Computer Science, University of California at Los Angeles.
[70] Yang, G., R. Wang, M. Y. Sanadidi, and M. Gerla. 2002. Performance of TCPW BR in Next Generation Wireless and Satellite Networks. Technical Report Department of Computer Science, University of California at Los Angeles.
[71] Fu, C. P. 2001. TCP Veno: End-to-End Congestion Control over Heterogeneous Networks. PhD diss., Chinese University of Hong Kong.
[72] Fu, C. P., and Soung C. Liew. 2003. TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks. *IEEE Journal On Selected Areas in Communications*
[73] Xu, K., Y. Tian, and N. Ansari. 2004. TCP-Jersey for Wireless IP Communications. *IEEE Journal on Selected Areas in Communications*
[74] Aggarwal, A., S. Savage, and T. Anderson. 2000. Understanding the Performance of TCP Pacing. In Proc. of IEEE Conference on Computer Communication (INFOCOM),
[75] Zhang, L., S. Shenker, and D. D. Clark. 1991. Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two Way Traffic. Paper presented at the ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM) 91 Conference on Communications Architectures and Protocols, September
[76] Aron, M., and P. Druschel. 1998. TCP Improving Startup Dynamics by Adaptive Timers and Congestion Control, Technical Report, Rice University, Houston, TX.
[77] Padmanabhan, V. N., and R. H. Katz. 1998. TCP Fast Start: A Technique for Speeding Up Web Transfers. Paper presented at the IEEE Globecom 98 Internet Mini-Conference, Sydney, Australia, November.

[78] Hoe, J. 1995. Start-Up Dynamics of TCPs Congestion Control and Avoidance Schemes. Master’s thesis, MIT, Cambridge, MA, June.
[79] Mathis, M., J. Semke, J. Madhavi, and K. Lahey. 1999. The Rate-Halving Algorithm for TCP Congestion Control. Internet Draft, July.
[80] Partridge, C. 1998. ACK Spacing for High Bandwidth-Delay Paths with Insufficient Buffering. Intemet Draft, draft-rfced-infopartridge-01.txt, September.
[81] Zhang, C., and V. Tsaoussidis. 2001. TCP Real: Improving Real-Time Capabilities of TCP over heterogeneous Networks. Pesented at the 11th IEEE/ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), New York.
[82] Zhang, C., and V. Tsaoussidis. 2002. TCP-Real: Receiver-Oriented Congestion Control. *Computer Networks*
[83] Tsaoussidis, V., H. Badr, and R. Verma. 1999. Wave & Wait Protocol (WWP) An Energy-Saving Transport Protocol for Mobile IP-Devices. Paper presented at the 7th IEEE International Conference on Network Protocols.
[84]. Tsaoussidis, V., A. Lahanas, and H. Badr. 2000. The Wave and Wait Protocol: High Throughput and Low Energy Expenditure for Mobile-IP Devices. Paper presented at the 8th IEEE Conference on Networks, Singapore.
[85] Tsaoussidis, V., A. Lahanas, and C. Zhang. 2001. The Wave and Probe Communication Mechanisms. *Journal of Supercomputing*
[86]. Liu, J., and S. Singh. 2001. ATCP: TCP for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*
[87] V. Srivastava and M. Motani, “Cross-layer design: a survey and the road ahead,” IEEE Communications Magazine, vol. 43, no. 12, pp. 112 – 119, December 2005.
[88] Z. H. Haas, “Design Methodologies for Adaptive and Mul-timedia Networks,” *Guest Editorial, IEEE Communications Magazine*, vol. 39, no. 11, pp. 106-107, November 2001.
[89] Q. Wang and M.A. Abu-Rgheff, “Cross-layer signalling for next-generation wireless systems,” in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, March 2003.
[90] V. T. Raisinghani and S. Iyer, “Cross-layer design optimiza-tions in wireless protocol stacks,” *Computer Communica-tions*, v. 27, n. 8, pp. 720 – 724, 2004.
[91] M. Chinta and S. Helal, “ILC-TCP: An Interlayer Collabora-tion Protocol for TCP Performance Improvement in Mobile and Wireless Environments,” in *Proc. of the third IEEE Wire-less Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, March 2003.
[92] M. Chinta and S. Helal, “ILC-TCP: An Interlayer Collabora-tion Protocol for TCP Performance Improvement in Mobile and Wireless Environments,” in *Proc. of the third IEEE Wire-less Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, March 2003.
[93]V.T. Raisinghani, A.K. Singh, and S. Iyer, “Improving TCP performance over mobile wireless environments using cross layer feedback,” in *Proc. of the IEEE Internationa Confer-ence on Personal Wireless Communications*, December 2002.
[94] T. Holliday and A. Goldsmith, “Optimal Power Control and Source Channel Coding for Delay Constrained Traffic Wireless Channels,” *IEEE ICC 2003*, Anchorage, AK, May 11–15, 2003.
[95] J. Gross *et al.*, “Cross-Layer Optimization of OFDM Transmission Systems for MPEG-4 Video Streaming,” *Comp. Commun.*, vol. 27, 2004, pp. 1044–55.
[96] R. S. Tupelly, J. Zhang, and E. K. P. Chong, “Opportunistic Scheduling for Streaming Video in Wireless Networks,” *37th Annual Conf. Info. Sci. and Sys.*, Baltimore, MD, Mar. 12–14, 2003.

[97] Q. Liu, S. Zhou, and G. Giannakis, "Cross-layer Scheduling with Prescribed QoS Guarantees in Adaptive Wireless Networks," *IEEE JSAC*, vol. 23, no. 5, May 2005, pp. 1056–66.
[98] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks," *IEEE Int'l. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 26–29, 2002.
[99] M. van der Schaar et al., "Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission over 802.11 WLANs," *IEEE JSAC*, vol. 21, no. 10, Dec. 2003, pp. 1752–63.
[100] M. van der Schaar and S. Shankar, "Cross-Layer Wireless Multimedia Transmission: Challenges, Principles, and New Paradigms," *IEEE Wireless Commun.*, vol. 12, no. 4, Aug. 2005, pp. 50–58.
[101] Y. Peng et al., "Adaptive Resource Allocation and Frame Scheduling for Wireless Multi-user Video Streaming," *IEEE Int'l. Conf. Image Proc.*, Genova, Italy, Sept. 11–14, 2005.
[102] D. Kliazovich, P. E. Gomes, F. Granelli1, and Nelson L. S. da Fonseca, "Receiver-Driven Queue Management For Achieving RTT-fairness in Wi-Fi Networks," IEEE Global Communications Conference (GLOBECOM), Honolulu, Hawaii, December 2009.
[103]W. C. Y. Lee, "Mobile Communications Design Fundamentals," 2nd Edition, John Wiley and Sons,1993.
[104] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S.C. Diot, "Packet-level traffic asurements from the Sprint IP backbone," IEEE Network, vol. 17, no. 6, pp. 6 – 16, Nov.-Dec. 2003.
[105]. A. Partow, "General Purpose Hash Function Algorithms," http://www.partow.net/ programming/hashfunctions/
[106] D. Kliazovich and F. Granelli, "A Cross-layer scheme for the TCP Performance Improvement in Wireless LANs," IEEE Global Communications Conference, GLOBECOM'04, Dallas, U.S.A,
December 2004.
[107] NS-2 simulator tool home page, http://www.isi.edu/nsnam/ns/
[108]F. Kelly, "Mathematical Modelling of the Internet," B. Engquist and W. Schmid (ed.), "Mathematics Unlimited - 2001 and Beyond", Springer-Verlag, Berlin, pp. 685-702, 2001.
[109]Y. -D. Yao, "An effective go-back-N ARQ scheme for variable-error-rate channels," IEEE Trans. Commun., vol. 43, no. 1, pp. 20-23, Jan. 1995.

[110]H. Minn, M. Zeng, and V. K. Bhargava, "On ARQ scheme with adaptive error control," IEEE Transactions on Vehicular Technology, vol. 50, no. 6, pp. 1426 – 1436, Nov. 2001.

[111]Kurtis J. Guth and Tri T. Ha, "An adaptive stop-and-wait ARQ strategy for mobile data communications," Computer Communications archive, vol. 14, no. 7, pp. 405 – 412, 1991.

[112] C. Hoene, B. Rathke, and A. Wolisz, "On the Importance of a VoIP Packet", In Proc. of ISCA Tutorial and Research Workshop on th Auditory Quality of Systems, Herne, Germany, April 2003.

[113]J. Klaue, B. Rathke and A. Wolisz, "EvalVid – A Framework for Video Transmission and Quality Evaluation", Proceedings of the 13th International Conference on Modeling, Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, 2003. http://www.tkn.tu-berlin.de/research/evalvid

[114]Chih-Heng Ke, Cheng-Han Lin, Ce-Kuen Shieh, Wen-Shyang Hwang, “A Novel Realistic Simulation Tool for Video Transmission over Wireless Network,” The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Com-puting (SUTC2006), June 5-7, 2006, Taichung, Taiwan. Code available on line in: http://140.116.72.80/~smallko/ns2/Evalvid_in_NS2.htm

[115]ffmpeg multimedia system, http://ffmpeg.mplayerhq.hu/

[116] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, “Robust Internet video transmission based on scalable coding and unequal error protection,” Image Communication, vol. 15, no. 1-2, pp. 77-94, Sept. 1999.

[117] Z. Wu, A. Bilgin and M.W. Marcellin, “Unequal error protection for transmission of JPEG2000 codestreams over noisy channels,” ICIP, 2002.

[118] M. van der Schaar and H. Radha, “Unequal packet loss resilience for Fine-Granular-Scalability video,” IEEE Trans. on Multimedia, vol. 3, pp. 381–394, Dec. 2001.

[119] X. K. Yang, C. Zhu, Z. G. Li, G. N. Feng, S. Wu, and N. Ling, “A degressive error protection algorithm for MPEG-4 FGS video streaming,” Proc. of IEEE ICIP, Rochester, USA, Sept. 2002.

[120] Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Network-adaptive scalable video streaming over 3G wireless network,” IEEE ICIP 2001, Thessaloniki, Greece.

[121] F. Granelli, D. Kliazovich and N. L. S. da Fonseca, "Performance Limitations of IEEE 802.11 Networks and Potential Enhancements," Yang Xiao and Yi Pan (ed), "Wireless LANs and Bluetooth", Nova Science Publishers, Hardbound, 2005.

[122] Haejung Lim, Kaixin Xu, and M. Gerla, “TCP performance over multipath routing in mobile ad hoc networks,” IEEE International Conference on Communications, vol. 2, pp. 1064 – 1068, May 2003.

[123] D. Chase, “Code combining: A Maximum-Likelihood Decoding App. for Comb. an Arbitrary Number of Noisy Packets,” *IEEE Trans. on Comm.*, May 1985.

[124] C. Hoene, B. Rathke, and A. Wolisz, "On the Importance of a VoIP Packet", In Proc. of ISCA Tutorial and Research Workshop on th Auditory Quality of Systems, Herne, Germany, April 2003.

[125] *A. Bacioccola, C. Cicconetti, G. Stea,* “User-level Performance Evaluation of VoIP Using ns-2,” Workshop on Network Simulation Tools (NSTools), Nantes, France, October 2007.

[126] P. J. A. Gutíerrez, “*Packet Scheduling and Quality of Service in HSDPA*,” Ph .D. dissertation, Aalborg University, October 2003.

[127] VLC media player home page, http://www.videolan.org/vlc/

[128] Enhanced UMTS Radio Access Extensions http://www.ti-wmc.nl/eurane/

[128] P. Chou and Z. Miao, "Rate-distortion Optimized Streaming of Packetized Media", IEEE Transactions on Multimedia, vol.8, no.2, pp. 390-404, April, 2006.

[129] G. Cheung, W.-t. Tan, T. Yoshimura, "Rate-distortion Optimized Application-level Retransmission Using Streaming Agent for Video Streaming over 3G Wireless Network", IEEE ICIP, September, 2002.

[130] F. Yang, Q. Zhang, W. Zhu, Y.-Q. Zhang, "Bit Allocation for Scalable Video Streaming over Mobile Wireless Internet," IEEE Infocom, 2004.

[131] H. Balakrishnan, V. Padmanabhan, S. Sshan and R. Katz, "A comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Trans. Networking, vol.5, no.6, December, 1997.

[132] G. Liebl, H. Jenkac, T. Stockhammer, C. Buchner, "Radio Link Buffer Management and Scheduling for Video Streaming over Wireless Shared Channels", International Packet Video Workshop, December, 2004.

[133] Q. Zhang, W. Zhu, and Y-Q. Zhang, "QoS-adaptive resource allocation for video transmission over 3G wireless network", IEEE Trans. on Circuit and System for Video Technology, Volume 14, Issue 8, Aug. 2004 Page(s):1049 - 1063.