Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 831707, 17 pages
doi:10.1155/2010/831707

Research Article

QoS Modeling for End-to-End Performance Evaluation over Networks with Wireless Access

Gerardo Gómez, Javier Poncela González, Mari Carmen Aguayo-Torres, and José Tomás Entrambasaguas Muñoz

Department of Communications Engineering, University of Malaga, 29071 Malaga, Spain

Correspondence should be addressed to Gerardo Gómez, ggomez@ic.uma.es

Received 31 July 2009; Revised 13 January 2010; Accepted 31 January 2010

Academic Editor: Máirtín O'Droma

This paper presents an end-to-end Quality of Service (QoS) model for assessing the performance of data services over networks with wireless access. The proposed model deals with performance degradation across protocol layers using a bottom-up strategy, starting with the physical layer and moving on up to the application layer. This approach makes it possible to analytically assess performance at different layers, thereby facilitating a possible end-to-end optimization process. As a representative case, a scenario where a set of mobile terminals connected to a streaming server through an IP access node has been studied. UDP, TCP, and the new TCP-Friendly Rate Control (TFRC) protocols were analyzed at the transport layer. The radio interface consisted of a variable-rate multiuser and multichannel subsystem, including retransmissions and adaptive modulation and coding. The proposed analytical QoS model was validated on a real-time emulator of an end-to-end network with wireless access and proved to be very useful for the purposes of service performance estimation and optimization.

1. Introduction

Quality of Service (QoS) over networks with wireless access is a common research topic and is often studied in relation to end-to-end QoS or cross-layer architectures. Most authors focus on particular network elements or domains (e.g., terminals, radio interfaces, or core networks) or on specific protocol layers, such as congestion control schemes for wireless multimedia at the transport layer (TCP-friendly) [1] or QoS-scheduling techniques at the radio interface [2].

However, the QoS perceived by end users is an end-to-end issue and is therefore affected by every part of the network, the protocol layers, and the way they all interact. Moreover, seamless connectivity requires wireless and wired networks to operate in a coordinated manner in order to support packet data services with different QoS requirements. In such scenarios, data service performance assessment is usually addressed through active terminal monitoring over real networks [3]. However, such a method proves to be costly if the operator wants to collect statistics from a reasonable number of terminals, applications, and locations. It may also prove to be a highly time-consuming process due to the variety of potential scenarios, both in terms of the type of service being offered and their spatial location.

Only a small number of works in the literature describe a general framework for end-to-end QoS control. One such end-to-end QoS framework for streaming services in 3G mobile networks is considered in [4], analyzing the interaction between UMTS and IETF's protocols and mechanisms. In [5], several key elements in the end-to-end QoS support for video delivery are addressed, including network QoS provisioning and scalable video representation. A small number of works have begun to include proposals involving end-to-end QoS management over wireless networks. In [6], a theoretical model for integrated cross-layer control and optimization in wireless multimedia communications is introduced. The work presented in [7] proposes an adaptive protocol suite for optimizing service performance over wireless networks, including rate adaptation, congestion control, mobility support, and coding. An overview of the current cross-layer solutions for QoS support in multihop wireless networks including cooperative communication and networking or opportunistic transmission can be found in [8]. However, none of the previous works presents a method or tool for assessing and/or optimizing end-to-end QoS in a simple manner.

In this paper, the problem of providing accurate end-to-end performance estimations over networks with wireless access is addressed through a QoS model. The quality of packet data services is analyzed by calculating the performance degradation that occurs at each protocol layer. The overall degradation is analyzed starting from the physical layer up to the application layer. The performance assessment model described herein can be used to estimate the end-to-end performance of services in this type of networks before deployment. In addition, the proposed model is a useful tool for achieving end-to-end optimization, as it helps to find an appropriate configuration for each layer, thereby optimizing the end-to-end performance.

The proposed model was validated using a set of mobile terminals which were connected to a streaming server through an IP network with wireless access. We paid special attention to the impact of different radio interface mechanisms and transport layer protocols on streaming service performance.

The remainder of this paper is organized as follows. The general system model for multimedia streaming services over the wired-wireless network is outlined in Section 2. The QoS modeling process of the streaming protocol stack is presented in Section 3. Section 4 presents the end-to-end model results, whereas their validation results from a real-time emulator are shown in Section 5. Section 6 discusses the applicability of the proposed architecture for assessing the Quality of Experience (QoE) for data service users. Finally, Section 7 states the main conclusions of this work.

2. System Model

This section presents the scenario and protocol stack under analysis. As mentioned earlier, a streaming service was chosen as the representative case to be studied (see Figure 1). The system is divided into two subsystems: the radio access network segment and the transport network segment. An access node is responsible for interconnecting the two segments in order to provide an end-to-end connection between the User Equipment (UE) and streaming server.

Across the protocol stack, Packet Data Units (PDUs) of Layer i (Li) will hereinafter be referred to as Li-PDUs. The size of the PDUs at each layer is denoted by B_{Li} and the Li-PDU header length is denoted by H_{Li}. The following terminology is used for performance indicators.

(a) S_{Li} is the mean information rate offered to layer i.

(b) R_{Li} is the mean net throughput achieved at layer i (at the receiver).

(c) D_{Li} is the mean Li-PDU delay.

(d) P_{Li} is the mean Li-PDU loss rate.

A description of the system model is given from Layer 1 (L1) to Layer 5 (L5).

($L1$) A variable-rate multiuser and multichannel subsystem is considered for the radio interface. Channel multiplexing is performed at the PHYsical (PHY) layer, where the radio channel is divided into resources independently allocated to users. Also, the PHY layer performs adaptive modulation and channel coding [9].

($L2$) The link layer is responsible for performing user multiplexing; that is, resources are temporarily assigned to users following a specific scheduling algorithm. Moreover, selective retransmissions of erroneous $L2$-PDUs (if so configured) and the compression of upper layer headers are also performed at this layer. Traffic shaping is performed at the upper interface of the network side L2; when the network load is high, data may be lost due to overflow in the queue.

($L3$) An IP-based radio access node is considered at the network layer (L3), through which mobile terminals connect to the streaming server.

($L4$) At the transport layer (L4), several options were analyzed at the user plane (UDP, TCP, and TFRC [1]).

($L5$) At the user plane, the Real-time Transport Protocol (RTP) carries delay-sensitive data while the Real-time Transport Control Protocol (RTCP) conveys information on the participants and monitors the quality of the RTP session. Performance analysis of streaming signaling protocols during session setup is out of the scope of this paper; however, further details can be found in [1, 5].

In this work, throughput, delay, and loss rate indicators at each layer are modeled analytically, except the delay associated to scheduling algorithms at the radio and IP domains, which is still an open issue and has been obtained from simulations.

For the traffic model, variable rate information sources are considered at the application layer. A sufficiently large application buffer is assumed; thus network jitter is compensated at this layer. A summary of the numerical parameters used in this work at all layers is given in Table 12 at the end of the paper.

3. Protocol Layer Modeling

3.1. Physical Layer Model. The physical radio resources considered in this work are based on an Orthogonal Frequency Division Multiple Access (OFDMA) scheme, as defined for 3 GPP Long-Term Evolution (LTE) [10, 11]. OFDM subcarriers are organized into N_c channels, each of which groups M_c subcarriers together that can be reallocated to users on a frame-by-frame basis. A frame is a set of M_T OFDM symbols with a duration of TTI (Time Transmission Interval). The resource allocation unit (M_c subcarriers during a TTI) is referred to as a Physical Resource Block (PRB) and allows for the transmission of $M_c \cdot M_T$ Quadrature Amplitude Modulation (QAM) symbols, as shown in Figure 2.

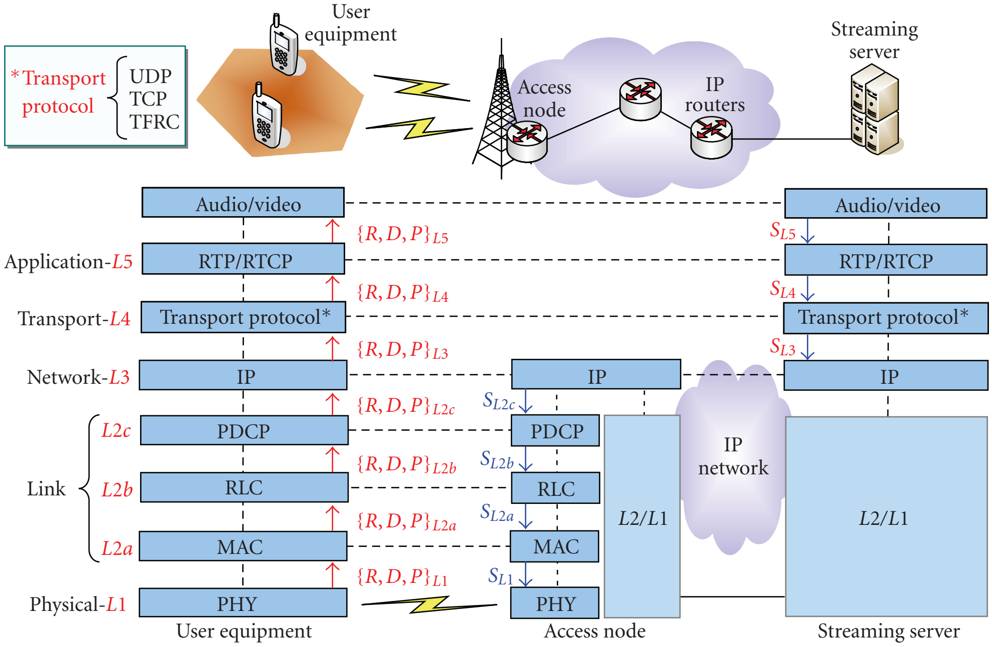


FIGURE 1: Scenario and protocol stack under analysis.

TABLE 1: Parameters associated to the physical layer model.

	Parameter	Description
Physical layer	BER_T	Target Bit Error Rate (BER)
	R_k	Constellation set (modulation levels)
	TTI	Transmission Time Interval
	N_c	Number of channels
	ρ_c	Correlation between consecutive channels
	T_B	Frame duration
	M_c	Number of QAM symbols multiplexed on a channel
	M_T	Number of QAM symbols per TTI
	C	Coding Rate
	G_{cod}	Channel Coding Gain
Radio Channel	f_D	Doppler spread
	$\overline{\gamma}$	Average Signal to Noise Ratio (SNR)

Adaptive modulation is used to follow the fading behavior of the channels represented by its instantaneous Signal to Noise Ratio (SNR); such behavior is different for each user and PRB [12]. Let $\gamma_{i,k}[n]$ be a matrix representing the received instantaneous SNR for user i and channel k at frame n, and let $m_{i,k}[n]$ be the number of bits/symbol of a QAM constellation that should ideally fulfill a certain target bit error rate (BER_T). Channel coding (with coding rate C) is used to obtain a certain coding gain G_{cod} that generally ranges from 2 to 10 dB.

The same constellation $m_{i,k}[n] = f(\gamma_{i,k}[n], \mathrm{BER}_T, G_{\mathrm{cod}})$ is used for all QAM symbols within a PRB, making it possible to transmit a total of $r_{i,k}[n] = M_c \cdot M_T \cdot C \cdot m_{i,k}[n]$ bits. The term $r_{i,k}[n]$ can be seen as the potential rate (in bits/frame) of channel k if it is assigned to user i (see MAC layer model in the following section). The actual rate of a channel will be $R_k[n] = r_{\hat{\imath},k}[n]$, where $\hat{\imath}$ represents the user who is actually allocated to channel k.

Regarding the radio channel's behavior represented by the random process $\gamma_{i,k}[n]$, its temporal variation (fading) is assumed to follow the usual Jakes' model [9]; an exponential decay model with factor ρ_c is assumed for correlation between channels k; independence is assumed between users i.

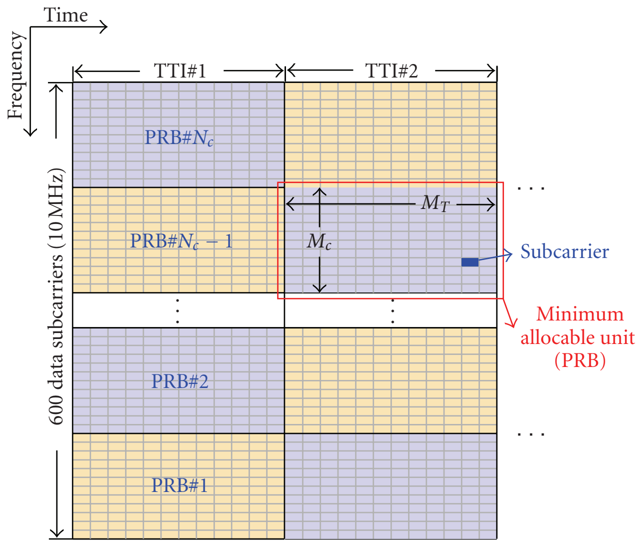


Figure 2: LTE physical resources structure.

The following expressions together with the parameters listed in Table 1 provide a summary of the performance indicators at the physical layer.

PHY Model is

$$P_{L1} \le \mathrm{BER}_T, \tag{1}$$

$$r_{i,k}[n] = M_c \cdot M_T \cdot C \cdot m_{i,k}[n], \quad m_{i,k}[n] = f\left(\gamma_{i,k}[n], \mathrm{BER}_T, G_{\mathrm{cod}}\right), \tag{2}$$

$$D_{L1} \approx \mathrm{TTI}. \tag{3}$$

3.2. Link Layer Model. The link layer includes the Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP) sublayers (as shown in Figure 1).

3.2.1. MAC Layer Model. A set of N_u users share the radio transmission resources. The MAC layer at the access node allocates channels to users on a frame-by-frame basis; that is, for each new frame, the system assigns each physical channel to a single user. OFDMA allocation is applied according to a particular scheduling algorithm, considering different PRBs with adaptive modulation per user. The actual number of bits extracted from the ith user queue and allocated on channel k, denoted by $R_{i,k}[n]$, will be zero or $r_{i,k}[n]$, depending on the user scheduler decision. The total number of bits extracted from the ith user queue at frame n is given by.

$$R_{L2a}\{\text{user } i\}[n] = \sum_{k=1}^{N_c} R_{i,k}[n]. \tag{4}$$

Two scheduling algorithms were assessed: *Round Robin* (RR) and *Modified Largest Weighted Delay First* (M-LWDF) [12]. RR is fair among users, although it fails to achieve any multiuser or multichannel diversity gain. On the other hand, M-LWDF considers both channel quality and QoS indicators in its scheduling criteria by allocating the resources to the user with the highest potential rate and delay product. According to [2], the M-LWDF algorithm is throughput optimal; that is, it gets the maximum possible diversity gain for stable queues. Other scheduling algorithms such as *Best Channel* (BC) or *Proportional Fair* (PF) algorithms achieve better throughput for some users, but this comes at the expense of others, who experience throughput starvation [12]. As mentioned earlier, the delay associated to the scheduling process was obtained from simulations.

Table 2: Parameters associated to the MAC layer model.

Parameter	Description
—	Scheduling algorithm
B_{L2a}	Size of $L2a$-PDUs
H_{L2a}	Header length of $L2a$-PDUs

The error rate at the MAC layer (P_{L2a}) depends on the BER achieved at the physical layer (P_{L1}) and the size of an $L2a$-PDU (B_{L2a}). In order to provide an expression for the Block Error Rate (BLER) at the MAC layer, P_{L2a}, instantaneous BER is assumed to be equally distributed along bits, which is reasonably true if proper interleaving is performed.

A summary of the performance indicators and parameters at the MAC layer is shown in (5)-(6) and Table 2, respectively.

MAC Model is

$$P_{L2a} \approx 1 - \left(1 - P_{L1}\right)^{B_{L2a}}, \tag{5}$$

$$R_{L2a} = \sum_{k=1}^{Nc} R_{i,k}[n], \quad R_{i,k}[n] = \begin{cases} r_{i,k}[n], & \text{if channel } k \text{ is assigned to user } i, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

3.2.2. RLC Layer Model. While some streaming applications are error-tolerant, others may require reliable data delivery. In this case, the network can optionally retransmit erroneous $L2b$-PDUs (i.e., RLC blocks). Thus, the error rate can be lowered at the expense of decreasing throughput and increasing mean delay and jitter.

The retransmission mechanism analyzed in this paper considers a generic link layer retransmission scheme (based on the ARQ protocol) [13]. ARQ protocol behavior is described as follows. Incoming upper layer PDUs are segmented into N_r $L2b$-PDUs and buffered. The transmitter sends all $L2b$-PDUs and polls the receiver in the last $L2b$-PDU of a higher layer PDU ($L2c$-PDU). A status report request is issued if no response is received to the polling upon expiration of T_{w_stat}. Selective acknowledgement is used to report which $L2b$-PDUs have been incorrectly received. Nonacknowledged $L2b$-PDUs are retransmitted if the maximum number of retransmissions has not been reached; we call cycle i the ith (re)transmission attempt. Further details can be found in [14].

Assuming a maximum number of retransmission attempts N_{rtx}, the loss rate P_{L2b} is given by the probability that an *L2b*-PDU is not correctly received after N_{rtx} retransmissions, that is, $P_{L2b} = P_{L2a}^{N_{rtx}+1}$.

MAC layer throughput comes from the aggregation of two types of PDUs: data and control *L2b*-PDUs, that is, $R_{L2a} = R_{L2a}^{\text{data}} + R_{L2a}^{\text{stat}}$. The first contribution, R_{L2a}^{data}, is computed as

$$R_{L2a}^{\text{data}} = R_{L2b} \cdot \left[1 + \sum_{i=1}^{N_{rtx}} i \cdot P_{L2a}^{i}\right] \cdot \frac{B_{L2b}^{\text{data}}}{B_{L2b}^{\text{data}} - H_{L2b}}, \tag{7}$$

where the term between brackets represents the average number of (re)transmissions per *L2b*-PDU, and the last term corresponds to the RLC overhead.

The second contribution, R_{L2a}^{stat}, represents the throughput generated by status report requests. Such requests are sent whenever no answer to the last *L2b*-PDU of a cycle is received. This contribution is given by

$$R_{L2a}^{\text{stat}} = R_{L2b} \cdot \frac{N_{\text{cycles}}}{N_{L2b/L2c}} \cdot P_{L2a} \cdot \frac{B_{L2b}^{\text{stat}}}{B_{L2b}^{\text{data}}}, \tag{8}$$

where N_{cycles} represents the required mean number of retransmission cycles to send one *L2c*-PDU; B_{L2b}^{data} and B_{L2b}^{stat} represent the mean size of a data and control *L2b*-PDU, respectively; $N_{L2b/L2c}$ represents the number of *L2b*-PDUs per *L2c*-PDU (including retransmissions).

If n_i is the number of (re)transmitted *L2b*-PDUs in cycle i, then the average number of *L2b*-PDUs per *L2c*-PDU can be computed as

$$N_{L2b/L2c} = \sum_{i=0}^{N_{rtx}} \sum_{j=1}^{N_r} \left(j \cdot \Pr\{n_i = j\}\right), \tag{9}$$

where $\Pr\{n_i = j\}$ is the probability of sending j *L2b*-PDUs in cycle i, given by the following recursion:

$$\begin{aligned} &\Pr\{n_i = j\} \\ &\quad = \sum_{k=j}^{N_r} \Pr\{n_{i-1} = k\} \cdot B(k,j) \text{ being } B(k,j) \\ &\quad = \binom{k}{j} \cdot P_{L2a}^{j} \cdot (1 - P_{L2a})^{k-j}. \end{aligned} \tag{10}$$

Solving the recursion

$$\begin{aligned} &\Pr\{n_i = j\} \\ &= \sum_{n_{i-1}=j}^{N_r} \sum_{n_{i-2}=n_{i-1}}^{N_r} \cdots \sum_{n_m=n_{m+1}}^{N_r} \left(\Pr\{n_0 = j\} \cdot \left(\prod_{r=0}^{i-1} B(n_r, n_{r+1})\right)\right). \end{aligned} \tag{11}$$

The required mean number of retransmission cycles to send one *L2c*-PDU can be expressed as

$$N_{\text{cycles}} = \sum_{i=0}^{N_{rtx}} P_i, \tag{12}$$

Table 3: Parameters associated to the RLC layer model.

Parameter	Description
N_{rtx}	Maximum number of RLC retransmissions
T_{w_stat}	Timeout to retransmit new polling request
B_{L2b}^{data}	Size of data *L2b*-PDUs
B_{L2b}^{stat}	Size of status report *L2b*-PDUs
H_{L2b}	Header length of *L2b*-PDUs

where P_i is the probability of requiring the ith cycle to successfully complete the transmission, computed as [14]

$$\begin{aligned} &P_i \\ &\quad = 1 - \sum_{k=0}^{i-1} \left[\sum_{n_0=1}^{N_r} \left(\Pr\{n_0\} \cdot \left[\left(1 - P_{L2a}^{k+1}\right)^{n_0} - \left(1 - P_{L2a}^{k}\right)^{n_0}\right]\right)\right]. \end{aligned} \tag{13}$$

From previous equations, RLC throughput is given by

$$\begin{aligned} &R_{L2b} \\ &\quad = \frac{R_{L2a}}{\left[\left(1+\sum_{i=1}^{N_{rtx}} i \cdot P_{L2a}^{i}\right) \cdot \frac{B_{L2b}^{\text{data}}}{B_{L2b}^{\text{data}} - H_{L2b}}\right] + \left(\frac{P_{L2a} \cdot N_{\text{cycles}}}{N_{L2b/L2c}} \cdot \frac{B_{L2b}^{\text{stat}}}{B_{L2b}^{\text{data}}}\right)}. \end{aligned} \tag{14}$$

To compute the mean *L2b*-PDU delay, D_{L2b}, we need to analyze the impact of retransmissions, which depend on the loss rate at the next lower layer, P_{L2a}. In particular, the additional delay introduced by retransmissions at each cycle comes from (a) delay in retransmitting n_i *L2b*-PDUs, noted as D_{n_i}; and (b) delay in correctly receiving the status report from the receiver, noted as $T_{\text{stat\_}ok}$. These factors are combined as follows

$$D_{L2b} = D_{L2a} + \sum_{i=1}^{N_{rtx}} {P_{L2a}}^{i} \cdot \left(D_{n_i} + T_{\text{stat\_}ok}\right), \tag{15}$$

where the terms D_{n_i} and $T_{\text{stat\_}ok}$ can be computed as a function of $D_{L2a}, P_{L2a}, N_{rtx}, B_{L2b}^{\text{data}}, B_{L2c}$, and T_{w_stat}, whose details can be found in [14].

A summary of the performance indicators and parameters at the RLC layer is shown in (16)–(18) and Table 3, respectively

$$P_{L2b} = P_{L2a}^{N_{rtx}+1}, \tag{16}$$

$$R_{L2b} = f\left(R_{L2a}, P_{L2a}, N_{rtx}, B_{L2b}^{\text{data}}, B_{L2b}^{\text{stat}}, H_{L2b}, B_{L2c}\right), \tag{17}$$

$$D_{L2b} = f\left(D_{L2a}, P_{L2a}, N_{rtx}, T_{w_{\text{stat}}}, B_{L2b}^{\text{data}}, B_{L2c}\right). \tag{18}$$

3.2.3. PDCP Layer Model. The PDCP layer is in charge of adapting the data to achieve efficient transport through the radio interface. This layer performs header compression, which reduces network and transport headers (e.g., TCP/IP or RTP/UDP/IP). The most advanced header compression technique is known as *RObust Header Compression* (ROHC)

[15], which has been adopted by cellular standardization bodies such as 3 GPP. Using ROHC, the RTP/UDP/IPv4 header is compressed from 40 bytes to approximately 1 to 4 bytes, providing a compression gain G_c.

$L2c$-PDU loss rate, P_{L2c}, comes from erroneous $L2c$-PDUs (P_{L2c}^{error}), and $L2c$-PDUs discards at PDCP queues (P_{L2c}^{over})

$$P_{L2c} = P_{L2c}^{\text{error}} + P_{L2c}^{\text{over}} - P_{L2c}^{\text{error}} \cdot P_{L2c}^{\text{over}} \approx P_{L2c}^{\text{error}} + P_{L2c}^{\text{over}}. \tag{19}$$

In the access node, there is one dedicated PDCP buffer for each connection, whose size is Q_{L2c}. The term P_{L2c}^{over} is determined by the buffer size and the incoming traffic load.

Taking into account that an $L2c$-PDU is correctly transmitted if the N_r $L2b$-PDUs (in which it was segmented) arrive correctly at the receiver, the term P_{L2c}^{error} is computed as the probability of requiring at least $N_{rtx} + 1$ retransmissions, $P_i|_{i=N_{rtx}+1}$ (see (13)):

$$\begin{aligned} P_{L2c}^{\text{error}} &= P_i|_{i=N_{rtx}+1} \\ &= 1 - \sum_{k=0}^{N_{rtx}} \left[\sum_{j=1}^{N_r} \left(\Pr\{n_0 = j\} \cdot \left[\left(1 - P_{L2a}^{k+1}\right)^j - \left(1 - P_{L2a}^{k}\right)^j \right] \right) \right]. \end{aligned} \tag{20}$$

The computation of PDCP throughput, R_{L2c}, must take into account the lower layer throughput as well as the effect of ROHC. Assuming an average ROHC compression gain G_c, R_{L2c} is given by the following expression:

$$R_{L2c} = R_{L2b} \cdot \frac{B_{L2c} + (H_{L3} + H_{L4}) \cdot (1 - G_c^{-1})}{B_{L2c} + H_{L2b}}. \tag{21}$$

Average $L2c$-PDU delay has been defined as the time elapsed from when a PDU arrives (from upper layers) to the PDCP sublayer at the transmitter until an acknowledgement is received from the receiver. Hence, the average delay at the PDCP layer, D_{L2c}, comprises the time to correctly receive all $L2b$-PDUs in which an $L2c$-PDU is segmented; such delay includes potential $L2b$-PDU retransmissions, up to a maximum of N_{rtx}. Each retransmission cycle i adds two delay contributions: (a) delay in (re)transmitting n_i $L2b$-PDUs (D_{n_i}), and (b) delay in receiving the status report from the receiver ($T_{\text{stat\_}ok}$):

$$D_{L2c} = \sum_{i=0}^{N_{rtx}} \left(D_{n_i} + P_i \cdot T_{\text{stat\_}ok} \right), \tag{22}$$

where P_i is the probability of requiring i (re)transmission cycles, as defined in (13), whereas where the terms D_{n_i} and $T_{\text{stat\_}ok}$ can be computed as a function of D_{L2a}, $P_{L2a}, B_{L2b}^{\text{data}}, N_{rtx}$, B_{L2c}, and $T_{w\text{\_stat}}$, whose details can be found in [14].

Finally, D_{L2c} can be expressed as

$$\begin{aligned} D_{L2c} &= \sum_{i=0}^{N_{rtx}} \left[\sum_{j=1}^{N_r} \left(\Pr\{n_i = j\} \cdot D_{L2a} \right) \right. \\ &\qquad \left. + P_i \cdot \left(D_{L2a} + P_{L2a} \cdot \left(D_{L2a} + T_{w\text{\_stat}} \right) \right) \right]. \end{aligned} \tag{23}$$

A summary of the performance indicators and parameters at the PDCP layer is shown in (24)–(26) and Table 4, respectively

$$P_{L2c} = f\left(P_{L2a}, N_{rtx}, P_{L2c}^{\text{over}}, B_{L2b}^{\text{data}}, B_{L2c} \right), \tag{24}$$

$$R_{L2c} = f(R_{L2b}, B_{L2c}, H_{L2c}, H_{L3}, H_{L4}, G_c), \tag{25}$$

$$D_{L2c} = f\left(D_{L2a}, P_{L2a}, N_{rtx}, T_{w_{\text{stat}}}, B_{L2b}^{\text{data}}, B_{L2c} \right). \tag{26}$$

3.3. *Network Layer Model.*

The network layer is based on an end-to-end IP connection from the mobile terminal to the streaming server. IP links are assumed to be over-dimensioned compared to radio links. The well-known *Weighted Fair Queuing* (WFQ) multiplexing algorithm was assessed in the IP routers by means of simulations.

End-to-end IP performance is analyzed from the performance results obtained at IP-fixed and radio domains, as shown in Figure 1. The following considerations are made.

(1) The $L3$-PDU loss rate can be computed as the aggregation of the $L3$-PDU losses occurred in each domain: radio ($P_{L3}{}'$) and fixed ($P_{L3}{}''$).

(2) The mean throughput achieved by the mobile terminal is given by the most limiting point in the network, that is, radio interface ($R_{L3}{}'$).

(3) The mean end-to-end IP delay can be computed as the aggregation of the delays experienced in each domain: radio ($D_{L3}{}'$) and fixed ($D_{L3}{}''$).

Considering previous statements, performance indicators and parameters at the IP layer is shown in (27)–(29) and Table 5, respectively.

$$P_{L3} = P_{L3}{}' + P_{L3}{}'' = P_{L2c} + P_{L3}{}'', \tag{27}$$

$$R_{L3} = \min(R_{L3}{}', R_{L3}{}'') = R_{L3}{}' = R_{L2c} \cdot \frac{B_{L3}}{B_{L3} + H_{L2c}}, \tag{28}$$

$$D_{L3} = D_{L3}{}' + D_{L3}{}'' \approx D_{L2c} + D_{L3}{}''. \tag{29}$$

3.4. *Transport Layer Model.*

This section aims to model the performance of three different transport protocols (UDP, TCP, and TFRC) based on performance indicators of the lower layers.

Table 4: Parameters associated to the PDCP layer model.

Parameter	Description
Q_{L2c}	PDCP queues size
H_{L2c}	Header length of $L2c$-PDU
G_c	Compression gain achieved by ROHC

Table 5: Parameters associated to the IP layer model.

Parameter	Description
—	IP multiplexing algorithm
H_{L3}	IP header length (version 4)
N_n	Number of IP nodes from server to client
C_{L3}	Minimum IP link capacity
Q_{L3}	IP queue size

Table 6: Parameters associated to the UDP model.

Parameter	Description
H_{L4}	Transport header length

3.4.1. UDP Model. Since UDP does not include any congestion control or retransmission mechanisms, UDP throughput can be simply computed from the IP throughput by considering the header overhead. Performance indicators and parameters at the UDP layer is shown in (30)–(32) and Table 6, respectively

$$P_{L4} = P_{L3}, \tag{30}$$

$$R_{L4} \approx R_{L3} \cdot \frac{B_{L4}}{(B_{L4} - H_{L4})}\ , \tag{31}$$

$$D_{L4} \approx D_{L3}. \tag{32}$$

3.4.2. TCP Model. TCP includes a congestion control mechanism to react against network congestion. When TCP is used as transport protocol, application throughput behavior depends on the specific TCP implementation. An analytic characterization of the steady-state throughput for TCP-Reno protocol has been applied in this work. This model characterizes TCP throughput as a function of loss rate in the network P_{L3}, Round-Trip-Time (RTT), Retransmission Time-Out duration (T_0), maximum TCP window size (W) for a bulk transfer TCP flow, and the number of packets (b) acknowledged by each received ACK. The complete characterization of the TCP source rate, assuming that the maximum TCP window size has been reached, is computed in [16].

TCP performance is highly sensitive to packet losses because of its inherent congestion control mechanism, which decreases the window transmission, even if such losses are not due to congestion. Besides, the higher the RTT, the lower the throughput at the transport layer, because the congestion window is increased at a rate of RTT.

An appropriate congestion window setting (in addition to adequate queue dimensioning in network elements) is a key factor in optimizing end-to-end performance. In particular, the maximum window size is suggested to be slightly higher than the Bandwidth-Delay Product (BDP) [3] in order to exploit the available radio capacity. Consequently, a maximum TCP window size of $W = 32\,\text{kB}$ was chosen. Since queue sizes (per user) are higher than the W value, we may assume that the probability of overflow in the queues is negligible; thus, the contribution to the $L4$-PDU loss rate only comes from lost $L2c$-PDUs at the radio interface. In a steady state, TCP source rate, that is, incoming rate to $L3$ (S_{L3}), can be characterized by [16]

$$S_{L3} = \frac{(\mathfrak{A}/p) + W + Q(W)(1/\mathfrak{A})}{\mathrm{RTT}((b/8)W + (\mathfrak{A}/pW) + 2) + Q(W) \cdot T_0 \cdot (f(p)/\mathfrak{A})}, \tag{33}$$

where $\mathfrak{A}$ denotes $(1 - p)$, where p represents the loss rate in the network P_{L3}, and the RTT can be approximated by the mean two-way delay over the end-to-end network: $\mathrm{RTT} \approx 2 \cdot D_{L3}$.

From (33), the following dependence is clearly identified: $S_{L3} = \Phi(D_{L3}, P_{L3})$ where Φ represents the TCP throughput (33). In addition, average delay D_{L3} and loss rate P_{L3} in the network depend on the total network load, $S_{L3} \cdot N_u$, for example, high load in the network lead to higher delays and losses. Hence, the source rate S_{L3} can be computed by solving the following system of equations:

$$\begin{aligned} S_{L3} &= \Phi(D_{L3}, P_{L3}), \\ D_{L3} &= \mathrm{f}_1(S_{L3} \cdot N_u), \\ P_{L3} &= \mathrm{f}_2(\mathrm{S}_{L3} \cdot N_u), \end{aligned} \tag{34}$$

which can be expressed by the following equation:

$$S_{L3} = \Phi(\mathrm{f}_1(S_{L3} \cdot N_u), \mathrm{f}_2(S_{L3} \cdot N_u); W, b, T_0). \tag{35}$$

In order to solve this nonlinear equation, the behavior of $D_{L3} = \mathrm{f1}(S_{L3} \cdot N_u)$ has been parameterized using standard curve fitting methods from the result of (29) and (26).

TCP delay depends on the probability of retransmissions and the period of time D_{loss} required by the transmitter to detect the need for a retransmission (via duplicated ACKs or timer expiration). As stated by[16], such a time period D_{loss} can be computed as

$$D_{\text{loss}} \approx \mathrm{RTT} \cdot \left(\frac{2+b}{6} + \sqrt{\frac{2b(1-p)}{3p} + \left(\frac{2+b}{6}\right)^2} + 1 \right) + Q(W) \cdot T_0 \cdot \frac{f(p)}{1-p}. \tag{36}$$

Thus, TCP delay (D_{L4}) can be computed from the IP level delay by adding the effect of TCP retransmissions:

$$D_{L4} \approx D_{L3} + \sum_{i=0}^{\infty} (P_{L3})^i \cdot (D_{\text{loss}} + D_{L3}). \tag{37}$$

Table 7: Parameters associated to the TCP model.

Parameter	Description
W	Maximum TCP window size
b	Number of packets that are acknowledged by a received ACK
T_0	Retransmission Time-Out
H_{L4}	Transport header length

Once S_{L3} is obtained by solving the aforementioned nonlinear equation, performance indicators and parameters at the TCP layer is shown in (38)–(40) and Table 7, respectively

$$P_{L4} = 0, \tag{38}$$

$$\begin{gathered} R_{L4} \approx S_{L3} \cdot \left(1 - \sum_{i=0}^{\infty} (P_{L3})^i\right), \\ S_{L3} = \Phi(\mathrm{f}_1(S_{L3} \cdot N_u), \mathrm{f}_2(S_{L3} \cdot N_u); W, b, T_0), \end{gathered} \tag{39}$$

$$D_{L4} \approx D_{L3} + \sum_{i=0}^{\infty} (P_{L3})^i \cdot (D_{\text{loss}} + D_{L3}). \tag{40}$$

Note that the transport layer becomes error-free ($P_{L4} = 0$) since TCP is a reliable protocol.

3.4.3. TFRC Model. TFRC has less throughput variation over time in comparison to TCP, which, in principle, makes it more suitable for real-time applications such as telephony or streaming media where a relatively smooth sending rate is important. The recommended TFRC throughput equation described in [1] was used, which is a simplified version of the throughput equation for Reno TCP when $P_{L4} < 0.54$ and no delayed-ACK is applied; that is, $b = 1$ [17]. TFRC source throughput can be computed by [17]

$$S_{L3} = \frac{1}{\mathrm{RTT} \cdot \sqrt{(2p/3)} + T_0 \cdot 3 \cdot \sqrt{(3p/8)} \cdot p(1 + 32p^2)}. \tag{41}$$

The evaluation of S_{L3} is performed following the same procedure as in the TCP case, that is, resolving the nonlinear equation described in (35). A summary of the performance indicators and parameters at the TFRC layer is shown in (42)–(44) and Table 8, respectively

$$P_{L4} \approx P_{L2c}, \tag{42}$$

$$\begin{gathered} R_{L4} \approx S_{L3} \cdot (1 - P_{L3}), \\ S_{L3} = \Phi(\mathrm{f}_1(S_{L3} \cdot N_u), \mathrm{f}_2(S_{L3} \cdot N_u); W, b, T_0), \end{gathered} \tag{43}$$

$$D_{L4} \approx D_{L3}. \tag{44}$$

Since TFRC only includes the congestion control mechanism (and not retransmissions), losses remaining at the transport layer come from noncorrected errors at the radio link, and TFRC delay is similar to the network delay (D_{L3}).

Table 8: Parameters associated to the TFRC model.

Parameter	Description
b	Number of packets that are acknowledged by a received ACK
T_0	TFRC timer used for rate adaptation
H_{L4}	Transport header length

Table 9: Parameters associated to the application layer model.

Parameter	Description
N_u	Number of users
H_{L5}	RTP header length
L	Socket buffer size

3.5. Application Layer Model. The application layer is responsible for establishing the streaming session, and thereafter, for transferring the multimedia content (at the server side) and reproducing the content (at the client side).

The streaming server delivers application data to the transport layer at an average rate defined by the codec (see Table 10). However, if the transport layer includes a congestion control mechanism (e.g., TCP or TFRC), the socket between these layers must temporarily buffer the packets when the transport layer rate is lower than the codec rate. This mechanism has been approximated by an M/M/1/L queue system where the arrival rate is given by $\lambda = S_{L4}/B_{L4}$ and the service rate is given by $\mu = S_{L3}/B_{L3}$. The loss rate in an M/M/1/L queue is given by

$$P_{\text{socket}} = \frac{\rho^L(1-\rho)}{1-\rho^{L+1}}, \quad \rho = \frac{\lambda}{\mu}, \tag{45}$$

whereas the average waiting time in the socket can be obtained from

$$D_{\text{socket}} = \frac{\rho(1 - (L+1)\rho^L + L\rho^{L+1}) - (\rho - \rho^{L+1})(1-\rho)}{\lambda(1-\rho)(1-\rho^{L+1})} + \frac{1}{\mu}. \tag{46}$$

On the receiver side, the application layer adds an additional delay because of the application buffer of the streaming player. A sufficiently large application buffer size that hides network jitter to application performance has been assumed. Then, considering that the application throughput is not interrupted by buffer starvation, the following expressions can be obtained.

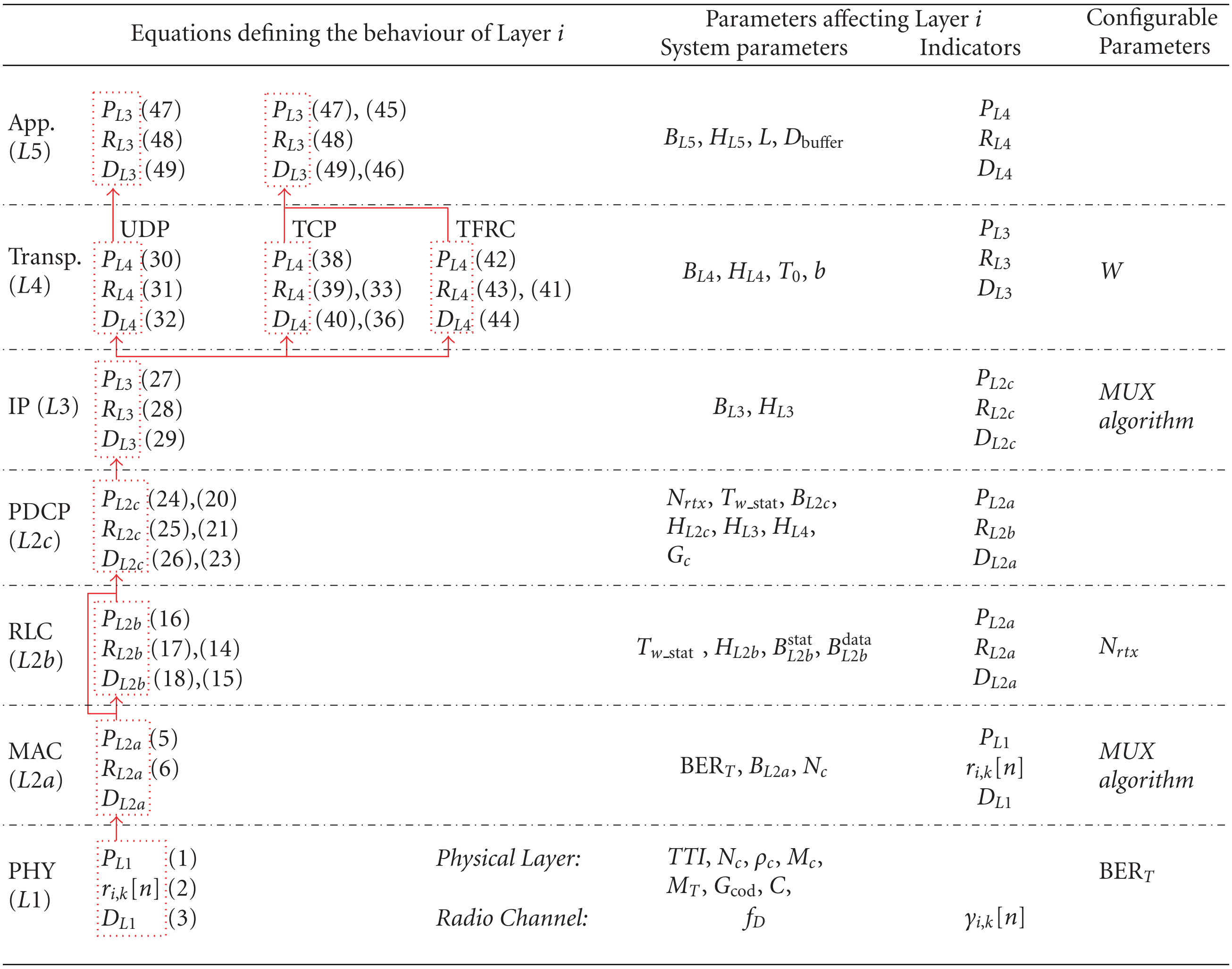
Performance indicators and parameters at the aaplication layer is shown in (47)–(49) and Table 9, respectively.

$$P_{L5} = P_{L4} + P_{\text{socket}}, \tag{47}$$

$$R_{L5} = R_{L4} \cdot \frac{B_{L5}}{(B_{L5} - H_{L5})}, \tag{48}$$

$$D_{L5} \approx D_{L4} + D_{\text{Buffer}} + D_{\text{socket}}, \tag{49}$$

where P_{socket} and P_{socket} contributions are only applicable to TCP- or TFRC-based applications.



	Equations defining the behaviour of Layer i				Parameters affecting Layer i		Configurable Parameters
					System parameters	Indicators	
App. ($L5$)	P_{L3} (47) R_{L3} (48) D_{L3} (49)	P_{L3} (47), (45) R_{L3} (48) D_{L3} (49),(46)			B_{L5}, H_{L5}, L, D_{buffer}	P_{L4} R_{L4} D_{L4}	
Transp. ($L4$)	UDP P_{L4} (30) R_{L4} (31) D_{L4} (32)	TCP P_{L4} (38) R_{L4} (39),(33) D_{L4} (40),(36)	TFRC P_{L4} (42) R_{L4} (43), (41) D_{L4} (44)		B_{L4}, H_{L4}, T_0, b	P_{L3} R_{L3} D_{L3}	W
IP ($L3$)	P_{L3} (27) R_{L3} (28) D_{L3} (29)				B_{L3}, H_{L3}	P_{L2c} R_{L2c} D_{L2c}	*MUX algorithm*
PDCP ($L2c$)	P_{L2c} (24),(20) R_{L2c} (25),(21) D_{L2c} (26),(23)				N_{rtx}, T_{w_stat}, B_{L2c}, H_{L2c}, H_{L3}, H_{L4}, G_c	P_{L2a} R_{L2b} D_{L2a}	
RLC ($L2b$)	P_{L2b} (16) R_{L2b} (17),(14) D_{L2b} (18),(15)				T_{w_stat} , H_{L2b}, B_{L2b}^{stat}, B_{L2b}^{data}	P_{L2a} R_{L2a} D_{L2a}	N_{rtx}
MAC ($L2a$)	P_{L2a} (5) R_{L2a} (6) D_{L2a}				BER_T, B_{L2a}, N_c	P_{L1} $r_{i,k}[n]$ D_{L1}	*MUX algorithm*
PHY ($L1$)	P_{L1} (1) $r_{i,k}[n]$ (2) D_{L1} (3)			*Physical Layer:* *Radio Channel:*	TTI, N_c, ρ_c, M_c, M_T, G_{cod}, C, f_D	 $\gamma_{i,k}[n]$	BER_T

Figure 3: Summary of the end-to-end QoS model.

Table 10: Content encoding description.

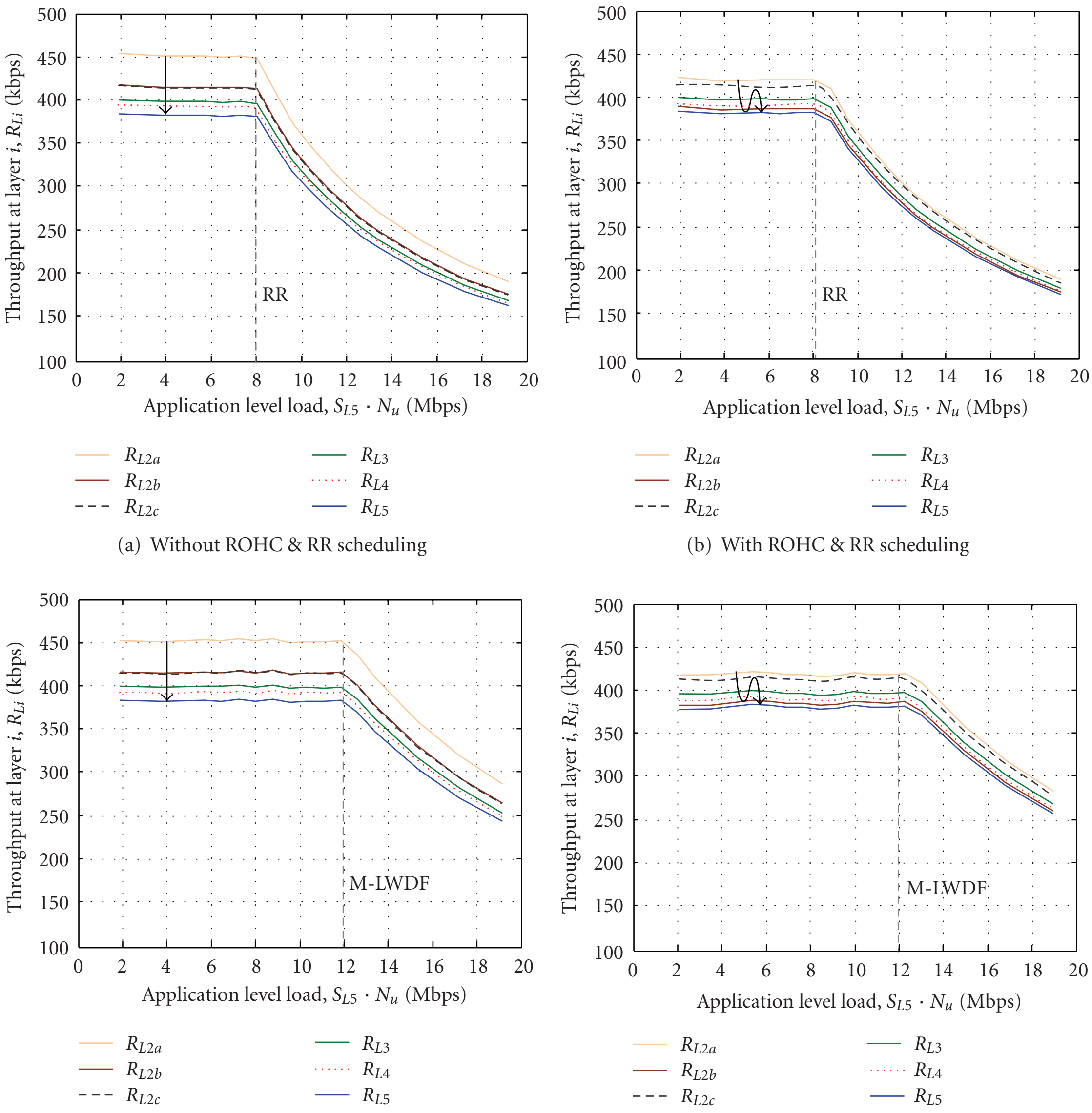
Parameter	Description	Value
S_{L5}	Mean source rate at application layer	384 kbps
V_R	Video resolution	QVGA, 320 × 240 pixels
F_R	Frame rate	15 frames/sec
F	Video encoding format	3 GPP (based on MPEG-4)

From the end user perspective, the delay introduced by the application buffer, D_{Buffer}, can be considered as part of the session establishment, since the application does not start reproducing media until the buffer is full. The buffer usually spans from 1 to 10 (depending on the technology). However, in two-way streaming services (like Push-to-Talk over Cellular, PoC) the lower limit is generally small (not higher than 500 ms) since the interactivity requirements are much stricter than they are in one-way streaming services.

4. Results

The end-to-end QoS model shown in Figure 3 was used for different purposes. Firstly, the model is used to estimate the performance at different protocol layers for a UDP-based streaming solution. Then, a design example for TCP-based applications is described.

4.1. Performance Estimation. Figure 4 shows an example of performance estimation for a UDP-based streaming solution. Average throughput at different layers is shown as a function of the total application load, $S_{L5} \cdot N_u$. Mean source rate per user was kept constant ($S_{L5} = 384$ kbps) while the number of users N_u in the system increased. Figures 4(a) and 4(c) on the left show throughput results without header compression (ROHC), whereas Figures 4(b) and 4(d) on the right include this feature.



(a) Without ROHC & RR scheduling

(b) With ROHC & RR scheduling

(c) Without ROHC & M-LWDF scheduling

(d) With ROHC & M-LWDF scheduling

FIGURE 4: Throughput results for UDP-based streaming.

Analyzing the performance shown in Figure 4, the following effects can be observed at layer i.

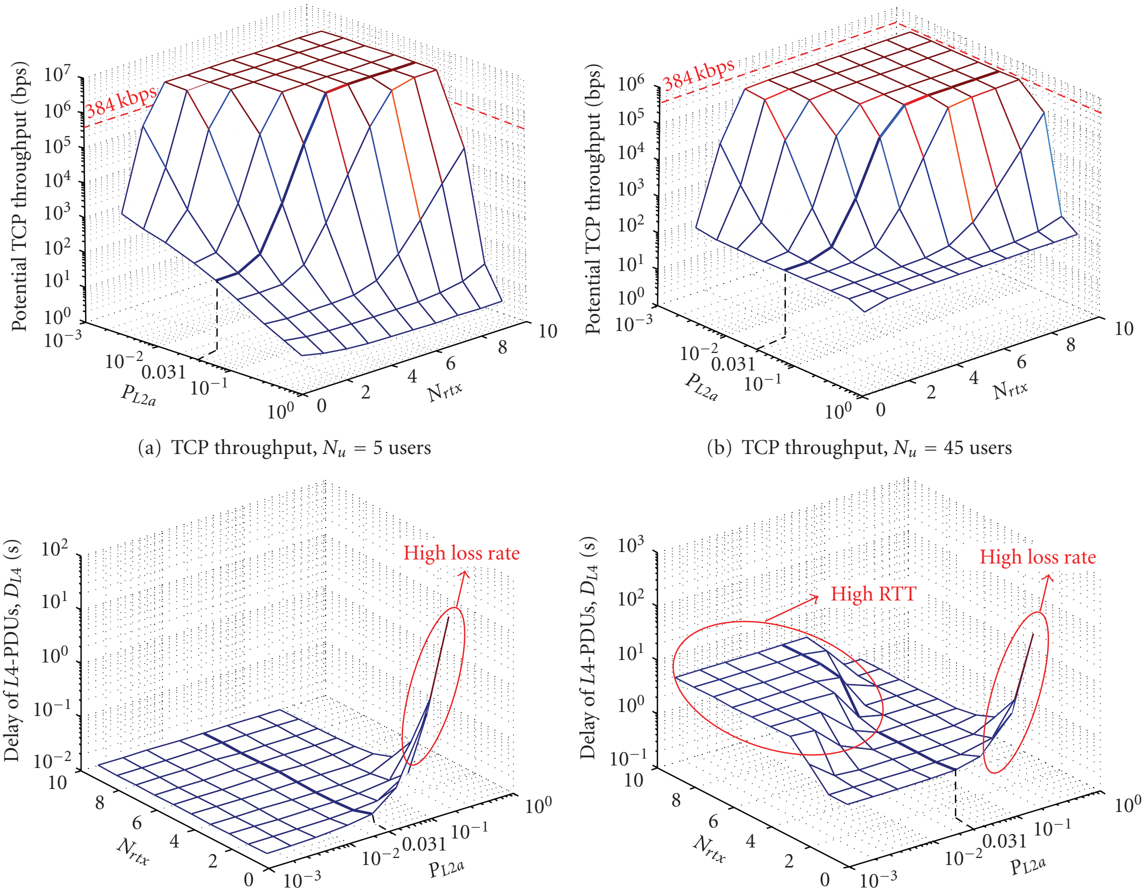
- ($L2a$) MAC layer throughput, R_{L2a}, is rapidly degraded above a certain *critical load* point, which corresponds to the maximum achievable system throughput for a particular multiplexing algorithm. As expected, the M-LWDF algorithm achieves a higher system throughput (about 12 Mbps with scenario settings) than RR, since M-LWDF takes Channel State Information (CSI) into account, thus providing a higher diversity gain [12].

- ($L2b$) The RLC layer introduces additional throughput degradation due to $L2b$-PDU retransmissions, as described in (17).

- ($L2c$) The use of ROHC makes it possible to decrease the required amount of resources below the PDCP layer while achieving the same application level throughput. Specifically, ROHC achieves a capacity gain of 7% in our scenario. Due to compression, the PDCP layer may even compute a higher throughput (after decompression) than the lower layers, as illustrated in Figures 4(b) and 4(d).

- ($L3$–$L5$) Throughput at the upper layers only suffers from RTP/UDP/IP header overheads.

Throughput curves in Figure 4 also provide very valuable information about the required resources at each layer in order to fulfill the desired QoS at the application level. For instance, the proposed model is able to map application level QoS requirements onto lower layer requirements; for



(a) TCP throughput, $N_u = 5$ users

(b) TCP throughput, $N_u = 45$ users

(c) TCP delay, $N_u = 5$ users

(d) TCP delay, $N_u = 45$ users

Figure 5: Effect of the number of retransmissions on TCP throughput and delay.

example, a 384 kbps coding rate requires performing a resource reservation of 400 kbps at the IP level or assigning 450 kbps at MAC layer scheduling.

4.2. End-to-End Design. In this section, an end-to-end design example for TCP-based applications is described. The analysis is focused on those parameters having a higher influence on the overall performance: TCP window size (W), maximum number of RLC retransmissions (N_{rtx}), and number of users in the system (N_u). The following parameter values were used: $b = 2$ packets and $T_0 = 4 \cdot \text{RTT}$.

Figure 5 shows the maximum achievable TCP throughput and delay as a function of N_{rtx} and loss rate at the MAC layer after decoding (P_{L2a}) for $W = 32$ kB. Results are shown for two load conditions ($N_u = 5$ users and $N_u = 45$ users).

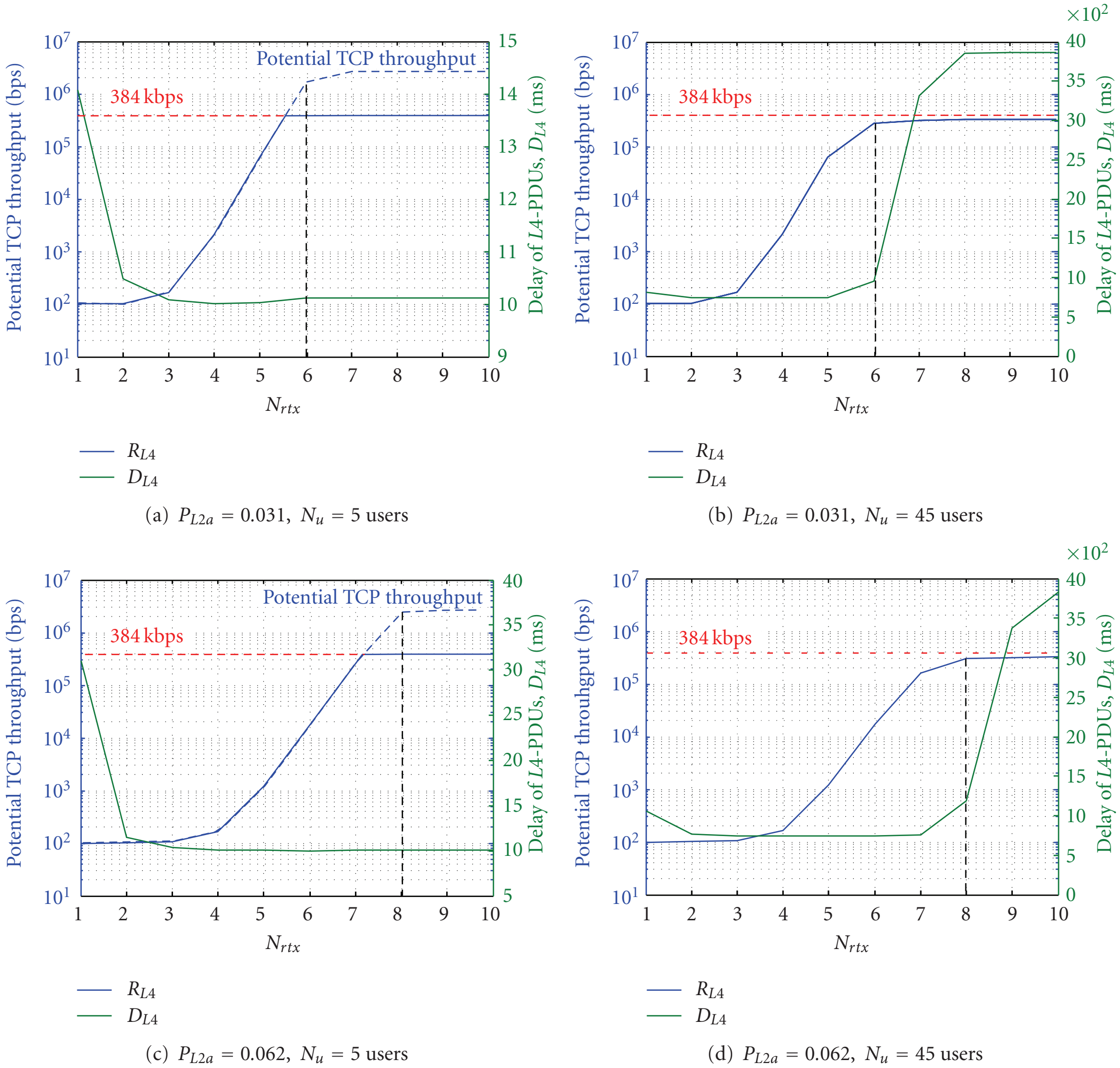
In terms of TCP throughput results, which are depicted in Figures 5(a) and 5(b), it is shown how high P_{L2a} values require a higher number of RLC retransmissions to minimize data losses, and consequently, maximize throughput. For low load conditions ($N_u = 5$ users), potential TCP throughput is higher than the video codec rate (384 kbps) as long as a proper N_{rtx} value is configured. However, for high load conditions ($N_u = 45$ users), TCP is not able to achieved the desired throughput.

Concerning TCP delay results, shown in Figures 5(c) and 5(d), two scenarios are analyzed.

(a) Low load ($N_u = 5$): in general, high loss rates at MAC sublayer (P_{L2a}) must be reduced by RLC retransmissions (configuring a high value of N_{rtx} parameter). As the radio interface delay is very low in low load conditions, the impact of RLC retransmissions on TCP delay is almost negligible. Otherwise, if N_{rtx} is set to a low value, TCP will be responsible for performing end-to-end retransmissions, thus increasing $L4$-PDUs delay.

(b) High load ($N_u = 45$): in addition to the previous effect, high load conditions increase the radio interface delay, and thus consecutive RLC retransmissions will increase the end-to-end RTT. As the TCP delay depends on the average RTT, a high N_{rtx} will leads to high TCP delays. Besides, as the TCP throughput (per user) increases for high N_{rtx} values, the overall load in the network is higher, thereby further increasing the TCP delay.

According to the results shown in Figure 5, for a given P_{L2a} there is an optimum N_{rtx} value that maximizes



(a) $P_{L2a} = 0.031$, $N_u = 5$ users

(b) $P_{L2a} = 0.031$, $N_u = 45$ users

(c) $P_{L2a} = 0.062$, $N_u = 5$ users

(d) $P_{L2a} = 0.062$, $N_u = 45$ users

FIGURE 6: Potential throughput versus delay at the transport layer (TCP).

throughput while keeping delay as low as possible. This value depends on the loss rate in the network. For instance, for $P_{L2a} = 0.031$ (obtained from a $\text{BER}_T = 10^{-4}$ at the physical layer), the optimum value of N_{rtx} is 6.

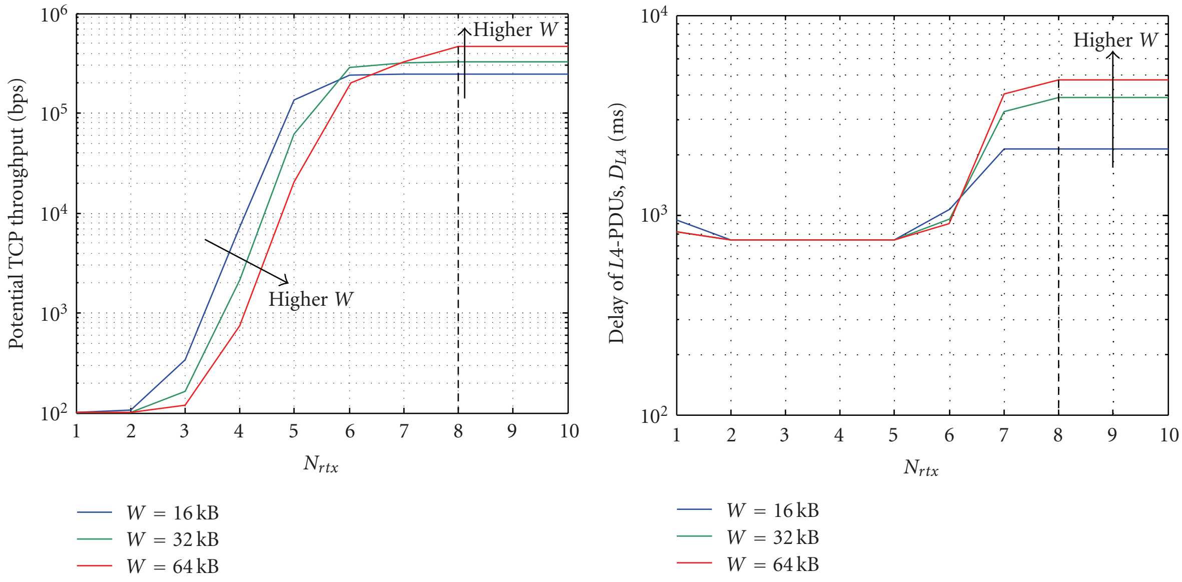
Figure 6 shows joint TCP throughput and delay results for different loss rates ($P_{L2a} = 0.031$ and $P_{L2a} = 0.062$) and load conditions ($N_u = 5$ and $N_u = 45$ users). For $P_{L2a} = 0.031$, the minimum value of N_{rtx} that allows achieving the maximum potential throughput is $N_{rtx} = 6$, regardless of the number of users in the system, as shown in Figures 6(a) and 6(b). This minimum value of N_{rtx} is selected in order to minimize the end-to-end delay. However, for $P_{L2a} = 0.062$ ($\text{BER}_T = 2 \cdot 10^{-4}$), the value of N_{rtx} that optimizes the transport layer performance is 8, as shown in Figures 6(c) and 6(d).

The impact of the maximum TCP window size (W) on TCP throughput and delay is shown in Figure 7. Performance results show that excessively small values of the maximum congestion window (W) do not allow one to make full use of network resources, which thus reduces the maximum throughput. On the other hand, excessively large values of W require a high reliability (in terms of loss rate) in order to use the whole window; thus, too many RLC retransmissions are required, which increases the end-to-end delay.

In sum, the values of BER_T, N_{rtx}, and W parameters must be jointly decided upon, making trade-offs between throughput and delay. For a given $\text{BER}_T = 10^{-4}$, a trade-off value for N_{rtx} was 6 in order to limit the end-to-end delay. For these values of BER_T and N_{rtx}, the maximum TCP window that maximizes throughput was $W = 32$ kB.

5. Model Validation

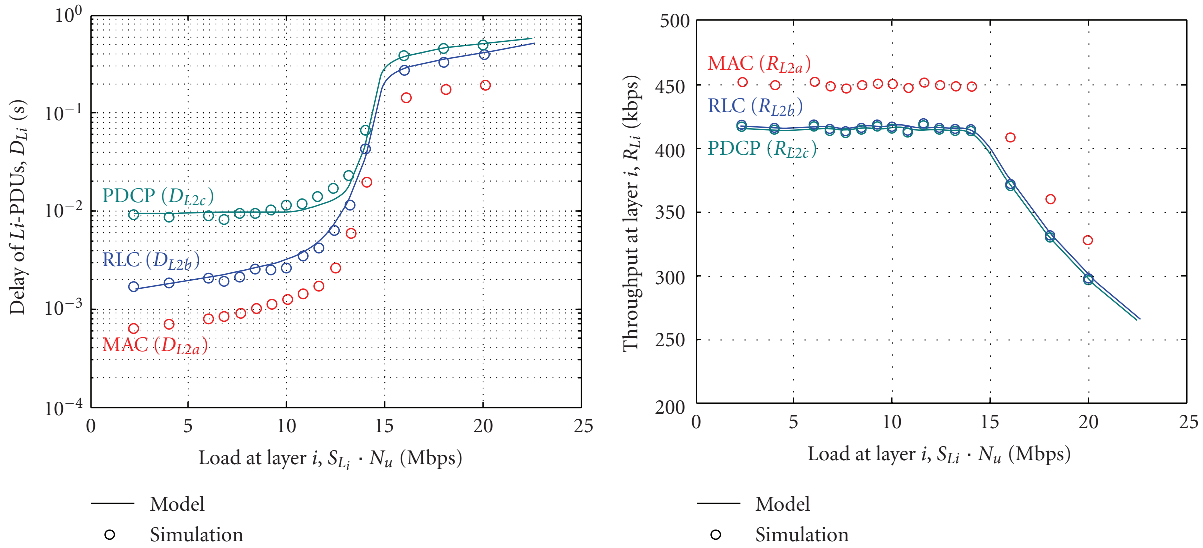
The objective of this section is to validate the theoretical model proposed in this work. The validation process is divided in two phases: (1) validation of the radio interface model, and (2) validation of the upper layer model.



(a)

(b)

Figure 7: Impact of maximum TCP window size (W) on TCP ($\mathrm{BER}_T = 10^{-4}$ and $N_u = 45$ users).



(a)

(b)

Figure 8: Radio Interface Model Validation.

5.1. Radio Interface Model Validation. Since the radio technology under study is not yet available, the validation process of the radio subsystem is based on link level simulations. Such simulations have been performed for a frequency-selective Rayleigh fading channel using adaptive modulation with a $\mathrm{BER}_T = 10^{-4}$. The feedback channel is assumed to be ideal (with no delay or losses).

Figure 8 shows the validation results for the radio interface model, assuming the M-LWDF multiplexing algorithm and $N_{rtx} = 6$. Since the QoS model is based on PHY/MAC layer simulations as a starting point, the goal of these simulations is to validate RLC and PDCP models. Performance estimations from the theoretical model, in terms of delay Figure 8(a) and throughput Figure 8(b), are compared to simulation results.

5.2. Upper Layer Model Validation. The validation process of the upper layer model (i.e., network, transport, and application) was performed by developing a real-time end-to-end system [18]. Figure 9 shows the validation system architecture, which includes the following modules.

(i) Streaming Server. Darwin Streaming Server v5.5.5 was used on the server side. This server allows one to select

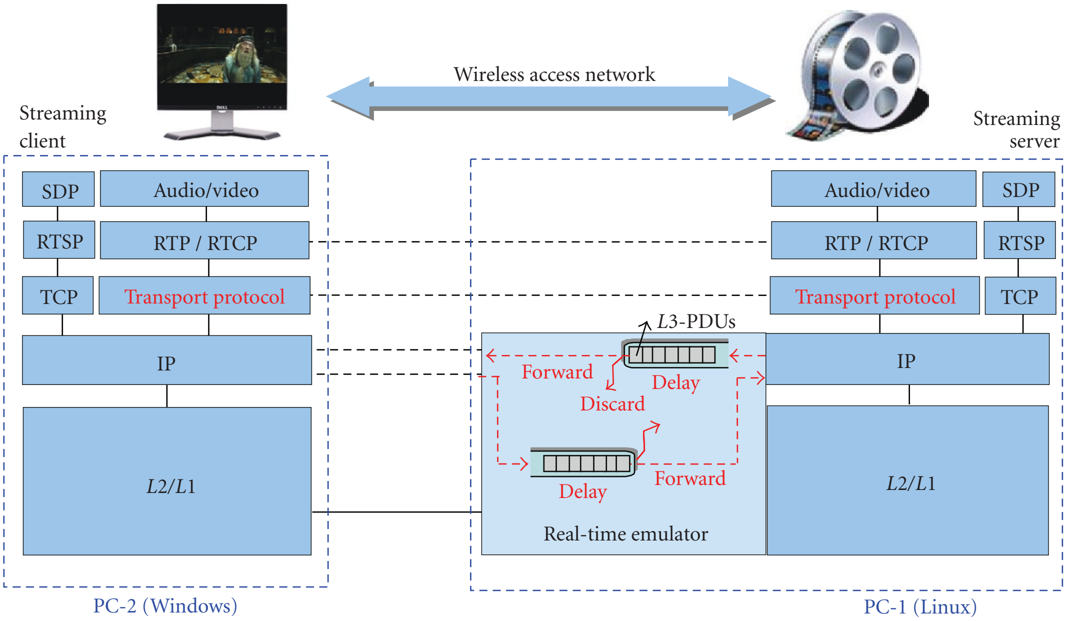


FIGURE 9: Validation system architecture [18].

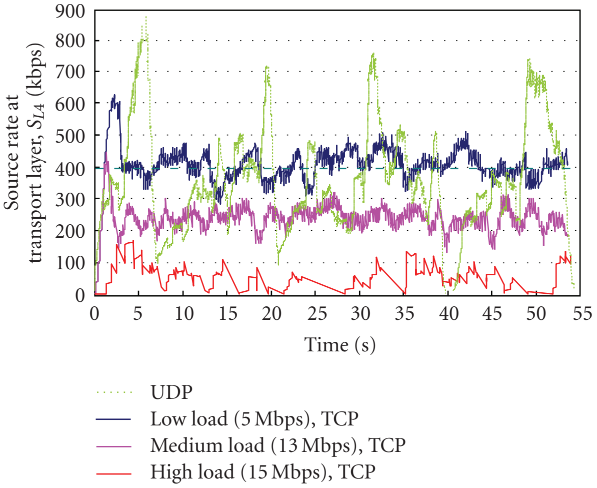


FIGURE 10: Impact of network load on the source rate at the server.

UDP or TCP as the transport protocol. Streaming content is based on a single video flow whose parameters are listed in Table 10. A packet sniffer (*Wireshark v0.99.7*) is used on both sides (server and client) to capture and analyze the traffic between peers.

(ii) Real-Time Emulator. Between the server and the client, a real-time emulator models the behavior of the whole network, so that the client-server connection experiences (in real-time) the quality degradation introduced over the end-to-end path. This emulator uses the packet filtering framework included in the Linux 2.4.x and 2.6.x kernel series together with the *iptables* utility: *iptables* allows one to configure the packet filtering rule set. Certain quality degradation (in terms of delay or packet loss) is applied to the filtered packets. Such degradation is set according to the quality indicators obtained at the IP layer: loss rate P_{L3} and delay D_{L3}. In this way, the emulator offers a real-time data flow that experiences the degradation introduced by the network with wireless access.

(iii) Streaming Client. A *VLC Media Player* .8.6d is responsible for establishing the streaming session with the server and reproducing content. For the TCP-based solution, *TweakMaster* v2.50 was also used to align TCP settings on the client side with the parameters assumed in the theoretical model.

Figure 10 shows the instantaneous source rate generated at the transport layer on the server side, considering the content encoding characteristics described in Table 10. The aim of Figure 10 is to clarify the impact of the network conditions on the UDP and TCP source rate at the server (S_{L4}).

It is shown that UDP delivers data to the network at a source rate determined by the encoding process, independently of the network status (loss rate and delay). Average UDP source rate can be computed from the average application source rate ($S_{L5} = 384$ kbps) and taking into account UDP headers, yielding $S_{L4(\text{UDP})} = 394$ kbps. On the other hand, the TCP source rate at the server is highly influenced by network conditions as a consequence of the TCP congestion control mechanism, which tries to react against congestion. This mechanism leads to an important reduction in the average TCP source rate ($S_{L4(\text{TCP})}$), as the network load increases.

TCP throughput and delay results on the client side obtained from the analytical model are compared to real measurements in Figure 11. Good behavior of the theoretical model is observed. The proposed model provides less accurate values for high load conditions due to the assumption taken during the TCP modeling that the retransmission timeout duration is constant ($T_0 = 4 \cdot \text{RTT}$). In a real implementation, T_0 is adaptively determined by estimating the mean and variance of the RTT [19], thus providing slightly better performance in the real system.

Delay validation results are shown at the transport and application layers. TCP delays were measured by tracing the received ACKs from the terminal (using *Wireshark* and *tcptrace* software), taking into account that $D_{L4} = 1/2\,\mathrm{RTT}$. Validation of RTP delay is more complex, as there is no feedback information from the receiver to measure the RTP RTT. The solution involves using an RTCP time stamp to measure the delay from sender to receiver; this solution requires the sender and receiver to be synchronized via Network Time Protocol (NTP).

6. Use of the Model for QoE Assessment

The proposed end-to-end emulator delivers a detailed real-time analysis and understanding of the service quality for any application and technology by applying a proper configuration. This approach provides a simple mapping from network-level performance indicators to service-level performance indicators.

From a mobile operator's point of view, knowing how subscribers perceive the performance of the services they are offered is a key issue. Quality of Experience (QoE) is the term used to describe this end user perception.

As the complexity of the lower layers in the end-to-end connection is simplified by means of network performance indicators from the QoS model, our proposed emulator is able to run in real-time. This real-time emulator provides certain quality degradation (in terms of delay or packet loss) to the filtered packets, offering a data flow experiencing the degradation that a real end-to-end network would add. In this manner, the user QoE can be assessed for different network types, configurations, and topologies.

Figure 12 shows a comparison between the throughput obtained from the end-to-end model and measurement results for a UDP-based solution. In addition, a snapshot of the video captured at the client side is shown for three different load levels in order to illustrate the image quality degradation as load increases.

The end-to-end emulator can also be used to evaluate the video quality for different network configurations and conditions, either by means of objective metrics like PSNR (*Peak-to-peak Signal-to-Noise Ratio*) or subjective metrics like the MOS (*Mean Opinion Score*). Although recently a number of more complex metrics have been defined, in this paper, the PSNR metric was used to evaluate the video quality for different network loads as it is the most widely used objective video quality metric [20]. PSNR is defined by

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{MaxErr^2 \cdot w \cdot h}{\sum_{i=0,j=0}^{w,h} \left(x_{i,j} - y_{i,j}\right)^2}, \qquad (50)$$

where *MaxErr* represents the maximum possible absolute value of colour components difference, w is the video width, and h is the video height.

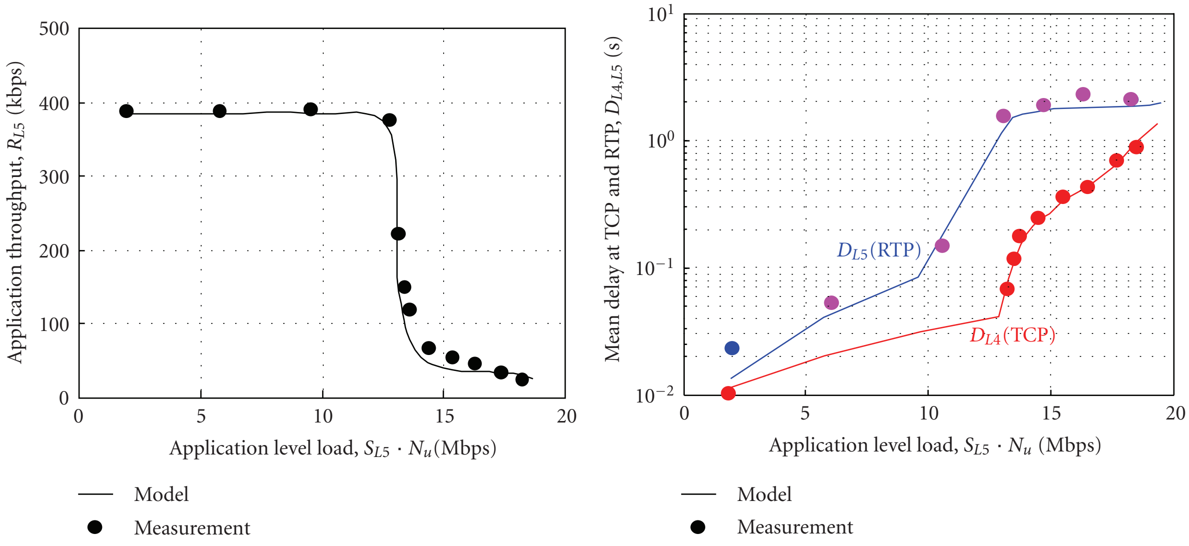
Table 11 shows the average PSNR results obtained from the MSU Video Quality Measurement Tool v2.01. Taking

Table 11: PSNR evaluation of video quality.

Network Load	Application throughput	Average PSNR
11.5 Mbps	381 kbps	36.7 dB
13.4 Mbps	350 kbps	25.1 dB
15.3 Mbps	304 kbps	16.9 dB

Table 12: Numerical Parameters at different layers.

Parameter	Value
Application Layer	
N_u	2–50
H_{L5}	12 bytes
L	64 kbytes
Transport Layer	
W	32 kbytes
b	2 (TCP), 1 (TFRC)
T_0	$4 \cdot \mathrm{RTT}$
H_{L4}	8 bytes (UDP), 20 bytes (TCP), 16 bytes (TFRC)
Network Layer	
Multiplexing	WFQ
H_{L3}	20 bytes
N_n	3
C_{L3}	20 Mbps
Q_{L3}	64 kbytes
PDCP Layer	
Q_{L2c}	32 kbytes
H_{L2c}	1 byte
G_c	10
RLC Layer	
N_{rtx}	3 (UDP), 6 (TCP & TFRC)
T_{w_stat}	200 ms
B_{L2b}^{data}	40 bytes
B_{L2b}^{stat}	4 bytes
H_{L2b}	2 bytes
MAC Layer	
Scheduling	RR, M-LWDF
B_{L2a}	40 bytes
H_{L2a}	0 bytes
Physical Layer	
BER_T	10^{-4}
R_k	0, 2 (QPSK), 4 (16QAM), 6 (64QAM)
TTI	1 ms
N_c	50
ρ_c	0.6
T_B	TTI
M_c	12
M_T	12
G_{cod}	8 dB
Radio Channel	
f_D	8 Hz
$\overline{\gamma}$	15 dB



(a)

(b)

Figure 11: Application throughput validation (TCP-based solution).

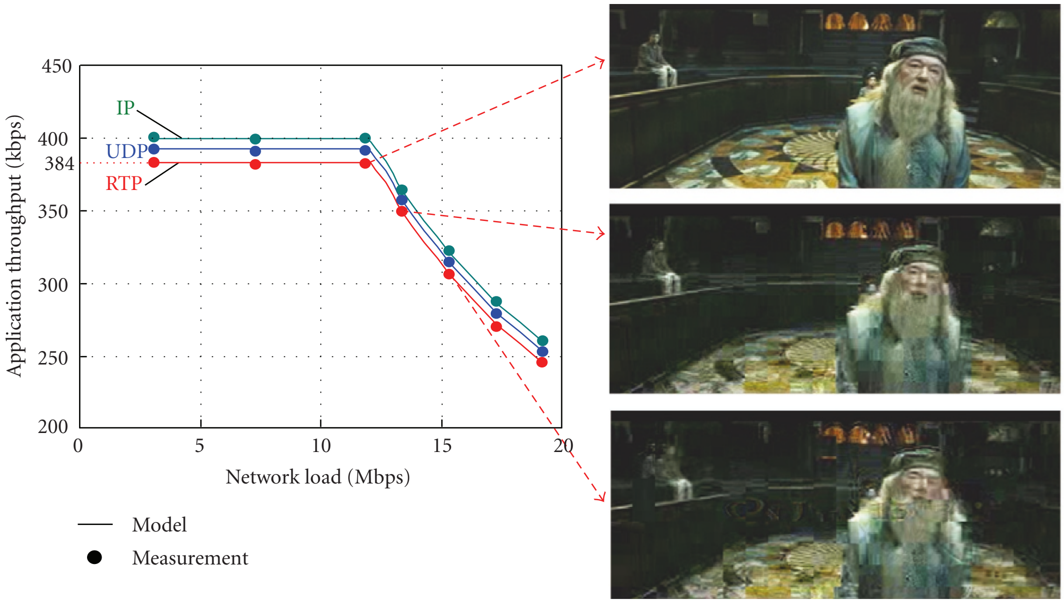


Figure 12: Application throughput validation (UDP-based solution).

into account that PSNR values higher than 35 dB are usually considered good quality, this is only achieved under load conditions below 12 Mbps approximately.

7. Conclusions

In this work, a detailed analysis of the end-to-end QoS assessment over networks with wireless access has been presented. This paper proposes a new modeling methodology based on QoS models for each protocol layer, providing a set of performance indicators across the protocol stack.

Based on this methodology, a QoS model for streaming services has been developed. This model can be used to estimate the performance at any protocol layer. In addition, the model makes it possible to identify the main factors affecting the quality of service, which is very useful for end-to-end parameter optimization. Finally, the model can also be used to map QoS needs at different layers from application requirements (e.g., to reserve appropriate resources at each layer). The framework applied in this work for streaming can be extended to other services (e.g., VoIP) and radio technologies (e.g., WiMax).

In terms of performance results, it was shown that multiplexing algorithms which take into account both channel state information and QoS indicators (such as M-LWDF) provide the best performance (in terms of capacity and fairness). The values of BER_T, N_{rtx}, and W parameters must be jointly decided upon, making trade-offs between throughput and delay; for example, for a given BER_T of 10^{-4}, the maximum number of RLC retransmissions N_{rtx} should be set to 6 in order to limit the end-to-end delay. With these values, the maximum TCP window that maximizes throughput is $W = 32\,\mathrm{kB}$.

In order to validate the proposed QoS model, a real-time emulation platform was developed. Additionally, this emulator makes it possible to experience the end-to-end quality of service and facilitates QoE assessment using appropriate measurement tools.

Acknowledgments

This work is partially supported by the Spanish Government under project TEC-2007-67289 and by the Junta de Andalucia under Proyecto de Excelencia P07-TIC-03226.

References

[1] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): protocol specification," RFC 3448, January 2003.

[2] H. M. Andrews, G. K. Kumaran, R. K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "CDMA data QoS scheduling on the forward link with variable channel conditions," *Bell Labs Technical Memorandum*, 2002.

[3] G. Gómez and R. Sanchez, *End-to-end quality of service over cellular networks: data services performance optimization in 2G/3G*, John Wiley & Sons, New York, NY, USA, 2005.

[4] H. Montes, G. Gómez, R. Cuny, and J. F. Paris, "Deployment of IP multimedia streaming services in third-generation mobile networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 84–92, 2002.

[5] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for video delivery over wireless Internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123–133, 2005.

[6] S. Ci, H. Wang, and D. Wu, "A theoretical framework for quality-aware cross-layer optimized wireless multimedia communications," *Advances in Multimedia*, vol. 2008, Article ID 543674, 10 pages, 2008.

[7] I. Akyilduz, Y. Altunbasak, F. Fekri, and R. Sivakumar, "AdaptNet: an adaptive protocol suite for the next-generation wireless internet," *IEEE Communications Magazine*, vol. 42, no. 3, pp. 128–136, 2004.

[8] Q. Zhang and Y.-Q. Zheng, "Cross-layer design for qos support in multihop wireless networks," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 64–76, 2008.

[9] A. J. Goldsmith, *Wireless Communications*, Cambridge University Press, Cambridge, UK, 2005.

[10] G. Gómez, D. Morales-Jiménez, F. J. López-Martínez, J. J. Sánchez, and J. T. Entrambasaguas, "Radio-interface physical layer," in *Long Term Evolution: 3GPP LTE Radio and Cellular Technology*, chapter 3, CRC Press, Boca Raton, Fla, USA, 2009.

[11] 3GPP 36.201, "Long Term Evolution (LTE) physical layer; general description," V8.3.0, March 2009.

[12] J. T. Entrambasaguas, M.C. Aguayo-Torres, G. Gómez, and J. F. Paris, "Multiuser capacity and fairness evaluation of channel/QoS-aware multiplexing algorithms," *IEEE Network*, vol. 21, no. 3, pp. 24–30, 2007.

[13] J. Peisa and M. Meyer, "Analytical model for TCP file transfers over UMTS," in *Proceedings of International Conference on Third Generation Wireless and Beyond*, pp. 42–47, San Francisco, Calif, USA, June 2001.

[14] G. Gómez, *QoS modeling for end-to-end streaming performance evaluation over wireless access networks*, Ph.D. thesis, Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, Malaga, Spain, 2009.

[15] C. Bormann, C. Burmeister, M. Degermark, et al., "Robust Header Compression (ROHC)," RFC 3095, July 2001.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," *Computer Communication Review*, vol. 28, no. 4, pp. 303–314, 1998.

[17] L. Xu and J. Helzer, "Media streaming via TFRC: an analytical study of the impact of TFRC on user-perceived media quality," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, Barcelona, Spain, April 2006.

[18] G. Gómez, J. Poncela-González, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "A real-time end-to-end testbed for evaluating the performance of multimedia services," in *Proceedings og the 2nd International Workshop on Future Multimedia Networking (FMN '09)*, vol. 5630 of *Lecture Notes in Computer Science*, pp. 212–217, 2009.

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

[20] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.