Optimizing TCP and RLC Interaction in the UMTS Radio Access Network

Juan J. Alcaraz, Fernando Cerdan, and Joan García-Haro, Polytechnic University of Cartagena

Abstract
TCP, the dominant transport protocol for Internet applications, suffers severe performance degradation due to packet losses when a wireless link is present in the end-to-end path. For this reason, the 3G specification entity, 3GPP, has defined a reliable link layer protocol, RLC, to support packet switched services over UMTS. RLC provides error recovery in the radio access network by means of an ARQ algorithm. Early studies supported the benefit of using a reliable link protocol, while more recent studies outline new problems arising from RLC and TCP interaction and how to overcome them. This article describes the most relevant issues concerning TCP–RLC interaction and evaluates the most practical enhancement approaches, based on optimum parameter configuration at the transport and link layers. We devote special attention to RLC, whose specific configuration decisions are left to operators, and provide specific guidelines for setting its parameters. In addition, we propose two operational changes for enhancing the buffer management strategy, one of the main drawbacks of RLC.
access network delay, and error burstiness in the channel. Additionally, we propose slight modifications to the radio link layer protocol that, in contrast to other proposals in the literature, are scalable, do not violate the end-to-end semantics of TCP, and respect the principle of protocol layering, while improving TCP performance. Our proposals are:

- Active queue management for RLC buffers
- Acknowledgment delay control based on the status of the wireless channel

The rest of the article is organized as follows. We give an overview of TCP and describe the problems arising from the interaction between the transport layer and the link layer in 3G. We focus on the end-to-end approach, evaluating the IETF recommendations. We explain RLC operation and give a general guideline for its parameter configuration. We describe the proposed modifications for RLC. Finally, we outline open issues and offer some conclusions.

**Problem Description**

**An Overview of TCP**

TCP is the most commonly used protocol to provide reliable transport service in the Internet. TCP sends data segments (fixed amounts of application data that fill the TCP payload) according to a sliding transmission window, while the receiver generates cumulative acknowledgments (ACKs) for packets received in sequence. The arrival of an out-of-sequence packet causes retransmission of the last ACK. The size of the send window limits the amount of data the sender may have outstanding in the network at any given time. When a packet loss is detected, the congestion control mechanisms decrease the throughput of the source by reducing the size of the send window. Packet losses are detected by means of two mechanisms, each triggering a different congestion response:

- The arrival of three duplicate ACKs activates the Fast Retransmit — Fast Recovery algorithm, which reduces the window to 50 percent.
- The expiration of a retransmission timer, whose value depends on an estimation of the round-trip time (RTT) of the connection.

If no ACK is received in the meantime, the timer expires and the congestion window is reduced to one packet. In congestion-free periods, the sender is allowed to increase the window size by one segment for every ACK received (slow start phase) as long as the window is below a certain threshold. When this threshold is exceeded, the source enters the congestion avoidance phase, where every segment in the window has to be acknowledged to increase the window size by one segment.

**TCP Problems over 3G Links**

In this section we describe the characteristics of the radio bearers that may harm TCP performance, an issue that has been studied by several researchers [1, 7, 8] and the IETF Performance Implications of Link Characteristics (PILC) working group, whose conclusions are summarized in [6]. Taking this previous work into consideration, we survey the most relevant features focusing on 3G links.

**Latency** — The TCP source has to maintain a window size larger than or equal to the bandwidth delay product (BDP) of the connection in order to use the whole available bandwidth. In this way, the source can fill the link until the window slides forward upon the arrival of an ACK. In 3G networks the latency may range from 100 ms to 1 s, depending on RLC configuration and channel radio conditions. In the early stages of a TCP connection or after a timeout, the window size starts with a small value, generally one segment. A connection over a 3G link with a data rate of 384 kb/s and an RTT of 500 ms has a bandwidth delay product of 24 kbytes, which is large enough to cause link underuse for three to six round-trips, depending on the segment size and initial window size.

**Link Data Rate** — The data link rate offered by the radio bearer is dynamic. This rate variability is caused by changes in the traffic load of the radio cell, and alterations in the propagation conditions and user mobility. For example, a user moving from a cell offering a low bit rate to a higher-bit-rate cell might experience underutilization of the new available bandwidth if TCP is in congestion avoidance. Changing from a cell offering a high data rate to a lower one may cause overflow at the link layer buffer, packet losses, and a sudden increase in RTT (delay spikes).

**Delay Spikes** — A delay spike is a sudden increase in end-to-end delay. The most frequent reasons for a delay spike are:

- The ARQ algorithm recovering from an outage
- A handover
- Blocking due to the presence of higher-priority traffic in the channel
- Withdrawal of the channel when the network must provide access to higher-priority users

A delay spike may cause a retransmission timeout (RTO) in the TCP source. This event is known as spurious timeout if packets are just delayed at the link layer, but none is actually dropped. The retransmission triggered by a spurious timeout is redundant, and the reduction of throughput is generally excessive. In addition, several consecutive timeouts cause a decrease in the slow start threshold (ssthresh); therefore, the optimal window size is recovered more slowly.

**ACK Compression** — This phenomenon, also called burstiness of ACKs, is studied in [8], and affects the self-clocking nature of the TCP flow control. The queuing of ACK packets in the reverse path of a TCP flow may result in the almost instantaneous arrival of bursts of ACKs at the sender. This can break the TCP self-clocking operation and cause long packet bursts. This situation may happen when the downlink connection is delivering packets and the receiver, at the user side, is generating ACKs while the uplink connection is suffering a burst of frame losses. The uplink and downlink channels in the wideband code-division multiple access (WCDMA) radio interface are separated by about 200 MHz, so they may experience different and independent channel fading behavior. Therefore, one direction could be error-free while the other could be suffering a deep fading period.

**Packet Losses** — Depending on the link layer configuration, the radio bearer can support a low packet loss rate even in the presence of a transient high frame loss ratio in the channel. The ARQ mechanism may mask almost every packet loss in the wireless channel, but the recovery at the link layer appears to the higher layer as delay jitter. However, even if the link layer is configured to offer a highly reliable connection, packets may be lost because, as described in [4, 8], frame losses in the downlink wireless channel result in higher buffer occupancy at the network side RLC entity. If the situation persists, packets may be lost due to buffer overflow.

**Enhancement Strategies**

Many solutions have been proposed to improve data delivery at the transport layer over wireless links. They can be classified into four categories:

- **End-to-end**. This consists of optimizing the transport protocol.
— The development of new algorithms to enhance interlayer interactions
— The search for the optimum parameter configuration of the current link layer protocol standard

• **Split connections**: An intermediate node, generally called a performance enhancing proxy, splits the TCP connection between the sender and receiver into two separate connections. Specific enhancements are applied to the connection containing the wireless link.

• **Cross-layer**: The main concept of this scheme is to break the client-server working principle of the traditional layer stacks, allowing information exchange between protocols of different layers to adapt the behavior of each protocol accordingly.

### Optimizing TCP

The most practical end-to-end approach consists of configuring TCP according to the properties of 3G radio bearers. In [6] the IETF summarizes and recommends some currently approved configuration options that may enhance TCP over 3G links. In this section we explain these options and present our own evaluation based on simulation experiments. The simulated channel generates error bursts according to the model described in [11] where the Doppler frequency ($f_d$) of the user equipment determines the average burst length with lower $f_d$ values causing longer bursts of errors. The Doppler frequency represents the deviation from the nominal channel frequency at the receiver side, and depends on the user speed. It is usual to employ the normalized Doppler frequency, which, in our environment, is equal to the product of $f_d$ and the radio frame duration (10 ms).

To evaluate the real benefit of the proposed TCP options, the RLC parameters were set according to the optimizing considerations described later and detailed in Table 1 (see Fig. 4 for an explanation of the abbreviations). The radio bearer nominal rate is 384 kb/s in both directions. The error probability is the same in the uplink and downlink. Unless stated otherwise, the frame loss ratio is fixed at 10 percent, a typical UMTS design value.

#### Increase of Bandwidth Utilization

The goal is to avoid idle initial periods of the connection and react faster to sudden bandwidth increases even if TCP is in congestion avoidance. Two configuration options were proposed to meet this target, *Increased Initial Window* and *IP MTU Larger than Default*. The first mechanism has been proven as a safe option and is especially effective when transmitting few data packets. The second scheme allows the source to use larger packet sizes; therefore, the transmission window increases faster. Figure 1 shows how the configuration values of these parameters affect the downloading time of a 5-kbyte file. The figure shows that using 1500-byte packets, which is also the maximum allowed in the UMTS radio access network, in combination with an initial window of 4 packets provides the best download times.

#### Robustness against Packet Losses

Packet losses should not be a cause of concern because, according to 3GPP quality of service (QoS) objectives, the packet loss probability of the radio bearer should range from $10^{-6}$ to $10^{-9}$. Nevertheless, the IETF recommends two mechanisms, *limited transfer* and *SACK*, whose main objective is efficient recovery from packet losses. In a SACK connection, the receiver can explicitly inform which packets are received, even if they arrive out of sequence. This permits the sender to retransmit only the missing packets. Limited transfer extends Fast Retransmit — Fast Recovery mechanisms to connections with small congestion windows, and is therefore best suited to short-lived flows.

### Table 1. Simulation parameters setting values. RLC parameters not present in this table are deactivated.

<table>
<thead>
<tr>
<th>RLC parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDU payload size</td>
<td>320 bits</td>
</tr>
<tr>
<td>PDUs per TTI</td>
<td>12</td>
</tr>
<tr>
<td>TTI (transmission time interval)</td>
<td>10 ms</td>
</tr>
<tr>
<td>RB nominal bit rate</td>
<td>384 kb/s</td>
</tr>
<tr>
<td>Transmission window</td>
<td>2047 PDUs</td>
</tr>
<tr>
<td>Allowed retransmissions (maxDAT)</td>
<td>10</td>
</tr>
<tr>
<td>In-order-delivery</td>
<td>True</td>
</tr>
<tr>
<td>Buffer size</td>
<td>80 SDUs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RLC one-shot polling triggers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Poll window = 50%</td>
<td></td>
</tr>
<tr>
<td>Last PDU in buffer</td>
<td></td>
</tr>
<tr>
<td>Last retransmitted PDU</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RLC recurrent poll timers</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poll timer</td>
<td>110 ms</td>
</tr>
<tr>
<td>Receiver status generation</td>
<td>Setting</td>
</tr>
<tr>
<td>Status prohibit timer</td>
<td>300 ms</td>
</tr>
<tr>
<td>Missing PDU detection</td>
<td>True</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection characteristics</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTT wireless access network</td>
<td>100 ms</td>
</tr>
<tr>
<td>RTT fixed network</td>
<td>200 ms</td>
</tr>
<tr>
<td>Wireless channel average frame loss ratio</td>
<td>10%</td>
</tr>
<tr>
<td>Normalized doppler frequency</td>
<td>0.01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TCP parameters</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP flavor</td>
<td>Reno</td>
</tr>
<tr>
<td>Maximum TCP/IP packet size</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Maximum allowed window</td>
<td>64 kbytes</td>
</tr>
<tr>
<td>Initial window</td>
<td>1</td>
</tr>
</tbody>
</table>

The selection of TCP configuration options within the framework of approved IETF recommendations

• **Link layer**: The objective of this approach is to hide wireless errors from the transport layer, while minimizing negative interactions. Proposals in this area can be further classified into two subcategories:
For connections with a large bandwidth delay product, the probability of multiple packet losses in a single window increases. In such cases TCP SACK has proven to be effective, providing higher robustness than other TCP implementations. Previous results [12], based on extensive measurements on 2.5G networks, showed that the SACK option improves the goodput of TCP flows by about 10 percent. The goodput estimates the amount of successfully delivered data per second, providing the user perceived quality of the connection, instead of the sender data rate (throughput), which does not account for packet losses and retransmissions. Our simulation results, shown in Fig. 2, compare TCP Reno and SACK goodput performance in a 3G network for several RLC buffer capacities. We can see how the SACK option improves TCP goodput, especially for lower buffer sizes, where overflow causes frequent packet losses.

Reduction of Delay Jitter — The conservative computation algorithm of the TCP retransmission timer can mitigate but not avoid the problems arising from the delay jitter and delay spikes in a radio bearer. The IETF proposal for solving this problem is the TCP timestamp option, which permits TCP to sample the RTT for each packet instead of sampling it once in each window. A higher sampling rate should allow the sender to react more quickly to sudden increases of the delay after updating the RTO value to a more suitable one. However, this is not always true. Many previous studies [11] have shown that when the user equipment is stationary or moving slowly, frame error bursts in the wireless channel are highly correlated. This means an increase in the average length of frame loss bursts in the channel. The problem arises when relatively long periods of good channel conditions are followed by a long degraded period. In this case the timestamp option results in a higher probability of spurious timer expiration because, due to the higher sampling rate, it is probable that, during an error-free period, the estimation of the RTT could rapidly converge to an accurate but excessively small value, making the sender very vulnerable to delay spikes. The non-timestamp-based measure is coarser, but also more conservative in this scenario.

Figure 3 shows an example of the evolution of the RTT measure, with and without the timestamp option, in a situation where only the timestamp option suffers from spurious timeouts. The effect of the timestamp option can also be noticed in Fig. 2. Experimental measurements in [12] also showed this degradation in 2.5G networks, although no explanation was provided.

Optimizing RLC

Optimizing TCP performance by means of a suitable configuration of RLC parameters has the advantage of not requiring any modification at the end hosts. The following subsections describe the RLC operation and provide specific recommendations for setting the parameters, considering the permitted 3GPP values, to optimize data transport.

RLC AM Operation

The RLC AM is basically an SR-ARQ link layer protocol. As shown in Fig. 4, packets from higher layers (service data units, SDUs) are segmented and concatenated into link layer frames (protocol data units, PDUs) for transmission over the radio link. Figure 5 shows the basic operation of the error recovery algorithm. The receiver side detects missing PDUs based on the error control field or by detecting gaps in the frame sequence numbers. The RLC receiver side requests retransmission by sending back a bitmap report, called status PDU, informing which frame within the receiving window has been received (ACK) or lost (negative ACK, NACK). Upon reception of a status message, the sender can advance its transmission window if one or more in-sequence frames are acknowledged, so that new PDUs can be sent. If there are NACKs in the status message, the sender retransmits the missing PDUs giving them priority over new ones.

A status message is either issued when the receiver is polled by the sender or self-triggered. At the sender side, a polling request is made by marking the poll bit in the header of a PDU. The poll request can be triggered by seven configurable mechanisms, two of them considered recurrent:

- **Timer-based polling**, consisting of a periodic timer whose expiration triggers a poll.
- **Poll timer**, also driven by a periodic timer that is only started when a poll PDU is sent. The aim of this timer is to retransmit the poll if it is lost in the link, so the timer is stopped upon receiving a status message.

The rest of the mechanisms are considered one-shot polling methods, because they are triggered when the sender is in a particular state:

- **Last PDU in buffer** activates the poll bit in the last frame in the transmission buffer.
- **Last PDU in retransmission buffer**, same as above for frames in the retransmission buffer.
- **Poll every N PDU** triggers the poll for every N PDUs sent.
Poll every N SDU, for SDUs.
Window-based polling consists of marking the poll bit when a certain percentage of the transmission window is sent.

The receiver side transmits a spontaneous status message when any of the following status triggers are configured: detection of missing PDUs, which causes the sending of a status message when a missing PDU is detected, or periodic status report, which is driven by a periodic timer.

An excessive generation of status messages may waste energy in the mobile device and trigger redundant retransmissions at the link layer. The maximum signaling rate is controlled by means of two timers. The poll prohibit timer, in the sender side, and the status prohibit timer, in the receiver side, set the minimum allowed time lapse between successive transmissions of signaling frames at each side.

General Framework for RLC Parameter Configuration

Based on previous research and our own findings, this section provides a general guideline for configuring the radio network link layer to enhance upper layer performance. These configuration options have been checked by means of extensive simulation experiments. The TCP version used was TCP Reno, without the timestamp option. Unless stated otherwise, the parameter settings are those shown in Table 1.

Polling Strategy — The RLC AM polling function should be sufficiently frequent to make full use of the available link bandwidth and reduce packet queuing time. However, an excessively aggressive polling strategy has two main drawbacks:

• The possible generation of unnecessary control messages, wasting radio resources and increasing mobile energy consumption

• The possible triggering of redundant retransmissions

It is advisable to use a combination of polling mechanisms in both the sender and the receiver.

Figure 3. Variables of a TCP Reno connection with and without the timestamp option, running over an RLC radio bearer with the same channel evolution. The wireless link suffers an average frame error ratio of 20 percent in both directions. Errors in the channel occur in bursts. For the sake of clarity, only frame errors in the downlink direction are shown. a) TCP with timestamp option. The value of the computed RTO rapidly converges to the measured RTT value. Three consecutive delay spikes result in three spurious RTO timeouts. b) TCP without timestamp option maintains a more conservative RTO value under good wireless conditions. There are no timer expirations.

Transmission Buffer Size — In case of full buffer occupancy, the RLC specification [3] defines a simple droptail policy. The buffer size is not a protocol parameter but should be carefully selected when dimensioning the hardware and configuring the operating system of the equipment. Buffer overflow in the RLC buffer has been identified as one of the main threats to transport layer performance. In [4] it was shown that, under some assumptions (e.g., error-free uplink channel and uncorrelated downlink frame losses), buffer occupancy is limited to...
the maximum allowed value of the TCP transmission window ($awnd$), usually 64 kbytes. However, more realistic network situations, with burst errors in the wireless channel, in both the uplink and downlink, show that the $awnd$ value may be exceeded when the channel is temporally closed and a spurious timeout occurs at the transport layer. The performance degradation caused by a small buffer size can be seen in Figs. 3 and 10, where the RLC queue size is expressed in maximum sized SDUs (1500 bytes).

**Discarding Mode** — RLC can be configured in two different SDU discarding modes:

- When its buffering time exceeds a certain amount of time
- When one of the frames containing a segment of the PDU is retransmitted a specific number of times ($maxDAT$)

Previous results [4] show that the second strategy benefits TCP, especially when there is a high limit in the number of retransmissions. Our simulation experiments show that the optimum value of $maxDAT$ depends on the round trip time of the radio access network, ranging from 6 to 8 when the round trip delay varies from 60 ms to 300 ms. On the other hand, the burstiness degree of frame losses in the channel has no notable impact on these optimum values. According to the common delays of the radio network [14], a value of 10 for $maxDAT$ could be considered safe for a wide range of network situations.

The parameter $maxDAT$ has a strong interaction with the polling frequency. If $maxDAT$ is set to a low value (less than 10) and the polling strategy is aggressive, the link layer may be forced to discard long bursts

**Figure 4.** Elements and protocols involved in the end-to-end connection. The wireless channel connects the user equipment with the UMTS radio station (Node B). The Node B transparently connects the user data flow with the radio network controller. The lower layer offers an unreliable transport service to the RLC protocol through the dedicated transport channel.

**Figure 5. Basic RLC AM error recovery operation with status prohibit timer.**
of packets when a transient period of high error rate occurs in the channel. The reason is that, according to the RLC protocol operation, every time a polling action is triggered, the sender retransmits the highest unacknowledged frame in order to obtain the maximum information from the generated status message. If the error period is sufficiently long, this polling frame could reach the maximum number of retransmissions before the connection is re-established, especially if the polling frequency is high. When maxDAT is reached, the frame is discarded, as well as all the unacknowledged ones, causing a burst of packet losses.

In-Order Delivery — The RLC layer may be configured to deliver packets in sequence to upper layers. In a TCP connection, out-of-sequence packets have a negative effect on TCP throughput, which was studied in depth in [15]. It is generally accepted that in-sequence delivery at RLC should be the selected option when serving TCP flows, as long as more robust TCP versions are not developed and deployed. In the simulation environment depicted in Table 1, the goodput performance of TCP Reno fell by almost 40 per-

![Figure 6](image-url)

Figure 6. a) TCP goodput for different values of the status prohibit timer; and b). signaling efficiency for different values of the status prohibit timer.

![Figure 7](image-url)

Figure 7. a) Influence of the RLC window size on TCP throughput for different radio access network delays; b) influence of the RLC window size on TCP end-to-end delay for different radio access network delays.

![Figure 8](image-url)

Figure 8. Implementation of the RED buffer management algorithm at the RLC layer.
cent when the RLC did not deliver packets in sequence.

**Proposed Improvements to RLC**

Our proposals for optimizing radio bearer behavior are based on the idea that when the link layer has been configured to be reliable, only buffer overflow may cause packet losses in the radio access network. The RLC entity will experience increasing buffer occupancy during periods with errors in the channel, such as a router in a congestion situation. The drop-tail packet dropping strategy of the current RLC standard may cause the loss of consecutive packets, which have a negative effect on TCP throughput. Hence, it seems appropriate to apply known congestion management techniques from Internet routers to handle this congestion-like situation within a radio bearer flow.

We propose an adaptation of the following two mechanisms:

- Random early detection
- ACK delay control

These approaches present two common advantages over other schemes like split connections. First, they do not require snooping into packet headers, so they are compatible with security schemes. Second, they keep the end-to-end semantics of TCP, because no transport layer packet is generated at intermediate nodes.

**Random Early Detection**

Random early detection (RED) [16] consists of detecting incipient congestion by computing the average queue size. The congestion is notified to the TCP source by dropping a packet arriving at the node. This lets the source adapt its sending rate before the buffer capacity is exceeded and consecutive packets are discarded. When the average queue size exceeds a preset threshold \( \text{min}_\text{th} \), this is taken to be a sign of congestion and the node could drop a packet with a certain probability. This probability is a function of the queue size and the number of packets accepted since the last discarded packet within the congested period. Figure 8 shows how this mechanism could be adapted to handle the congestion situation in the downlink buffer of the network side RLC entity. Given the small number of flows multiplexed in the RLC buffer, using the instantaneous queue size instead of the average queue size allows a faster reaction against signals of channel degradation and increases the algorithm efficiency.

Several advantages may be obtained from the RED strategy:

- **Long bursts of packet losses are avoided.**
- **The average buffer occupancy is held around a certain value, and so packet delay is also controlled.**
- RED is a resource effective algorithm and, with a relatively small buffer we can achieve almost the same performance as a droptail scheme with an overdimensioned buffer size (no drops possible).
- **When several TCP flows are multiplexed within the same radio bearer, this technique ensures a fair share of the bandwidth between the flows.**
- **It is not necessary to separate each transport flow, so its scalability is higher than other proposals [17] aimed at preventing buffer overflow at the link layer.**

The main drawback of RED may be the additional computational cost of estimating a separate packet dropping probability for each user. In [16] simple algorithms were proposed for the computation of this probability, which were suitable for routers at the beginning of the last decade. The computational complexity of RED is still lower than that of schemes that require per-transport-flow management.

**ACK Delay Control**

ACK Delay Control was presented in [18] as an algorithm for routers aimed at improving the performance of TCP over satellite links. The basic idea of this algorithm is to delay acknowledgments traversing a node whose forward connection is congested. The congestion is detected when the buffer occupancy exceeds a fixed threshold \( \text{min}_\text{th} \). When this happens a delay is applied to the ACK packets in the reverse flow to slow down their departure rate. This mechanism takes advantage of the TCP self-clocking principle, as the reduction in the ACK flow will reduce the sending rate of the source.

In contrast to the basic algorithm, we propose that the delay applied to outgoing ACKs be determined by the buffer occupancy. This allows the source to gradually adapt its perception of the available bandwidth and the round-trip delay to conservative values. If the error burst in the wireless channel is long, the buffer occupancy will continuously increase. Higher buffer occupancies require higher reductions in the sending.
rate of the source, while for short error bursts excessive delay in the ACKs would be unnecessary. Another objective is to prevent spurious timeouts as long as no packet is dropped. A progressive increase of the ACK interarrival time helps the source increase its RTT measurement, preventing spurious timeouts. Besides, the magnitude of the delay spike caused by temporary channel unavailability is reduced, because the source receives ACKs within the degradation period.

The fixed delay scheme was compared to the variable delay strategy, simulating a wide range of configuration options. In most cases the variable delay scheme achieved better goodput performance, with gains ranging from 3 to 13 percent over the fixed delay strategy.

The diagram in Fig. 9 shows the main features of this algorithm. When the downlink buffer occupancy exceeds the minimum threshold \( \text{min}\_d \), the next incoming ACK packet in the reverse direction is delayed by \( \text{min}\_d \). If the queue occupancy keeps growing, the value of the induced delay is augmented linearly until the maximum delay value, \( \text{max}\_d \), is reached.

The advantages of this algorithm are:

- For RLC entities with small buffer, the probability of packet dropping is reduced.
- The size of the buffer required to achieve a target through-
  put is lowered with respect of the simple drdtai scheme.
- Reducing the reverse ACK pace helps the source to increase its round trip time estimation, reducing the probability of spurious timeouts.

This also benefits RLC entities with large buffers.

Figure 10 shows a comparison of the average goodput performance of these mechanisms. For almost every RLC buffer size, the new mechanisms improved the goodput. The parameter settings for the Delayed ACK and RED algorithms are shown in Table 2. Active queue management algorithms may also affect buffer dimensioning decisions. For the 384 kbps bidirectional bearer of this example, when an active queue management algorithm is used, a buffer capacity of 60 packets provides almost the best performance, while a drdtai scheme requires a buffering capacity of 70 packets to achieve the same performance. A smaller buffer size saves resources in network equipment and reduces the end-to-end delay.

Conclusions

One of the most used transport protocols for Internet applications, TCP Reno, requires protection against the frequent error bursts in wireless environments. This is the main purpose of the 3G link layer protocol when it is configured in Reno flows, so we propose incorporating a buffer management scheme in the RLC operation. Two existing algorithms have been adapted, RED and ACK Delay Control, taking into consideration the characteristics of radio bearers. We have shown that with proper configuration, both schemes may improve transport layer performance.

Acknowledgments

This work was supported by the Spanish Research Council under project ARFaQ (TEC2004-05622-C04-02/TCM).

References

[8] R. Chakravorty, A. Clark, and I. Pratt, “Optimizing Web Delivery over Wire-
[16] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Conges-

Biographies

JUAN J. ALCARAZ (juan.alcaraz@upc.es) obtained his engineering degree in the
telecommunication from the Polytechnic University of Valencia in 1999. His professional experience includes working at Siemens S.A. and at the Spanish 3G operator Xfera Móviles S.A. in the area of network design. He is currently working toward his Ph.D. degree in telecommunications at the Polytechnic University of Cartagena, Spain. His research interests include cellular access networks design and performance evaluation.

FERNANDO CERDÁN (fernando.cerdan@upct.es) obtained his engineering degree in telecommunications in 1994 from the Polytechnic University of Catalonia and received a Ph.D. degree from the same university in 2000. In 1996 he started working toward a Ph.D. degree in telecommunications supported by a national grant. He is currently a full associate professor at the Polytechnic University of Cartagena. Most of his published papers are in the field of service integration and QoS in wired IP networks, although his current interests include security and QoS in 3G and heterogeneous networks.

JOAN GARCÍA-HARO (joang.haro@upct.es) is a professor at the Polytechnic University of Cartagena. He is author or co-author of more than 50 papers mainly in the fields of switching and performance evaluation. From April 2002 to December 2004 he served as Editor-in-Chief of IEEE Global Communications Newsletter, included in IEEE Communications Magazine. He has been a Technical Editor of the same magazine since March 2001. He also holds an Honorable Mention for the IEEE Communications Society Best Tutorial Paper Award (1995).