Abstract

Performance of the RLC layer in UMTS is highly dependent on its initial configuration and possibility of dynamic reconfigurations. The goal of this paper is to present results of RLC AM performance analysis, with focus on the ARQ mechanism. The impact of different configuration options on RLC AM performance has been analyzed and verified by simulations. On the basis of obtained results, some optimal configuration guidelines have been proposed. Additionally a new, improved RLC AM architecture has been introduced and tested, along with some functional RLC extensions. The redesigned RLC layer enables efficient RLC SDU and STATUS PDU multiplexing. Proposed extensions allow decreasing retransmission time and avoiding possible deadlock situations. All improvements have a serious positive impact on overall UMTS RLC performance.

Keywords
UMTS, RLC layer, ARQ, performance

1. INTRODUCTION

The radio interface protocol stack is one of the main protocol stacks in UMTS [1]. It is located both in the UE (User Equipment) and in UTRAN (UMTS Terrestrial Radio Access Network). The radio interface protocol stack handles higher-level data, sent over the radio interface, by providing packet segmentation and reassembly, low-level ARQ (Automated Repeat Request) support, in-sequence delivery, data compression, flow control, QoS assurance, unicast, broadcast and multicast capabilities and several other functions. Its configuration is critical for the overall performance of UMTS. Correct configuration can maximize system performance, while incorrect configuration can even render it not operational [2].

The radio interface protocol stack can be functionally decomposed into three main layers (Fig. 1) [3,4]. Each layer consists of functionally separate protocol entities belonging to different sublayers, and so the second layer can be further decomposed into MAC (Medium Access Control) [5], RLC (Radio Link Control) [6], PDCP (Packet Data Compression Protocol) and BMC (Broadcast Multicast Control) sublayers.

Fig. 1. Functional decomposition of the UMTS radio interface protocol stack

The UMTS RLC layer [6] is a much improved version of the RLC layer used in GSM/GPRS and EGPRS [7]. It supports three modes of operation: Transparent Mode (TM), Unacknowledged Mode (UM) and Acknowledged Mode (AM).

- TM is used mainly by real-time services (especially voice connections in the CS, Circuit Switched, domain) when there is no time (and/or need) to perform retransmissions or reordering. In TM, RLC functions are limited to segmentation and reassembly.
- UM is used by time non-critical services, which allow some packet loss (retransmissions are not required or can be performed by upper layers) but expect packets to be correctly ordered. UM supports the same functions as TM and additionally enables detection of missing RLC PDUs and in-sequence delivery.
- AM is the most sophisticated mode, which additionally implements an ARQ mechanism to support retransmissions of lost or erroneous RLC PDUs.
PDUs. Services that require reliable transport and can tolerate low-level retransmission delays should use AM.

There are three basic ARQ schemes: SW (Stop and Wait), GBN (Go Back N) and SR (Selective Repeat). In the most inefficient SW ARQ scheme, the transmitter waits for the receiver to acknowledge each packet before sending the next one. In GBN ARQ, after receiving a NACK (Negative Acknowledgement) for a packet the transmitter resends this packet and N-1 packets that follow. In the most efficient SR ARQ scheme only NACKed packets are retransmitted.

![Diagram of ARQ mechanism](image)

**Fig. 2. Operation of the ARQ mechanism in RLC**

RLC AM incorporates the SR ARQ mechanism, as depicted in Fig. 2. Each radio frame contains several RLC PDUs. An ACK (Acknowledgement) is sent once in a while from the receiver back to the transmitter, acknowledging several received PDUs. If some PDUs are discovered missing or erroneous (as PDU x in Fig. 2) then, along with the ACK, a NACK is sent stating which PDUs should be retransmitted. The transmitter resends missing PDUs (PDU x’ in Fig. 2) in one of the nearest radio frames. The STATUS PDUs, i.e. ACKs and NACKs, are sent by the receiver as part of its regular traffic. The average frequency of sending STATUS PDUs can be described as the average number of data PDUs sent between subsequent STATUS PDUs. In this paper, if status frequency equals z then it means that a STATUS PDU is sent every z PDUs (in Fig. 2 z=4).

The influence of some RLC parameters on its performance has already been discussed in our previous work [9]. In our research, we do not distinguish between UDP and TCP traffic because RLC is not really aware of conveyed higher-layer protocols. The goal of this paper is to present results of further RLC AM analysis and simulations, with focus on performance of the RLC ARQ mechanism, and suggest improvements, some RLC extensions and guidelines for optimal RLC AM configuration.

2. SIMULATION SETUP

To perform all research and verify suggested configurations and improvements, a UMTS RLC layer simulator has been created and used. The simulator fully supports RLC AM, UM and TM and operates in an environment very similar to a real UMTS radio protocol stack. Other radio interface protocol stack layers, such as RRC, PDCP, MAC, FP (Framing Protocol) and characteristics of the WCDMA (Wideband Code Division Multiple Access) air interface have been simulated, allowing to perform simulations in a realistic UMTS environment. The simulator has been functionally verified by interoperability tests with commercial 3G protocol testers: Tektronix K1297 Compact Protocol Tester and CATV Saulpit DCT2000.

The simulator supports RLC SDU segmentation and reassembly, the RLC ARQ mechanism, concatenation, status piggybacking, all control PDU types and maintains a send and reception window. It also supports some additional functional and architectural enhancements, discussed in this paper. Measurements can be performed in any part of the RLC layer during simulation – header and payload sizes, buffer occupancy, frame timestamps and other statistics are reported and logged. Selected measurements are passed to the simulated RLC layer, which can perform RLC configuration adjustments. During the simulation automated, controlled error injection can be performed to simulate stable or varying radio channel BER (Bit Error Rate) and BLER (Block Error Rate).

Most simulations have been performed using an asymmetric data channel, which parameters are presented in Table 1. The downlink (DL, from UTRAN to UE) channel throughput is 384 kbps, and the uplink (UL) feedback channel throughput is 32 kbps. Such disproportion helps to underline the impact of different channel parameters on RLC performance.

<table>
<thead>
<tr>
<th>Rate [kbps]</th>
<th>RLC mode</th>
<th>Logical Channel</th>
<th>Transport Channel</th>
<th>TTI</th>
<th>TFS Size</th>
<th>TFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL</td>
<td>384</td>
<td>AM</td>
<td>DTCCH</td>
<td>DCH</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>UL</td>
<td>32</td>
<td>AM</td>
<td>DTCCH</td>
<td>DCH</td>
<td>40</td>
<td>4</td>
</tr>
</tbody>
</table>

*Table 1. 384 Kbps DL / 32 Kbps UL dedicated channel configuration*
3. RESULTS

3.1 Status frequency

Fig. 3 presents average UL and DL retransmission delays in case of an asymmetric data channel, which parameters are presented in Table 1. Retransmission delay is defined here as the time between receiving an incorrect RLC AMD (Acknowledged Mode Data) PDU and receiving the correctly retransmitted PDU.

![Graph showing UL and DL retransmission delays](image)

Fig. 3. Average UL and DL retransmission delays for configuration from Table 1

Average UL retransmission delay is much shorter than average DL retransmission delay because in the simulated asymmetric scenario up to 12 times more PDUs are sent in DL than in UL in the same period of time. This means that, for the same status frequency, the RNC receives a STATUS PDU up to 12 times more rarely than the UE. Because of this, the RNC performs DL retransmissions with a significantly greater average delay than the UE performs UL retransmissions. Thus, the throughput of the UL (usually feedback) channel can seriously influence DL channel performance in terms of average retransmission delay. The greater UL channel throughput or lower status frequency, the shorter is the average DL retransmission delay.

On the other hand, the lower status frequency (the more often STATUS PDUs are sent), the more throughput is consumed by control traffic, which is presented in Fig. 4. Hence, a tradeoff needs to be made between retransmission time and throughput used for RLC control traffic. The traffic QoS requirements and actual channel BLER should be taken into account while making a decision whether it is more important to decrease retransmission delay or increase available throughput by adjusting STATUS PDU transmission frequency. The adjustment decision should be made by RLC or RRC each time when traffic characteristics begin to differ from the required QoS. RLC and RRC should be fully aware of the discussed impact of STATUS PDU transmission frequency on RLC performance.

![Graph showing average throughput](image)

Fig. 4. Average throughput consumed by control traffic

3.2 Piggybacking and improved RLC architecture

RLC supports an optional STATUS piggybacking mechanism, which enables STATUS PDUs to be piggybacked in unused portions of AMD PDUs, in place of padding. If piggybacking is not supported, then each STATUS PDU, although typically very short (ACK is 3 bytes long), needs to be sent as a separate PDU which causes a significant waste of throughput. Fig. 4a depicts the amount of throughput consumed by control traffic if piggybacking is not enabled.

The usage of piggybacking should significantly decrease the amount of throughput consumed by control traffic by using otherwise wasted (filled with padding) portions of AMD PDUs. This is true when RLC SDU concatenation is not enabled. It appears that, when using both piggybacking and RLC SDU concatenation, the expected decrease does not occur in case of heavy traffic, which is presented in Fig. 4b. The 3GPP RLC specification [6] does not address this problem. RLC AM architecture proposed by 3GPP [6] makes it impossible to effectively use concatenation and piggybacking together in case of high channel utilization. In the standard RLC, RLC SDU concatenation is done first, and then a STATUS PDU (if ready to send) is tried to be placed in the available padding. In case of heavy traffic, practically no padding is generated (concatenation efficiently distributes SDUs in PDUs) so STATUS PDUs cannot be piggybacked.

As a solution to this problem we have invented an improved and architecturally modified version of RLC AM, presented in Fig. 5. The proposed solution enables efficient transmission of piggybacked STATUS PDUs even in case of concatenation and heavy traffic. The Mux functional block presented in
Fig. 5 is a sophisticated PDU multiplexer, handling data from 4 priority queues. The priorities, from highest to lowest, are assigned to RESET/RESET_ACK PDUs, STATUS PDUs, retransmitted PDUs, and new (sent for the first time) PDUs. Every TTI (Transmission Time Interval) MAC informs RLC how many PDUs will be sent this TTI and the Mux prepares these PDUs using data from all 4 input buffers (priority queues). If there is a RESET or RESET_ACK to send, it is transmitted in the first order. Then, if there is a status to send, it is tried to be piggybacked in the retransmitted PDUs (if there are any to be sent) or Mux tries to segment and concatenate new SDUs in such a way that the STATUS PDU is piggybacked and the smallest possible amount of padding is added. Only if no such combination is possible the STATUS PDU is sent as a separate AMD PDU. The gain of using our improved RLC architecture is presented in Fig. 4c. The amount of throughput consumed by control traffic is apparently lower compared to standard RLC (Fig. 4b).

![Fig. 5. Improved RLC AM architecture](image)

### 3.3 Polling modes and RLC extensions

Frequency of sending status information is controlled either by the sender (by polling the receiver), or by the receiver itself (by unsolicited status reporting). If retransmission delay or control traffic overhead requirements are very strict, periodic timer polling or periodic timer status reporting should be used because it assures constant status transmission frequency. If there are no such strict QoS requirements, it is better to configure status frequency to be dependent on actual traffic rate or packet length by enabling per PDU, per SDU or window occupancy based polling.

![Fig. 6. Usage of poll prohibit timer and the new poll force timer](image)
Another problem is that if there is no traffic sent through the channel for some time, some of the last sent PDUs may not be acknowledged for long because a poll will not be sent until transmission is resumed. This has a great, negative influence on system performance in case of serious periodical changes in the amount of sent traffic. If some of the last PDUs are lost, the receiver might not detect this, and a time consuming higher-layer (e.g. TCP) retransmission might be triggered. Even if periodic status reporting is configured at the receiver, an RLC retransmission will not be triggered for all missing PDUs not followed by a correctly received PDU.

The presented scenarios show that the standard RLC ARQ mechanism is not capable of adjusting correctly to changing traffic conditions. To make it more flexible and avoid some performance problems we have developed two new extensions to the RLC standard. The first one is the poll force timer. It is started each time the first RLC PDU is transmitted after sending a poll and stopped when a poll is sent. If it expires, a poll is sent and the poll force timer is stopped (Fig. 7). The value of this timer should be higher than the average interval between polls. Usage of the poll force timer prevents poll interval from becoming too long and causing very long retransmission delays or deadlocks (Fig 6b). The second extension is the possibility of sending empty RLC PDUs, with just the poll bit set (the last PDU in Fig. 7). This has to be done when a poll should be sent but there are no more RLC PDUs to be sent. Such a PDU should be handled by the receiver by correctly interpreting the poll bit in the RLC PDU header, although the PDU does not contain any payload.

![Diagram of RLC PDUs](image)

**Fig. 7. Example of using introduced RLC extensions (poll force timer set to 7 TTI)**

4. CONCLUSIONS

Performed simulations showed that, while configuring the frequency of sending status information, a tradeoff has to be made between retransmission delay and channel throughput consumed by control traffic. Decreasing retransmission delay generally results in more throughput used for sending STATUS PDUs. Using status piggybacking can save some throughput, but the standard RLC architecture appeared to be inefficient when piggybacking and RLC SDU concatenation is enabled in case of heavy traffic. An improved RLC architecture has been proposed, implemented and tested. The improved RLC is functionally similar with the standard RLC, but effectively handles piggybacked status information and traffic priorities. Control information is sent with highest priority, retransmitted PDUs with medium priority and new PDUs with lowest priority, thus improving RLC queue handling and overall performance. Smart SDU concatenation and status piggybacking is supported.

Further analysis and simulations were focused on configuring and optimizing the poll and status generation mechanism, especially the frequency of sending status information. Several ARQ configuration guidelines were given and two new RLC extensions introduced, providing improved RLC reliability and efficiency. The proposed new poll force timer and possibility of transmitting empty PDUs, just for poll purposes, enable RLC to achieve better performance and operate more effectively without encountering otherwise possible deadlocks.

Presented analysis, simulation results, guidelines and improvements can be treated as a reference for designers and implementers of software for UMTS.

BIBLIOGRAPHY


[4] 3GPP TS 25.401: UTRAN Overall Description, ver. 5.7.0, 1/2004


[7] 3GPP TS 44.006: Mobile Station (MS) - Base Station System (BSS) interface: Radio Link Control/Medium Access Control (RLC/MAC) protocol, ver. 5.9.0, 12/2003
