

RLC Design for GERAN'00

1. INTRODUCTION

As agreed in SMG2#34, the RLC layer for GERAN will support three different modes, *acknowledged*, *unacknowledged* and *transparent* for conveying data from four traffic classes: conversational, streaming, interactive and background. This paper analyzes the needs of GERAN RLC for its 3 modes.

2. RADIO BEARERS TRANSPORT OVER TBF

It is proposed in [1] that several Radio Bearers may be transferred within the same TBF in order to efficiently multiplex data with same or different QoS from the same MS on the air interface, without need for extra-signalling between Radio Bearers. This mechanism would be applicable in e.g. the non-limiting situations below:

- OS2: conversational call and best effort data
- Interactive data and background data: Web-browsing and email transfer

The main advantages of this proposal are to:

- Avoid extra signalling between Radio Bearers and therefore
- Use silent periods of a Radio Bearer in an optimal way
- Spare TFI: One TBF may embed several data connections (note that those data connections may need several consecutive TBFs), instead of having in parallel several TBFs. This way one TFI would refer at once to all the RB it carries.



Fig. 1. Multiple Radio Bearers per TBF

3. RLC ENTITIES

Multiple QoS per TBF means that multiple RLC entities may be multiplexed onto the same TBF. E.g. some conversational traffic would be realized through unacknowledged RLC and best effort data through acknowledged RLC, both of them being transferred over the same TBF. One RLC entity should therefore be associated to an RLC mode: Ack or Unack.

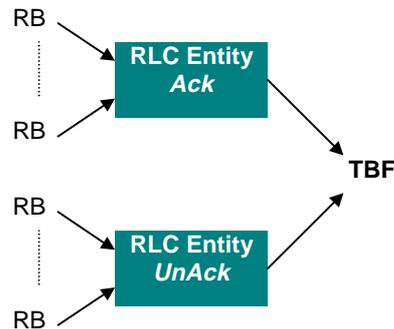


Fig. 2. Multiple RLC Entities per TBF

Multiplexing RLC Entities per TBF requires that distinction between RLC Entities be possible.

The implications of this proposal on RLC is analyzed below for Acknowledged and Unacknowledged modes respectively.

4. RLC ACKNOWLEDGED MODE

4.1 Limited & Unlimited Retransmissions

In order to limit some transfer delay for e.g. streaming data, it has been proposed that the retransmissions should be limited to a maximum given number [2].

Several parameters may affect the number of retransmissions allowed for a given maximum transfer delay, so that limiting the number of retransmissions is implementation dependent:

- (Buffer in the receiving application)
- Roundtrip delay
- Required Bitrate vs. Current throughput (channel conditions dependent)

Furthermore, a corrupted RLC block is of no use to upper layer as it leads to a corrupted PDCP PDU (one part of it, corresponding to the corrupted RLC PDU may be blanked), that cannot be used by the targetted application (e.g. for streaming). Limiting the number of retransmissions of RLC blocks hence decreasing the chances to receive correctly an RLC block, leads to increasing the probability of corrupted PDCP PDU, thus of loss of IP packets, and finally of no reception at the end-application. The table below compares both alternatives and their implications.

Source: Nokia

3 (6)

Cases	Limited retransmissions: max M			Unlimited retransmissions			
	RLC block reception	Action	Implication	RLC block Reception	Action	Implication	
N>M	X "too late"	Next RLC block	PDCP PDU corrupted. Nothing delivered	X	Block retransmitted until successful reception	Any PDCP PDU transmitted over the air will be correctly received. Delay: may be caught up by "free time" of other RLC blocks, and if needed by skipping PDCP PDU(s) not yet segmented at RLC. Transfer delay is in average within the requirements	
N=M	OK and just on time	Next RLC block	PDCP PDU delivered "on time" if <i>all other RLC blocks correctly received</i>	OK and just on time	Next RLC block		No additional free time
N<M	OK and "early"			OK and "early"			Additional free time

Notes:

M: max number of retransmissions

N: number of retransmissions needed to receive correctly the RLC block

Table 1. Limited vs Unlimited Retransmissions

Therefore it is suggested that no limitation be put on the maximum number of retransmissions: the current EGPRS RLC/MAC protocol need not be modified. It is up to the sending side to discard IP packets (PDCP PDUs), use preemptive retransmissions when needed.

Note that discarding PDCP PDUs for RT streaming is also under studies in UTRAN.

4.2 RLC blocks and Radio Bearers

The transfer of several Radio Bearers on the same TBF implies that distinction between RB be possible within the TBF¹. Therefore the RBid need to be embedded along with the data to which it refers. In order not to create unacceptable overhead and to redesign new channel coding it is proposed that RBid be embedded with the data part of the RLC block rather and not in the header part. There are two alternatives for including the RBid:

- Embed systematically the RBid within the data part of each block
- Embed the RBid only in the first block of the corresponding Radio Bearer: i.e. very first block and after each switch to this particular RB. This case requires one indicator of the presence of the RBid (RI: RBid Indicator).

Note that it is also proposed that one RLC Data Block contain only one RB, i.e. RBs are not multiplexed within RLC Data Blocks.

The first case implies a very slight throughput decrease as depicted in the table below.

MCS	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9
Throughput	8.8	11.2	14.8 13.6	17.6	22.4	29.6 27.2	44.8	54.4	59.2
Throughput with RBid	8.3	10.7	14.3 13.1	17.1	21.9	29.1 26.7	44.3	53.9	58.7
Throughput Loss	-5.7%	-4.5%	-3.5% -3.7%	-2.8%	-2.2%	-1.7% -1.9%	-1.1%	-0.9%	-0.8%

Table 2. Throughput Decrease due to inclusion of RBid in the data blocks

¹ If the TBF carries only one RB, RBid need not be embedded with the data to which it refers.

It should be noted that if the RBid is included only when there is a switch between radio bearers, the throughput decrease is dependent on the switch rate, and likely much lower than presented in the previous table. *In practice, this decrease is negligible.*

Because Incremental Redundancy may be used in acknowledged mode, combining between two blocks from two different Radio Bearers shall not be done. This could occur in case these blocks have the same block sequence number (BSN). Therefore in order to avoid this problem, it is proposed that one RLC Entity has only one BSN space shared among the different RB mapped onto this entity as is depicted below:

BSN	Data + RBid
0	RBid 1
1	RBid 1
2	RBid 3
3	RBid 2
4	RBid 2
5	RBid 2
6	RBid 3
7	RBid 3

RBid systematically embedded

BSN	RI	Data (+ RBid)
0	1	RBid 1
1	0	
2	1	RBid 3
3	1	RBid 2
4	0	
5	0	
6	1	RBid 3
7	0	

RBid embedded in first block of a given RB (after switch)

The first case, although its throughput is decreased, enables to directly identify to which RB a block belongs as soon as this block is correctly decoded. The other case relies on the correct decoding of the first block (very first or after switch to this RB) to be able to map the following blocks to the indicated RB. If e.g. block nr.3 is missed in the previous example (i.e. header correctly decoded but data part corrupted), following blocks 4 and 5 cannot be fully identified. Note however that block 3 cannot be interpreted as belonging to RBid3, since the RI in block 3 indicates a new RB. Whether RI need to be in the header part (replacing RSB or PI) or not is ffs: RI will not imply any change in any channel coding.

4.3 Segmentation / Reassembly

In order to optimize transmission of reassembled PDCP PDUs to upper layer, it may be beneficial not to deliver the RLC PDUs for reassembly, in sequence regardless of the RBid, but rather to transmit them "in-sequence" (by increasing BSN order) per RB and per PDCP PDU.

It is assumed one PDCP PDU carries only one RB. Depending how the application is running, different segmentation schemes are worth considering, depending on whether PDCP PDUs should be processed independently or not at RLC level:

a) Independent PDCP PDUs

In this case one PDCP PDU is mapped to an integer number of RLC blocks, i.e. one RLC Data Block may not contain data from two different PDCP PDUs, and therefore the last RLC block for that PDCP PDU may need padding:



Independent PDCP PDUs would allow for skipping the very next PDCP PDUs one-by-one without need for resegmentation at RLC level. All RLC PDUs may be transmitted regardless of the next PDCP PDU.

Skipping PDCP PDU allows for real-time streaming applications to catch-up time loss at RLC level as presented in 4.1.

b) PDCP PDUs concatenation from the same RB

Another possibility is to concatenate PDCP PDU issued from the same RB, to be transmitted over RLC. However, one RLC PDU shall not contain data from two different RBs.



This alternative allows for a higher possible throughput than with a). Similarly to a), only non-RLC segmented PDCP PDUs may be discarded. So in this case if there is an RLC PDU carrying the end of a PDCP PDU (1) and the beginning of the next PDCP PDU (2), this PDCP PDU (2) must be transmitted, but the next one may be discarded.

5. RLC UNACKNOWLEDGED MODE

5.1 RLC Blocks and Radio Bearers

As said earlier, it is proposed that several RLC Entities may be multiplexed onto the same TBF, requiring that distinction between RLC entities shall be possible. As it is also proposed that each RLC Entity has its own BSN space, then the distinction between RLC Entities cannot be made with BSN, but instead requires one bit in the header of all the RLC block indicating to which RLC Entity the block belongs. This bit could also be considered as an extended BSN space: each half would be reserved to a given RLC entity.

See 4.2. As retransmissions may not occur in unacknowledged mode, the RBid should be included systematically in each RLC block (see first case in 4.2), assuming several RBs may be transferred over the unack RLC entity, in order to identify directly to which RB a block belongs.

However, it is more likely to have a very limited number of RBs sharing the same unack RLC entity, therefore, a reduced RBid could be used instead of the RBid. Limiting to two² the number of RB in unack mode on the same TBF, would mean that one bit only is needed to distinguish them. This bit could be put in the header part of the RLC blocks. The advantage would be that even though the data part of an RLC block would be corrupted, the RB to which the block belongs may still be identifiable. This could then be indicated to the upper layer that could then simply discard the full PDCP PDU (indication to RLC would be needed probably, in order to avoid transmitting useless RLC blocks). Otherwise, having the RBid in the data part implies that a corrupted RLC block may not be indicated neither transferred to upper layer.

² If only one RB is carried in Unack mode, no RBid needs to be embedded.

5.2 Segmentation/Reassembly

Similar assumptions to 4.3

6. RLC TRANSPARENT MODE

RLC has no functionality when operating in transparent mode. The incoming SDUs are transferred to the MAC layer without being altered. No upper layer protocol information is removed. No RLC protocol information is added. All necessary signalling is made out of band [3].

7. CONCLUSION

This paper presented different alternatives for GERAN RLC. It is shown that multiple flows per TBF allows for additional flexibility enabling e.g. optimal use of silent periods without any additional significant overhead neither impact on existent protocol. It is further suggested that Rel'99 acknowledged mode RLC/MAC with IR be reused as is for RT streaming application.

8. REFERENCES

- [1] 3GPP TSG GERAN#1, GP-000246, "MAC Design for GERAN'00", Nokia, Seattle, 28 August – 1 September, 2000
- [2] 3GPP TSG GERAN AdHoc#1, Tdoc 2g00-085, "Limited Retransmissions for streaming", Lucent, Helsinki, 7-11 August, 2000
- [3] 3GPP TSG GERAN#1, GP-000242, "GERAN Stage 2 Description", GERAN AdHoc#1, Seattle, 28 August – 1 September, 2000