

Agenda item: 5.2.2
Source: Motorola
Title: RLC Status Report SUFIs for PDU/PDU segments ACK/NACK
Document for: Discussion and Approval

1 Introduction

In this document, we discuss the status report SUFI formats for ACK/NACK RLC PDUs and RLC PDU segments. As agreed, LTE supports both flexible RLC PDU size and unlimited number of re-segmentations for RLC PDU retransmission. It is worthwhile to examine what status report SUFIs are required due to the support of RLC PDU segments.

2 SUFIs for RLC PDU/PDU Segments ACK/NACK

2.1 The SLIST super-field

SLIST SUFI is proposed to report missing RLC PDU segments. It is currently FFS how ACK/NACK STATUS will be reported for PDU segments. Release 7 provides ACK, LIST, BITMAP and RLIST SUFI types for reporting the receiving status of RLC PDUs. However, none of these approaches are sufficient for PDU segment status, because PDU segments are identified by (sequence number, segment offset) pair [1], rather than by sequence number alone. Note that there may be multiple numbers of re-segmentations performed on the same RLC PDU, and it may be necessary to report on the status of a gap consisting of more than one missing PDU segments. Thus the end of the missing data needs to be specified as well, either by indicating the end offset, or the length of total missing segment(s) in bytes. A new SUFI type SLIST is needed, therefore, to report missing RLC PDU segments.

The SLIST Super-Field consists of a type identifier field (SLIST), a list length field (LENGTH) and a list of LENGTH number of (SN, SO, L) triplets for missing segments, as shown in Figure 1.

Type = SLIST
LENGTH
SN ₁
SO ₁
L ₁
SN ₂
SO ₂
L ₂
...
SN _{LENGTH}
SO _{LENGTH}
L _{LENGTH}

Figure 1: The SLIST field in a STATUS PDU

LENGTH

Length: FFS

The number of (SN_{*i*}, SO_{*i*}, L_{*i*}) triplets in the super-field of type SLIST.

SN_{*i*}

Length: FFS, and identical to configured AMD PDU header SN field.

“Sequence Number” of AMD PDU segment, which was not correctly received.

SO_i

Length: FFS, and identical to configured AMD PDU header LI field.

“Segment Offset” of AMD PDU segment, which was not correctly received. The value of SO_i is defined the same as the SO field of RLC PDU segment header.

L_i

Length: FFS, and identical to configured AMD PDU header LI field.

Number of consecutive bytes not correctly received in AMD PDU with sequence number SN_i starting from byte offset SO_i.

Note that the end of missing data within an AMD PDU can be indicated by either the offset of the ending position, or the byte counter of consecutive loss as in Figure 1. The length of field required for both cases should be the same as the LI field of AMD PDU header. Therefore, either one can be used.

We propose to not support a RLC PDU segment ACK/NACK SUFI based on the Release 7 BITMAP SUFI, for the reason that PDU segments are indexed by byte offset, instead of subsequence number, and there may be multiple numbers of re-segmentations in transit. Therefore, every byte of the original PDU has to be represented by one bit in the status bitmap in order to avoid ambiguity, which yields a much less efficient way of status reporting.

Proposal 1: BITMAP style SUFI not supported in EUTRA for RLC PDU segment ACK/NACK STATUS report.

It should be noted that SLIST is only used when status of PDU segments needs to be reported. The status of RLC PDUs are reported through separate SUFIs, such as LIST, BITMAP, RLIST, ACK, ALIST (the new SUFI proposed in Section 2.2), as specified in the technical specification. It is possible to combine SLIST and LIST into one SUFI by including an extra “type” bit for each entry in the list. However, given the fact that the RLC ARQ retransmission is a rare occurrence (in the worst case of the order of 1e-4), we propose to keep SLIST as a distinct SUFI, in order to reduce the ACK/NACK overhead for RLC PDUs.

2.2 The CLIST super-field

The CLIST SUFI is proposed to combine the ACK/NACK of PDUs with the ACK/NACK of PDU segments.

The CLIST Super-Field consists of a type identifier field (CLIST), a list length field (LENGTH) and a list of LENGTH number of (SN_{begin}, SO_{begin}, SN_{end}, SO_{end}) quartets for missing PDUs/PDU segments as shown in Figure 3 below:

Type = CLIST
LENGTH
SN _{begin,1}
SO _{begin,1}
SN _{end,1}
SO _{end,1}
SN _{begin,2}
SO _{begin,2}
SN _{end,2}
SO _{end,2}
...
SN _{begin,LENGTH}
SO _{begin,LENGTH}
SN _{end,LENGTH}
SO _{end,LENGTH}

Figure 2: The CLIST field in a STATUS PDU

LENGTH

Length: FFS

The number of (SN_{begin i}, SO_{begin i}, SN_{end i}, SO_{end i}) quartet in the super-field of type CLIST.

SN_{begin,i} , SN_{end,i}

Length: FFS, and identical to configured AMD PDU header SN field.

“Sequence Number” of RLC PDU/PDU segment.

$SO_{begin,i}$, $SO_{end,i}$

Length: FFS, and identical to configured AMD PDU segment header SO field.

“Segment Offset in bytes” of RLC PDU segment, with respect to the original RLC PDU.

Define the comparison between (SN_a, SO_a) and (SN_b, SO_b) as:

$(SN_a, SO_a) = (SN_b, SO_b)$, iff $((SN_a=SN_b) \text{ AND } (SO_a=SO_b))$;

$(SN_a, SO_a) < (SN_b, SO_b)$, iff $((SN_a < SN_b) \text{ OR } ((SN_a=SN_b) \text{ AND } (SO_a < SO_b)))$;

$(SN_a, SO_a) > (SN_b, SO_b)$, iff $((SN_a > SN_b) \text{ OR } ((SN_a=SN_b) \text{ AND } (SO_a > SO_b)))$;

Then $(SN_{begin,1}, SO_{begin,1}) < (SN_{end,1}, SO_{end,1}) < (SN_{begin,2}, SO_{begin,2}) < (SN_{end,2}, SO_{end,2}) < \dots < (SN_{begin,LENGTH}, SO_{begin,LENGTH}) < (SN_{end,LENGTH}, SO_{end,LENGTH})$.

A CLIST SUFI acknowledges RLC PDUs/PDU segments with (SN, SO) pairs satisfying $(SN, SO) < (SN_{begin,1}, SO_{begin,1})$, or $(SN_{end,1}, SO_{end,1}) < (SN, SO) < (SN_{begin,2}, SO_{begin,2})$, or, ..., or $(SN_{end,LENGTH-1}, SO_{end,LENGTH-1}) < (SN, SO) < (SN_{begin,LENGTH}, SO_{begin,LENGTH})$ as having been received successfully. In addition it indicates that PDUs/PDU segments with (SN, SO) pairs falling into range $(SN_{begin,1}, SO_{begin,1}) \leq (SN, SO) \leq (SN_{end,1}, SO_{end,1})$, or $(SN_{begin,2}, SO_{begin,2}) \leq (SN, SO) \leq (SN_{end,2}, SO_{end,2})$, or, ..., or $(SN_{begin,LENGTH}, SO_{begin,LENGTH}) \leq (SN, SO) \leq (SN_{end,LENGTH}, SO_{end,LENGTH})$ are missing.

The information conveyed by CLIST can be carried by combined usage of LIST/RLIST and SLIST on PDUs and PDU segments respectively. However, CLIST can be more efficient by ACK/NACK PDU and PDU segments jointly under certain scenarios. Note that CLIST may be more useful when PDCP sequence number is reused as RLC PDU sequence number [2][3], because SDU segments need to be ACK/NACK more often then.

3 Conclusions

Two new SUFIs are defined in this document for ACK/NACK status reporting for RLC PDUs/PDU segments.

- SLIST SUFI is proposed to report missing PDU segments;
- CLIST SUFI is proposed to ACK/NACK PDUs and ACK/NACK of PDU segments jointly;

In addition, LIST, BITMAP, ACK and RLIST SUFIs as specified in [4] shall be supported for RLC PDUs ACK/NACK status reporting.

It is proposed to agree on the SUFI formats suggested above. Motorola can provide text proposal for inclusion into the Stage 3 specification based on agreements.

References

- [1] 3GPP TS 36.322, “Radio Link Control (RLC) protocol specification (Release 8)”
- [2] R2-071837, “L2 sequence number in LTE”, Ericsson
- [3] R2-073533, “RLC PDU SN: To Reuse PDCP SN or not”, Motorola
- [4] 3GPP TS 25.322, “Radio Link Control (RLC) protocol specification (Release 7)”